

LilyPond

Il compositore tipografico per la musica

Guida alla Notazione

Il team di sviluppo di LilyPond

Questo manuale costituisce la guida di riferimento per tutti gli aspetti relativi alla notazione musicale in LilyPond versione 2.25.35. Si presuppone che il lettore conosca il materiale esposto nel Sezione “Manuale di Apprendimento” in *Manuale di Apprendimento*.

Questo manuale è disponibile in altri formati ed è integrato col resto della documentazione. Maggiori informazioni in Sezione “Manuali” in *Informazioni generali*.

La documentazione completa si trova all’indirizzo <https://lilypond.org/>.

Copyright © 1999–2023 degli autori.

La traduzione della seguente nota di copyright è gentilmente offerta alle persone che non parlano inglese, ma solo la nota in inglese ha valore legale.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

È garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della GNU Free Documentation License, versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation; senza alcuna sezione non modificabile. Una copia della licenza si trova nella sezione intitolata "GNU Free Documentation License".

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Per la versione di LilyPond 2.25.35

Sommario

1	Notazione musicale	1
1.1	Altezze	1
1.1.1	Inserimento delle altezze	1
	Ottava assoluta	1
	Ottava relativa	2
	Alterazioni	6
	Nomi delle note in altre lingue	8
1.1.2	Modifica di più altezze	10
	Controlli di ottava	10
	Trasposizione	11
	Inversione	14
	Retrogradazione	15
	Trasposizioni modali	15
1.1.3	Aspetto delle altezze	17
	Chiave	17
	Armatatura di chiave	22
	Segni di ottavazione	24
	Trasporto strumentale	27
	Alterazioni automatiche	29
	Glifi di alterazione alternativi	36
	Ambitus	38
1.1.4	Teste di nota	41
	Teste di nota speciali	41
	Testa di nota con nome della nota	42
	Teste di nota a forma variabile	44
	Improvvisazione	47
1.2	Ritmi	48
1.2.1	Inserimento delle durate	49
	Durata	49
	Gruppi irregolari	51
	Scalare le durate	57
	Legature di valore	58
1.2.2	Inserimento delle pause	63
	Pause	63
	Pause invisibili	65
	Pause d'intero	66
1.2.3	Aspetto dei ritmi	71
	Indicazione di tempo	71
	Indicazioni metronomiche	75
	Anacrusi	78
	Musica in tempo libero	79
	Notazione polimetrica	81
	Divisione automatica delle note	84
	Mostrare i ritmi della melodia	86
1.2.4	Travature	88
	Travature automatiche	88
	Impostare il comportamento delle travature automatiche	91
	Travature manuali	101

Travature a raggiera	104
1.2.5 Battute	105
Stanghette	105
Numeri di battuta	112
Controlli di battuta e del numero di battuta	118
Segni di chiamata	119
1.2.6 Questioni ritmiche particolari	122
Abbellimenti	122
Allineamento sulle cadenze	127
Gestione del tempo	128
1.3 Segni di espressione	129
1.3.1 Segni di espressione collegati alle note	129
Articolazioni e abbellimenti	129
Dinamiche	132
Nuove indicazioni dinamiche	140
1.3.2 Indicazioni espressive curvilinee	142
Legature di portamento	142
Legature di frase	145
Respiri	147
Portamenti indeterminati discendenti (cadute) e ascendenti	148
1.3.3 Indicazioni espressive lineari	149
Glissando	149
Arpeggio	154
Trilli	158
1.4 Ripetizioni	160
1.4.1 Ripetizioni lunghe	161
Ripetizioni normali	161
Indicazioni di ripetizione manuali	169
Ripetizioni ricopiate	171
1.4.2 Ripetizioni brevi	173
Ripetizioni con percentuale	173
Ripetizioni con tremolo	176
1.5 Note simultanee	178
1.5.1 Una voce	178
Note in un accordo	178
Ripetizione di un accordo	180
Espressioni simultanee	182
Cluster	183
1.5.2 Più voci	184
Polifonia su un solo rigo	184
Stili di voce	187
Risoluzione delle collisioni	188
Accorpare le pause	193
Combinazione automatica delle parti	194
Scrivere la musica in parallelo	199
1.6 Notazione del rigo	202
1.6.1 Aspetto del rigo	202
Istanziare nuovi righi	202
Raggruppare i righi	203
Gruppi di righi annidati	207
Separare i sistemi	209
1.6.2 Modificare singoli righi	210

Simbolo del rigo	210
Righi ossia	213
Nascondere i righi	217
1.6.3 Scrittura delle parti	219
Nomi degli strumenti	219
Citare altre voci	222
Formattazione delle notine	226
Compressione delle misure vuote	231
1.7 Note editoriali	231
1.7.1 Interne al rigo	231
Scelta della dimensione del tipo di carattere	231
Indicazioni di diteggiatura	235
Dita che scivolano	237
Note nascoste	240
Colorare gli oggetti	241
Parentesi	243
Gambi	244
1.7.2 Esterne al rigo	246
Nomi delle note	246
Nuvoletta di aiuto	247
Linee della griglia	248
Parentesi analitiche	250
1.8 Testo	253
1.8.1 Inserimento del testo	253
Scritte	253
Estensori del testo	254
Indicazioni testuali	256
Testo separato	258
1.8.2 Formattazione del testo	260
Introduzione al testo a margine	260
Scelta del tipo di carattere e della dimensione	261
Allineamento del testo	264
Notazione grafica nel blocco markup	268
Notazione musicale nel blocco markup	270
Testo formattato su più pagine	273
1.8.3 Tipi di carattere	273
Tipi di carattere in dettaglio	274
Tipi di carattere per singolo oggetto	277
Tipi di carattere per l'intero documento	277
2 Notazione specialistica	280
2.1 Musica vocale	280
2.1.1 Notazione comune per la musica vocale	280
Riferimenti per la musica vocale	280
Inserimento del testo vocale	281
Allineamento del testo alla melodia	282
Durate automatiche delle sillabe	284
Durate manuali delle sillabe	287
Più sillabe in una nota	288
Più note in una sillaba	289
Estensori e trattini	292

Cambi graduali di vocale	292
2.1.2 Tecniche specifiche per il testo vocale	293
Lavorare con testo e variabili	293
Posizionamento verticale del testo	295
Posizionamento orizzontale delle sillabe	297
Testo e ripetizioni	299
Testi alternati	307
Polifonia con testo in comune	308
2.1.3 Strofe	310
Aggiungere i numeri di strofa	310
Aggiungere le dinamiche alle strofe	311
Aggiungere i nomi dei cantanti alle strofe	312
Strofe con ritmi diversi	312
Stampare le strofe alla fine	315
Stampare le strofe alla fine in molteplici colonne	316
2.1.4 Canzoni	318
Riferimenti per canzoni	318
Canzonieri	319
2.1.5 Musica corale	319
Riferimenti per musica corale	319
Struttura di una partitura corale	320
2.1.6 Opera e musical	322
Riferimenti per opera e musical	323
Nomi dei personaggi	323
Suggerimenti musicali	325
Musica parlata	329
Dialogo sopra la musica	329
2.1.7 Canti salmi e inni	330
Riferimenti per canti e salmi	330
Impostare un canto	331
Salmi	337
Misure parziali nei motivi degli inni	340
2.1.8 Musica vocale antica	342
2.2 Tastiera e altri strumenti multirigo	343
2.2.1 Notazione comune per tastiere	343
Riferimenti per tastiere	343
Modifica manuale del rigo	344
Modifica automatica del rigo	347
Linee del cambio rigo	348
2.2.2 Pianoforte	351
Pedali del pianoforte	351
2.2.3 Fisarmonica	352
Simboli di discanto	352
2.2.4 Arpa	353
Riferimenti per arpe	354
Pedali dell'arpa	354
2.3 Strumenti a corde senza tasti	355
2.3.1 Notazione comune per strumenti a corde senza tasti	355
Riferimenti per archi senza tasti	355
Indicazioni di arcata	356
Armonici	356
Pizzicato alla Bartók	357

2.4	Strumenti a corde con tasti	358
2.4.1	Notazione comune per strumenti a corde con tasti	358
	Riferimenti per strumenti a corde con tasti	358
	Indicazioni di numero di corda	359
	Intavolature predefinite	360
	Intavolature personalizzate	378
	Diagrammi dei tasti	382
	Diagrammi dei tasti predefiniti	392
	Diagrammi dei tasti automatici	403
	Diteggiature della mano destra	405
2.4.2	Chitarra	407
	Indicazione di posizione e barré	407
	Indicazione di armonici e note smorzate	408
	Indicazione di power chord	409
2.4.3	Banjo	410
	Intavolature per banjo	410
2.4.4	Liuto	411
	Intavolature per liuto	411
2.5	Percussioni	412
2.5.1	Notazione comune per le percussioni	412
	Riferimenti per percussioni	412
	Notazione di base per percussioni	412
	Rulli di tamburo	413
	Percussioni a suono determinato	413
	Righi delle percussioni	414
	Righi delle percussioni personalizzati	416
	Ghost note	419
2.6	Strumenti aerofoni	419
2.6.1	Notazione comune per gli strumenti aerofoni	419
	Riferimenti per strumenti aerofoni	419
	Diteggiature	420
2.6.2	Cornamusa	422
	Definizioni per cornamusa	422
	Esempio per cornamusa	423
2.6.3	Legni	424
	2.6.3.1 Diagrammi per legni	424
2.7	Notazione per accordi	431
2.7.1	Modalità accordo	431
	Panoramica sulla modalità accordo	432
	Accordi comuni	432
	Accordi estesi e alterati	434
	Inversioni di accordo e voicing	436
2.7.2	Visualizzazione accordi	437
	Stampa dei nomi degli accordi	437
	Personalizzazione dei nomi degli accordi	439
2.7.3	Basso continuo	444
	Introduzione al basso continuo	445
	Inserimento del basso continuo	446
	Visualizzazione del basso continuo	448
2.8	Musica contemporanea	451
2.8.1	Altezza e armonica nella musica contemporanea	451
	Riferimenti per altezza e armonia nella musica contemporanea	451

Notazione microtonale	451
Armature di chiave e armonia contemporanee	451
2.8.2 Approcci contemporanei al ritmo	451
Riferimenti per approcci contemporanei al ritmo	451
Gruppi irregolari nella musica contemporanea	451
Indicazioni di tempo contemporanee	451
Notazione polimetrica estesa	451
Travature nella musica contemporanea	451
Stanghette nella musica contemporanea	451
2.8.3 Notazione grafica	452
2.8.4 Tecniche di arrangiamento contemporaneo	453
2.8.5 Nuove tecniche strumentali	453
2.8.6 Letture consigliate e partiture rilevanti	453
Libri e articoli sulla notazione musicale contemporanea	453
Partiture e esempi musicali	454
2.9 Notazione antica	454
2.9.1 Panoramica degli stili supportati	455
2.9.2 Notazione antica – funzionalità comuni	456
Contesti predefiniti	456
Legature	456
Custodes	457
2.9.3 Scrivere la musica mensurale	458
Contesti mensurali	458
Chiavi mensurali	459
Indicazioni di tempo mensurali	460
Teste di nota mensurali	460
Code mensurali	461
Pause mensurali	462
Alterazioni e armature di chiave mensurali	463
Alterazioni suggerite (<i>musica ficta</i>)	463
Legature mensurali bianche	464
2.9.4 Scrivere il canto gregoriano	465
Contesti del canto gregoriano	465
Chiavi gregoriane	466
Alterazioni e armature di chiave gregoriane	467
Divisiones	467
Segni di articolazione gregoriani	468
Punti di aumentazione (<i>morae</i>)	469
Legature di neumi quadrati gregoriani	469
2.9.5 Scrivere in notazione quadrata di Kiev	475
Contesti della notazione di Kiev	475
Chiavi della notazione di Kiev	476
Note della notazione di Kiev	476
Alterazioni della notazione di Kiev	477
Stanghetta della notazione di Kiev	477
Melismi della notazione di Kiev	477
2.9.6 Lavorare con la musica antica – scenari e soluzioni	478
Incipit	478
Formattazione mensurstriche	479
Trascrivere il canto gregoriano	480
Antico e moderno da un unico sorgente	483
2.10 World music	485

2.10.1	Notazione comune per la musica non occidentale	485
	Estensione dei sistemi di notazione e di accordatura	485
2.10.2	Musica araba	486
	Referenze per la musica araba	486
	Nomi delle note in arabo	487
	Armature di chiave arabe	487
	Indicazioni di tempo arabe	489
	Esempio di musica araba	490
	Lecture complementari sulla musica araba	491
2.10.3	Musica classica turca	491
	Riferimenti per la musica classica turca	491
	Nomi delle note in turco	491
3	Input e output	493
3.1	Struttura dell'input	493
3.1.1	Struttura di una partitura	493
3.1.2	Molteplici partiture in un libro	494
3.1.3	Molteplici file di output da un unico file di input	495
3.1.4	Nomi dei file di output	496
3.1.5	Struttura del file	497
3.2	Titoli e intestazioni	499
3.2.1	Creazione di titoli intestazioni e piè di pagina	499
	Come funzionano i titoli	499
	Formattazione predefinita dei titoli delle parti e dei brani	502
	Formattazione predefinita delle intestazioni e dei piè di pagina	505
3.2.2	Titoli intestazioni e piè di pagina personalizzati	506
	Titoli personalizzati	506
	Formattazione personalizzata dei titoli	507
	Formattazione personalizzata di intestazioni e piè di pagina	510
3.2.3	Creazione di metadati per i file di output	511
3.2.4	Creazione di note a piè di pagina	512
	Note a piè di pagina nelle espressioni musicali	512
	Note a piè di pagina nel testo separato	518
3.2.5	Riferimento ai numeri di pagina	520
3.2.6	Indice	521
3.3	Lavorare coi file di input	524
3.3.1	Inclusione di file LilyPond	524
3.3.2	Edizioni diverse da un unico sorgente	525
	Uso delle variabili	526
	Uso delle etichette	527
	Impostazioni globali	531
3.3.3	Caratteri speciali	531
	Codifica del testo	531
	Unicode	532
	Alias ASCII	533
3.4	Controllo dell'output	534
3.4.1	Estrarre frammenti musicali	534
3.4.2	Saltare la musica già corretta	534
3.4.3	Formati di output alternativi	535
	Output SVG	535
3.4.4	Cambiare il tipo di carattere della notazione	535

3.5 Creazione dell'output MIDI.....	537
3.5.1 Notazione supportata nel MIDI.....	537
3.5.2 Notazione non supportata nel MIDI.....	538
3.5.3 Il blocco MIDI.....	539
3.5.4 Gestione delle dinamiche nel MIDI.....	539
Dinamiche nel MIDI.....	540
Impostazione del volume MIDI.....	540
Impostazione delle proprietà del blocco MIDI.....	543
3.5.5 Uso degli strumenti MIDI.....	544
3.5.6 Uso delle ripetizioni nel MIDI.....	545
3.5.7 Mappatura dei canali MIDI.....	545
3.5.8 Proprietà di contesto per gli effetti MIDI.....	548
3.5.9 Miglioramento dell'output MIDI.....	549
Lo script articulate.....	549
3.6 Estrazione dell'informazione musicale.....	549
3.6.1 Mostrare la notazione LilyPond.....	550
3.6.2 Mostrare le espressioni musicali Scheme.....	550
3.6.3 Salvare eventi musicali in un file.....	550
4 Gestione dello spazio.....	552
4.1 Formattazione della pagina.....	552
4.1.1 Il blocco \paper.....	552
4.1.2 Formato carta e ridimensionamento automatico.....	553
Impostare il formato carta.....	553
Ridimensionamento automatico al formato carta.....	554
4.1.3 Variabili \paper della spaziatura verticale fissa.....	555
4.1.4 Variabili \paper della spaziatura verticale flessibile.....	555
Struttura delle liste associative flessibili della spaziatura verticale.....	556
Elenco delle variabili \paper flessibili della spaziatura verticale.....	557
4.1.5 Variabili \paper della spaziatura orizzontale.....	557
Variabili \paper per larghezze e margini.....	558
Variabili \paper per la modalità due pagine per foglio.....	559
Variabili \paper per spostamenti e indentazioni.....	560
4.1.6 Altre variabili di \paper.....	560
Variabili di \paper per l'interruzione di linea.....	560
Variabili di \paper per l'interruzione di pagina.....	561
Variabili di \paper per la numerazione delle pagine.....	562
Svariate variabili di \paper.....	562
4.2 Formattazione della partitura.....	563
4.2.1 Il blocco \layout.....	563
4.2.2 Impostare la dimensione del rigo.....	565
4.3 Interruzioni.....	567
4.3.1 Interruzioni di linea.....	567
4.3.2 Interruzioni di pagina.....	570
Interruzione di pagina manuale.....	570
Interruzione di pagina ottimale.....	571
Interruzione di pagina minimale.....	572
Interruzione di pagina di una pagina.....	572
Interruzione di pagina su una linea.....	572
Interruzione di pagina su una linea con altezza automatica.....	572
Voltata di pagina ottimale.....	572

4.4	Spaziatura verticale.....	574
4.4.1	Spaziatura verticale flessibile all'interno dei sistemi.....	574
	Proprietà della spaziatura dentro un sistema	574
	Spaziatura dei righi non raggruppati	577
	Spaziatura dei righi raggruppati	578
	Spaziatura delle linee che non sono righi	579
4.4.2	Posizionamento esplicito di righi e sistemi	581
4.4.3	Elusione delle collisioni verticali	589
4.5	Spaziatura orizzontale	590
4.5.1	Panoramica sulla spaziatura orizzontale	590
4.5.2	Nuova spaziatura nel corso di un brano	592
4.5.3	Modifica della spaziatura orizzontale	593
	Allungamento uniforme dei gruppi irregolari	594
	Spaziatura rigorosa della nota	594
4.5.4	Larghezza della linea.....	595
4.5.5	Notazione proporzionale	595
4.6	Riduzione del numero di pagine di una partitura	601
4.6.1	Visualizzare la spaziatura	601
4.6.2	Modificare la spaziatura	602
5	Modifica delle impostazioni predefinite.....	605
5.1	Contesti di interpretazione.....	605
5.1.1	Tutto sui contesti.....	605
	Definizioni di output – gerarchia dei contesti	605
	Score – il padre di tutti i contesti	606
	Contesti del livello superiore – contenitori di righi	606
	Contesti del livello intermedio – righi	606
	Contesti del livello inferiore – voci	607
5.1.2	Creazione e citazione di un contesto.....	607
5.1.3	Conservazione di un contesto.....	611
5.1.4	Modifica dei componenti aggiuntivi di un contesto.....	613
5.1.5	Modifica delle impostazioni predefinite di un contesto	615
	Modifica di tutti i contesti dello stesso tipo.....	615
	Modifica di un solo contesto specifico	618
	Ordine di precedenza	620
5.1.6	Definizione di nuovi contesti	620
5.1.7	Ordine di disposizione dei contesti	623
5.2	Come funziona la Guida al funzionamento interno.....	625
5.2.1	Navigazione nella guida al programma	625
5.2.2	Interfacce di formattazione.....	625
5.2.3	Determinazione della proprietà del grob.....	627
5.3	Modifica delle proprietà.....	627
5.3.1	Panoramica sulla modifica delle proprietà	628
5.3.2	Il comando <code>\set</code>	628
5.3.3	Il comando <code>\override</code>	630
5.3.4	Il comando <code>\tweak</code>	632
5.3.5	<code>\set</code> vs. <code>\override</code>	634
5.3.6	Il comando <code>\offset</code>	634
5.3.7	Modifica delle liste associative	639
5.4	Proprietà e concetti utili.....	641
5.4.1	Modalità di inserimento.....	641

5.4.2	Direzione e posizionamento	643
	Indicatori di direzione delle articolazioni	643
	La proprietà direction	643
5.4.3	Distanze e misurazioni	644
5.4.4	Dimensioni	645
5.4.5	Proprietà del simbolo del rigo	645
5.4.6	Estensori	645
	Uso di spanner-interface	646
	Uso di line-spanner-interface	649
5.4.7	Visibilità degli oggetti	652
	Soppressione dello stampo	652
	Rendere gli oggetti trasparenti	652
	Dipingere gli oggetti di bianco	653
	Uso di break-visibility	654
	Considerazioni speciali	655
5.4.8	Stili della linea	658
5.4.9	Rotazione degli oggetti	659
	Rotazione degli oggetti della formattazione	659
	Rotazione del testo	660
5.5	Ritocchi avanzati	660
5.5.1	Allineamento degli oggetti	660
	Impostazione diretta di X-offset e Y-offset	661
	Uso di side-position-interface	661
	Uso di self-alignment-interface	662
	Uso di break-alignable-interface	663
5.5.2	Raggruppamento verticale dei grob	665
5.5.3	Modifica degli stampi	665
5.5.4	Modifica delle forme	666
	Modifica di legature di valore e di portamento	666
5.5.5	Modifica degli estensori spezzati	671
	Uso di \alterBroken	671
5.5.6	Contenitori unpure-pure	672
5.6	Uso delle funzioni musicali	674
5.6.1	Sintassi della funzione di sostituzione	674
5.6.2	Esempi della funzione di sostituzione	675

Appendice A Tabelle del manuale della notazione..... 678

A.1	Grafico dei nomi degli accordi	678
A.2	Modificatori degli accordi comuni	678
A.3	Accordature predefinite	681
A.4	Diagrammi degli accordi predefiniti	682
	Diagrammi per chitarra	682
	Diagrammi per ukulele	684
	Diagrammi per mandolino	685
A.5	Formati carta predefiniti	687
A.6	Strumenti MIDI	689
A.7	Elenco dei colori	690
A.8	Il font Emmentaler	693
	Glifi della chiave	693
	Glifi delle indicazioni di tempo	693
	Glifi dei numeri	693

Glifi del simbolo di basso numerato	695
Glifi delle alterazioni	695
Glifi delle teste di nota predefinite	695
Glifi delle teste di nota speciali	696
Glifi delle teste di nota a forma variabile	696
Glifi delle pause	698
Glifi delle code	699
Glifi dei punti	700
Glifi delle dinamiche	700
Glifi dei segni	700
Glifi delle teste a forma di freccia	701
Glifi delle estremità delle parentesi	702
Glifi dei pedali	702
Glifi della fisarmonica	702
Glifi delle legature di valore	702
Glifi della notazione vaticana	702
Glifi della notazione medicea	703
Glifi Hufnagel	703
Glifi della notazione mensurale	704
Glifi della notazione neomensurale	706
Glifi Petrucci	706
Glifi Solesmes	707
Glifi della notazione di Kiev	707
A.9 Stili delle teste di nota	707
A.10 Gruppi di glifi di alterazione	708
A.11 Stili della chiave	709
Chiavi standard	709
Chiave del rigo della percussione	710
Chiavi del rigo dell'intavolatura	710
Chiavi della musica antica	710
A.12 Comandi per <i>markup</i>	713
A.12.1 Font markup	713
A.12.2 Markup for text alignment	725
A.12.3 Graphical markup	744
A.12.4 Markup for music and musical symbols	755
A.12.5 Conditional markup	767
A.12.6 Instrument-specific markup	768
A.12.7 Accordion registers	773
A.12.8 Other markup commands	777
A.13 Comandi per una lista di <i>markup</i>	789
A.14 Elenco dei caratteri speciali	792
A.15 Elenco delle articolazioni	793
Articolazioni	793
Ornamenti	794
Punti coronati	794
Segni specifici per strumento	794
Segni di ripetizione	795
Segni antichi	795
A.16 Elenco dei respiri	796
A.17 Note percussive	796
A.18 Valori predefiniti di <i>outside-staff-priority</i>	798
A.19 Glossario tecnico	799

alist	799
callback	799
closure	799
glyph	799
grob	800
immutable	800
interface	800
lexer	800
mutable	801
output-def	801
parser	801
parser variable	801
prob	802
smob	802
spanner	802
stencil	802
A.20 Funzioni musicali disponibili	802
A.21 Identificatori delle modifiche di contesto	821
A.22 Convenzioni per i nomi	821
A.23 Tipi di predicati predefiniti	822
A.23.1 R5RS primary predicates	822
A.23.2 R5RS secondary predicates	823
A.23.3 Guile predicates	823
A.23.4 LilyPond scheme predicates	823
A.23.5 LilyPond exported predicates	824
Appendice B Schema riassuntivo	826
Appendice C GNU Free Documentation License	829
Appendice D Indice	836

1 Notazione musicale

Questo capitolo spiega come creare la notazione musicale.

1.1 Altezze

dolce e molto legato

Red. * Red. * Red. *

Red. *

Questa sezione tratta il modo in cui si determina l'altezza delle note. Occorre considerare tre aspetti: input, modifica e output.

1.1.1 Inserimento delle altezze

Questa sezione spiega come indicare l'altezza delle note. Ci sono due modi di collocare le note in una determinata ottava: il modo assoluto e il modo relativo. Nella maggioranza dei casi il modo relativo è più funzionale.

Ottava assoluta

Le altezze, se non si adotta una lingua diversa, sono scritte in notazione olandese, che usa le lettere minuscole dalla a (La) alla g (Sol). Le note c (Do) e b (Si) vengono scritte un'ottava sotto il Do centrale.

```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```



Si possono indicare altre ottave con l'apice singolo (') o la virgola (,). Ogni ' alza l'altezza di un'ottava; ogni , abbassa l'altezza di un'ottava.

```
{
```

```

\clef treble
c'4 e' g' c''
c'4 g b c'
\clef bass
c,4 e, g, c
c,4 g,, b,, c,
}

```



I normali segni di ottava possono essere inseriti una sola volta se si imposta un'altezza di riferimento dopo `\fixed` e prima della musica. Le altezze inserite in un blocco `\fixed` hanno bisogno dei segni ' o , solo quando si trovano sopra o sotto l'ottava dell'altezza di riferimento.

```

{
  \fixed c' {
    \clef treble
    c4 e g c'
    c4 g, b, c
  }
  \clef bass
  \fixed c, {
    c4 e g c'
    c4 g, b, c
  }
}

```



Le altezze dell'espressione musicale che segue `\fixed` non cambiano se racchiuse da un blocco `\relative`, che vedremo tra poco.

Vedi anche

Glossario musicale: Sezione “Nomi delle altezze” in *Glossario Musicale*.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Ottava relativa

L'inserimento delle note con l'ottava assoluta costringe a specificare l'ottava di ogni singola nota. Al contrario, se si usa l'ottava relativa, ogni ottava è determinata dall'ultima nota: se si cambia l'ottava di una nota, cambieranno anche le ottave di tutte le note successive.

La modalità relativa deve essere impostata in modo esplicito col comando `\relative`:

```
\relative altezza_di_riferimento espressione_musicale
```

In modalità relativa ogni nota è collocata il più vicino possibile a quella precedente. Questo significa che l'ottava di ogni altezza all'interno di `espressione_musicale` viene calcolata nel modo seguente:

- In assenza di segni di cambiamento d'ottava, l'ottava di un'altezza viene calcolata in modo che l'intervallo con la nota precedente sia inferiore a una quinta. Tale intervallo è determinato senza considerare gli accidenti.

- Si può aggiungere un segno di cambiamento d'ottava ' o , per collocare l'altezza di una nota all'ottava superiore o inferiore a quella di riferimento.
- È possibile usare più di un segno di cambiamento d'ottava. Per esempio, '' e ,, modificano l'altezza di due ottave.
- L'altezza della prima nota è relativa a *altezza_di_riferimento*. *altezza_di_riferimento* è specificato nel modo di ottava assoluta. Quale di queste opzioni è la più conveniente?

un'ottava del c

Identificare il Do centrale con c' è molto semplice, quindi trovare le ottave del c (Do) sarà altrettanto semplice. Se la musica inizia con gis sopra c'', si scriverà qualcosa simile a `\relative { gis'' ... }`

un'ottava della prima nota

Scrivere `\relative { gis'' ... }` è un modo semplice per determinare l'altezza assoluta della prima nota dell'espressione musicale.

nessuna altezza di riferimento esplicita

La forma `\relative {gis'' ... }` è una versione compatta dell'opzione precedente: la prima nota dentro l'espressione musicale è scritta come altezza assoluta. In questo caso equivale a scegliere f come altezza di riferimento.

La documentazione di solito usa l'ultima opzione.

Ecco il modo relativo in azione:

```
\relative {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```



I segni di cambiamento d'ottava si impiegano per gli intervalli più ampi di quello di quarta:

```
\relative {
  c' g c f,
  c' a, e'' c
}
```



Una sequenza di note senza segni di ottava può tuttavia comprendere intervalli di grande estensione:

```
\relative {
  c f b e
  a d g c
}
```



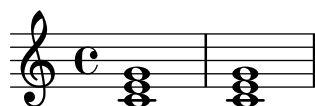

Nel caso di blocchi `\relative` annidati, il blocco `\relative` più interno inizia con la propria altezza di riferimento, indipendentemente dal `\relative` più esterno.

```
\relative {
  c' d e f
  \relative {
    c'' d e f
  }
}
```



`\relative` non ha effetto sui blocchi `\chordmode`.

```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```



`\relative` non può essere inserito all'interno dei blocchi `\chordmode`.

La musica all'interno di un blocco `\transpose` è considerata in notazione d'ottava assoluta, a meno che non sia incluso il blocco `\relative`.

```
\relative {
  d' e
  \transpose f g {
    d e
    \relative {
      d' e
    }
  }
}
```



Se l'elemento precedente è un accordo, il posizionamento dell'ottava della nota o dell'accordo che segue è riferito alla prima nota dell'accordo stesso. All'interno degli accordi la nota successiva è sempre relativa a quella precedente. Esaminate con attenzione l'esempio seguente, e in particolare le note c.

```
\relative {
```

```

c'
<c e g>
<c' e g'>
<c, e, g''>
}

```



Come spiegato sopra, il riferimento delle altezze a un'ottava è calcolato in base ai soli nomi delle note, senza considerare le alterazioni. Dunque un Mi doppio diesis che segue un Si verrà posizionato sopra, mentre un Fa doppio bemolle sarà posizionato sotto. In altre parole, un intervallo di quarta aumentata due volte viene considerato più piccolo di una quinta diminuita due volte, indipendentemente dal numero di semitoni contenuto in ogni intervallo.

```

\relative {
  c''2 fis
  c2 ges
  b2 eisis
  b2 feses
}

```

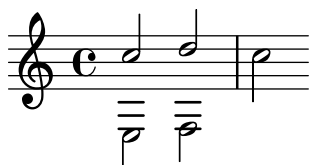


In situazioni complesse, può essere utile cambiare l'altezza di riferimento senza tener conto di ciò che è successo prima, usando `\resetRelativeOctave`:

```

\relative {
  <<
    { c''2 d }
    \\
    { e,,2 f }
  >>
  \resetRelativeOctave c''
  c2
}

```



Vedi anche

Glossario musicale: Sezione “quinta” in *Glossario Musicale*, Sezione “intervallo” in *Glossario Musicale*, Sezione “Nomi delle altezze” in *Glossario Musicale*.

Guida alla notazione: [Controlli di ottava], pagina 10.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “RelativeOctaveMusic” in *Guida al Funzionamento Interno*.

Alterazioni

Nota: I nuovi utenti sono talvolta confusi dalla gestione delle alterazioni e delle armature di chiave. In LilyPond i nomi delle note specificano le altezze; le armature e le chiavi determinano come queste altezze debbano essere rappresentate. Una nota non alterata come `c` significa ‘Do naturale’, indipendentemente dall’armatura o dalla chiave. Per maggiori informazioni si veda Sezione “Altezze e armature di chiave” in *Manuale di Apprendimento*.

Nella modalità di notazione predefinita un *diesis* si ottiene aggiungendo `is` al nome della nota, un *bemolle* aggiungendo `es`. Come potete immaginare, un *doppio diesis* o *doppio bemolle* si ottengono aggiungendo `isis` o `eses`. Questa sintassi è desunta dalla notazione olandese. Per usare altri nomi per le alterazioni, si veda [Nomi delle note in altre lingue], pagina 8.

```
\relative c'' { ais1 aes aisis aeses }
```



Un’altezza naturale è indicata con il semplice nome della nota; non è richiesto alcun suffisso. Un segno di bequadro apparirà automaticamente quando occorre cancellare l’armatura di chiave o l’effetto di un’alterazione precedente.

```
\relative c'' { a4 aes a2 }
```



È possibile indicare alterazioni di quarti di tono. Ecco una serie di Do con altezza crescente:

```
\relative c'' { ceseh1 ces ceh c cih cis cisih }
```



Di norma le alterazioni vengono mostrate automaticamente, ma è possibile anche inserirle manualmente. Si può forzare l’inserimento di un’alterazione di sicurezza aggiungendo il punto esclamativo `!` dopo l’altezza. Un’alterazione di cortesia (ovvero un’alterazione compresa tra parentesi) si ottiene aggiungendo il punto interrogativo `?` dopo l’altezza.

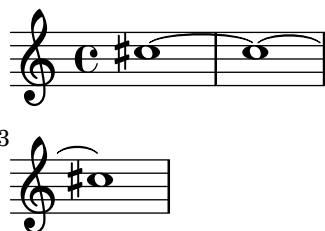
```
\relative c'' { cis cis cis! cis? c c c! c? }
```



Se una nota è prolungata attraverso una legatura di valore, l’alterazione viene ripetuta solo all’inizio di un nuovo sistema:

```
\relative c'' {
  cis1~ 1~
  \break
  cis
}
```

}



Frammenti di codice selezionati

Nascondere le alterazioni delle note con legatura di valore all'inizio di un nuovo sistema

Questo frammento mostra come nascondere le alterazioni delle note unite alla figura precedente mediante una legatura di valore all'inizio di un nuovo sistema

```
\relative c' ' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```



Impedire l'inserimento automatico dei bequadri supplementari

Secondo le norme tipografiche tradizionali, un segno di bequadro viene inserito prima di un diesis o di un bemolle se un precedente doppio diesis o bemolle sulla stessa nota è cancellato. Per cambiare questo comportamento e seguire la pratica contemporanea, si imposta la proprietà `extraNatural` su `f` (falso) nel contesto `Staff`.

```
\relative c' ' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Vedi anche

Glossario musicale: Sezione “diesis” in *Glossario Musicale*, Sezione “bemolle” in *Glossario Musicale*, Sezione “doppio diesis” in *Glossario Musicale*, Sezione “doppio bemolle” in *Glossario Musicale*, Sezione “Nomi delle altezze” in *Glossario Musicale*, Sezione “quarto di tono” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Altezze e armature di chiave” in *Manuale di Apprendimento*.

Guida alla notazione: [Alterazioni automatiche], pagina 29, [Alterazioni suggerite (*musica ficta*)], pagina 463, [Nomi delle note in altre lingue], pagina 8.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Accidental-engraver” in *Guida al Funzionamento Interno*, Sezione “Accidental” in *Guida al Funzionamento Interno*, Sezione “Accidental-Cautionary” in *Guida al Funzionamento Interno*, Sezione “accidental-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Poiché non esistono standard universalmente accettati per indicare le alterazioni di quarto di tono, i simboli impiegati da LilyPond non si riferiscono ad alcuno standard.

Nomi delle note in altre lingue

LilyPond comprende insiemi predefiniti di nomi di note e alterazioni in altre lingue. La scelta della lingua si fa solitamente all’inizio del file; l’esempio seguente è scritto in notazione italiana:

```
\language "italiano"
```

```
\relative {
  do' re mi sib
}
```



Le lingue disponibili e i tipi di notazione che definiscono sono:

Lingua	Nomi delle note
nederlands	c d e f g a bes b
català o catalan	do re mi fa sol la sib si
deutsch	c d e f g a b h
english	c d e f g a bf/b-flat b
español o espanol	do re mi fa sol la sib si
français	do ré/re mi fa sol la sib si
italiano	do re mi fa sol la sib si
norsk	c d e f g a b h
português o portugues	do re mi fa sol la sib si
suomi	c d e f g a b h
svenska	c d e f g a b h
vlaams	do re mi fa sol la sib si

Oltre ai nomi delle note, anche i suffissi per le alterazioni possono variare a seconda della lingua adottata:

Lingua	diesis	bemolle	doppio diesis	doppio bemolle
nederlands	is	es	isis	eses
català o catalan	d/s	b	dd/ss	bb
deutsch	is	es	isis	eses
english	s/-sharp	f/-flat	ss/x/-sharpsharp	ff/-flatflat

español o espanol	s	b	ss/x	bb
français	d	b	dd/x	bb
italiano	d	b	dd	bb
norsk	iss/is	ess/es	ississ/isis	essess/eses
português o portugues	s	b	ss	bb
suomi	is	es	isis	eses
svenska	iss	ess	ississ	essess
vlaams	k	b	kk	bb

In olandese, norvegese e finlandese *aes* viene contratto in *as*; in olandese e norvegese, tuttavia, entrambe le forme sono accettate da LilyPond. Lo stesso vale per *es* e *ees*, *aeses* e *ases*, e infine *eeses* e *eses*.

In tedesco e finlandese, LilyPond fornisce anche la forma più frequente *asas* per *ases*.

```
\relative c'' { a2 as e es a ases e eses }
```



In alcune forme musicali vengono usati i microtoni, le cui alterazioni sono frazioni di un ‘normale’ diesis o bemolle. La seguente tabella elenca i nomi delle note per le alterazioni di un quarto di tono; i prefissi *semi-* e *sesqui-* significano rispettivamente ‘metà’ e ‘uno e mezzo’.

Lingua	semi-diesis	semi-bemolle	sesqui-diesis	sesqui-bemolle
nederlands	ih	eh	isih	eseh
català o catalan	qd/qs	qb	tqd/tqs	tqb
deutsch	ih	eh	isih	eseh
english	qs	qf	tqs	tqf
español o espanol	cs	cb	tcs	tcb
français	sd	sb	dsd	bsb
italiano	sd	sb	dsd	bsb
norsk	ih	eh	issih/isih	esseh/eseh
português o portugues	sqt	bqt	stqt	btqt
suomi	ih	eh	isih	eseh
svenska	ih	eh	issih	esseh
vlaams	hk	hb	khk	bhb

In tedesco, esistono simili contrazioni dei nomi per i microtoni così come nelle normali altezze descritte sopra.

```
\language "deutsch"
```

```
\relative c'' { asah2 eh aih eisih }
```



Gran parte delle lingue presentate qui sono comunemente associate alla musica classica occidentale, nota anche come *Common Practice Period*. Sono tuttavia supportati anche altezze e

sistemi di accordatura alternativi: si veda Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

Vedi anche

Glossario musicale: Sezione “Nomi delle altezze” in *Glossario Musicale*, Sezione “Periodo di pratica comune” in *Glossario Musicale*.

Guida alla notazione: Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

File installati: scm/define-note-names.scm.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

1.1.2 Modifica di più altezze

Questa sezione tratta il modo di modificare le altezze delle note.

Controlli di ottava

In modalità relativa è facile dimenticare un segno di cambiamento d’ottava. I controlli di ottava permettono di rilevare questi errori più facilmente: infatti, generano un avviso e correggono l’ottava se una nota si trova in un’ottava diversa dal previsto.

Per controllare l’ottava di una nota, occorre specificare l’ottava assoluta dopo il simbolo =. Questo esempio genererà un avviso (e cambierà l’altezza) perché la seconda nota è l’ottava assoluta d' ' invece di d', come indicato dalla correzione di ottava.

```
\relative {
  c' '2 d='
  e2 f
}
```



L’ottava in cui si trovano le note può essere controllata anche col comando `\octaveCheck altezza_di_controllo`. L’`altezza_di_controllo` è specificata in modo assoluto. Questo comando controlla che l’intervallo tra la nota precedente e l’`altezza_di_controllo` sia compresa in una quinta (ovvero secondo il normale calcolo della modalità relativa). Se il controllo fallisce, compare un avviso. Benché la nota precedente non sia modificata, le note successive sono relative al valore corretto.

```
\relative {
  c' '2 d
  \octaveCheck c'
  e2 f
}
```



Nelle due battute che seguono, il primo e il terzo `\octaveCheck` falliscono, mentre il secondo non fallisce.

```
\relative {
  c' '4 f g f
```

```

c4
\octaveCheck c'
f
\octaveCheck c'
g
\octaveCheck c'
f
}

```



Vedi anche

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “RelativeOctaveCheck” in *Guida al Funzionamento Interno*.

Trasposizione

Un'espressione musicale può essere trasposta con `\transpose`. La sintassi è

```
\transpose altezza_di_partenza altezza_di_arrivo espressione_musicale
```

Significa che *espressione_musicale* viene trasposto dell'intervallo compreso tra le altezze *altezza_di_partenza* e *altezza_di_arrivo*: qualsiasi nota che presenti un'altezza corrispondente all'*altezza_di_partenza* viene modificata in *altezza_di_arrivo*, e qualsiasi altra nota viene trasposta dello stesso intervallo. Entrambe le altezze sono inserite in modalità assoluta.

Nota: La musica all'interno di un blocco `\transpose` è assoluta a meno che il blocco non includa un `\relative`.

Prendiamo come esempio un brano scritto in Re maggiore. Possiamo trasportarlo in Mi maggiore; si noti come anche l'armatura di chiave venga trasposta automaticamente.

```

\transpose d e {
  \relative {
    \key d \major
    d'4 fis a d
  }
}

```



Se una parte scritta in Do (l'*intonazione reale* abituale) deve essere suonata su un clarinetto in La (per il quale un La viene rappresentato da un Do e dunque suona una terza minore più basso), la trasposizione sarà ottenuta con:

```

\transpose a c' {
  \relative {
    \key c \major
    c'4 d e g
  }
}

```


}



Si noti che `\key c \major` è specificato esplicitamente. Se non si specifica un'armatura di chiave, le note verranno trasposte ma non apparirà alcuna armatura.

`\transpose` fa distinzione tra altezze enarmoniche: sia `\transpose c cis` che `\transpose c des` traspongono un brano di un semitono più alto. La prima versione mostrerà i diesis e le note rimarranno sullo stesso grado della scala, mentre la seconda versione mostrerà i bemolle sul grado superiore della scala.

```
music = \relative { c' d e f }
\new Staff {
  \transpose c cis { \music }
  \transpose c des { \music }
}
```



`\transpose` può essere usato anche in un altro modo, ovvero per inserire note scritte per uno strumento traspositore. Gli esempi precedenti mostrano come inserire altezze in Do (o *intonazione reale*) e mostrare le note di uno strumento traspositore, ma è possibile anche il contrario: per esempio, se da un insieme di parti strumentali si volesse ricavare una partitura per il direttore. Così, per inserire la parte per una tromba in Si bemolle che inizia con un Mi (intonazione reale Re), si può scrivere:

```
musicInBflat = { e4 ... }
\transpose c bes, \musicInBflat
```

Per stampare questa musica in Fa (ad esempio per riarrangiarla per corno) si può avvolgere la musica esistente in un altro `\transpose`:

```
musicInBflat = { e4 ... }
\transpose f c' { \transpose c bes, \musicInBflat }
```

Per maggiori informazioni sugli strumenti traspositori, si veda [Trasporto strumentale], pagina 27.

Frammenti di codice selezionati

Transposing pitches with minimum accidentals (“smart” transpose)

This example uses some Scheme code to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In this case, the following rules apply:

- double accidentals should be removed
- b sharp → c
- e sharp → f
- c flat → b
- f flat → e

In this manner, the most natural enharmonic notes are chosen.

```
#(define (naturalize-pitch p)
  (let ((o (ly:pitch-octave p)))
```

```

;; `ly:pitch-alteration` returns quarter tone steps.
(a (* 4 (ly:pitch-alteration p)))
(n (ly:pitch-notename p)))
(cond
  ((and (> a 1)
        (or (eqv? n 6) (eqv? n 2))))
    (set! a (- a 2))
    (set! n (+ n 1)))
  ((and (< a -1)
        (or (eqv? n 0) (eqv? n 3))))
    (set! a (+ a 2))
    (set! n (- n 1))))
(cond
  ((> a 2)
   (set! a (- a 4))
   (set! n (+ n 1)))
  ((< a -2)
   (set! a (+ a 4))
   (set! n (- n 1))))
(when (< n 0)
  (set! o (- o 1))
  (set! n (+ n 7)))
(when (> n 6)
  (set! o (+ o 1))
  (set! n (- n 7)))
(ly:make-pitch o n (/ a 4)))

#(define (naturalize music)
  (let ((es (ly:music-property music 'elements))
        (e (ly:music-property music 'element))
        (p (ly:music-property music 'pitch)))
    (when (pair? es)
      (ly:music-set-property! music 'elements
                              (map naturalize es)))
    (when (ly:music? e)
      (ly:music-set-property! music 'element
                              (naturalize e)))
    (when (ly:pitch? p)
      (set! p (naturalize-pitch p))
      (ly:music-set-property! music 'pitch p))
    music))

naturalizeMusic =
#(define-music-function (m) (ly:music?)
  (naturalize m))

music = \relative c' { c4 d e g }

\new Staff {
  \transpose c ais { \music }
  \naturalizeMusic \transpose c ais { \music }
  \transpose c deses { \music }

```

```
\naturalizeMusic \transpose c deses { \music }
}
```



Retrogradazione

Un'espressione musicale può essere invertita in modo da produrre il proprio retrogrado:

```
music = \relative { c'8. ees16( fis8. a16 b8.) gis16 f8. d16 }

\new Staff {
  \music
  \retrograde \music
}
```



Problemi noti e avvertimenti

\retrograde è uno strumento piuttosto semplice. Dato che molti eventi sono ‘riflessi’ (come in uno specchio) invece che scambiati, le modifiche manuali e i modificatori direzionali per gli estensori di apertura devono essere aggiunti ai corrispondenti estensori di chiusura: ^ (deve essere terminato da ^), ogni \< o \cresc deve essere terminato da \! o \endcr, ogni \> o \decr deve essere terminato da \enddecr. Comandi e modifiche che cambiano le proprietà con un effetto duraturo causeranno facilmente delle sorprese.

Vedi anche

Guida alla notazione: [Inversione], pagina 14, [Trasposizioni modali], pagina 15, [Trasposizione], pagina 11.

Trasposizioni modali

In una composizione musicale basata su una scala, un motivo viene frequentemente trasportato in differenti modi. Può essere *trasposto* per iniziare in punti diversi della scala o può essere *invertito* rispetto a un punto cardine della scala. Può anche essere rovesciato per produrre il *retrogrado*, si veda [Retrogradazione], pagina 15.

Nota: Le note che non si trovano all'interno della scala definita non vengono trasformate.

Trasposizione modale

Un motivo può essere trasposto entro una certa scala con:

```
\modalTranspose altezza-di-partenza altezza-di-arrivo scala motif
```

Le note di *motif* vengono spostate, se all'interno della *scala*, del numero di gradi della scala dati dall'intervallo tra *altezza-di-arrivo* e *altezza-di-partenza*:

```
diatonicScale = \relative { c' d e f g a b }
motif = \relative { c'8 d e f g a b c }

\new Staff {
  \motif
  \modalTranspose c f \diatonicScale \motif
  \modalTranspose c b, \diatonicScale \motif
}
```



È possibile indicare una scala ascendente di qualsiasi lunghezza e con qualsiasi intervallo:

```
pentatonicScale = \relative { ges aes bes des ees }
motif = \relative { ees'8 des ges,4 <ges' bes,> <ges bes,> }
```

```
\new Staff {
  \motif
  \modalTranspose ges ees' \pentatonicScale \motif
}
```



Se usato con una scala cromatica, `\modalTranspose` ha un effetto simile a `\transpose`, con in più la possibilità di specificare i nomi delle note da usare:

```
chromaticScale = \relative { c' cis d dis e f fis g gis a ais b }
motif = \relative { c'8 d e f g a b c }
```

```
\new Staff {
  \motif
  \transpose c f \motif
  \modalTranspose c f \chromaticScale \motif
}
```



Inversione modale

Una sequenza di note può essere invertita all'interno di una data scala intorno a una determinata nota cardine e quindi trasposto, in un'unica operazione, con:

```
\modalInversion altezza-cardine altezza-di-arrivo scala motif
```

Le note di *motif* vengono spostate dello stesso numero di gradi dalla nota dell'*altezza-cardine* all'interno della *scala*, ma nella direzione opposta, e il risultato viene poi spostato all'interno della *scala* per il numero di gradi dato dall'intervallo tra *altezza-di-arrivo* e *altezza-cardine*.

Dunque, per invertire intorno a una particolare nota della scala, è necessario usare il medesimo valore per *altezza-cardine* e *altezza-di-arrivo*:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }
```

```
\new Staff {
  \motif
  \modalInversion fis' fis' \octatonicScale \motif
}
```



Per invertire intorno a una nota cardine posta tra altre due note, si inverte intorno a una della note e poi si traspone di un grado della scala. Le due note specificate possono essere interpretate come parentesi del punto cardine:

```
scale = \relative { c' g' }
motive = \relative { c' c g' c, }

\new Staff {
  \motive
  \modalInversion c' g' \scale \motive
}
```



L'operazione combinata di inversione e retrogradazione produce la retrogradazione inversa:

```
octatonicScale = \relative { ees' f fis gis a b c d }
motif = \relative { c'8. ees16 fis8. a16 b8. gis16 f8. d16 }

\new Staff {
  \motif
  \retrograde \modalInversion c' c' \octatonicScale \motif
}
```



Vedi anche

Guida alla notazione: [Inversione], pagina 14, [Retrogradazione], pagina 15, [Trasposizione], pagina 11.

1.1.3 Aspetto delle altezze

Questa sezione tratta il modo di modificare l'aspetto delle altezze delle note.

Chiave

Senza un comando esplicito, la chiave predefinita in LilyPond è la chiave di violino (o di *Sol*).

```
c'2 c'
```



Per cambiare la chiave si usa il comando `\clef` seguito dal nome della chiave. In tutti gli esempi seguenti viene mostrato il *Do centrale*.

```
\clef treble
c'2 c'
\clef alto
c'2 c'
\clef tenor
c'2 c'
```

Aggiungendo _8 o ^8 al nome della chiave, la sua adozione comporta il trasporto all'ottava rispettivamente inferiore o superiore, mentre _15 e ^15 traspongono di due ottave. È possibile usare altri numeri interi, se necessario. I nomi di chiave contenenti caratteri non alfabetici devono essere racchiusi tra virgolette

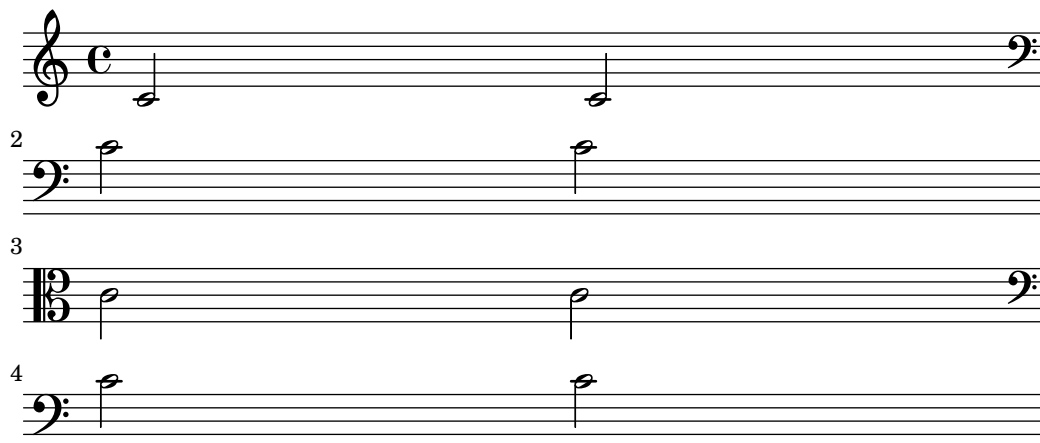
The first system of the musical score for 'The Rose Tree' consists of two staves. The top staff is in treble clef with a common time signature (C). It contains the melody for the first two lines of the song. The bottom staff is in bass clef and contains the accompaniment. The key signature has one flat (B-flat). The system ends with a double bar line. Above the first staff, the number '15' is written, and above the second staff, the number '5' is written.

```
\clef treble { c'2 c' } \break
\clef bass { c'2 c' } \break
```

```

\clef alto
  \set Staff.explicitClefVisibility = #end-of-line-invisible
  { c'2 c' } \break
  \unset Staff.explicitClefVisibility
\clef bass { c'2 c' } \break

```



Una chiave che è già stata visualizzata non viene ristampata se viene ripetuto lo stesso comando `\clef` e verrà quindi ignorata. Si può cambiare tale comportamento predefinito col comando `\set Staff.forceClef = ##t`.

```

\clef treble
c'1
\clef treble
c'1
\set Staff.forceClef = ##t
c'1
\clef treble
c'1

```



Per essere più precisi, non è proprio il comando `\clef` a stampare la chiave. Questo comando imposta o cambia una proprietà dell'incisore `Clef_engraver`, cui spetta la decisione se stampare o meno una chiave nel rigo attuale. La proprietà `forceClef` scavalca questa decisione localmente in modo da ristampare la chiave una volta.

Quando c'è un cambio di chiave manuale, il glifo della chiave cambiata sarà più piccolo del normale. Tale comportamento può essere modificato.

```

\clef "treble"
c'1
\clef "bass"
c'1
\clef "treble"
c'1
\override Staff.Clef.full-size-change = ##t
\clef "bass"
c'1
\clef "treble"
c'1

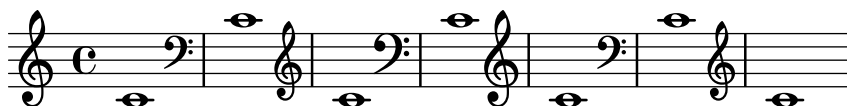
```



```

\revert Staff.Clef.full-size-change
\clef "bass"
c'1
\clef "treble"
c'1

```



Frammenti di codice selezionati

Modifiche manuali della proprietà della chiave

Cambiando il glifo della chiave, la sua posizione o l'ottavazione non cambia la posizione delle note successive nel rigo. Per far sì che le armature di chiave si trovino sulle linee del rigo corrette, bisogna specificare anche `middleCPosition`, con valori positivi o negativi che spostano il Do centrale rispettivamente su o giù in senso relativo alla linea centrale del rigo.

Per esempio, `\clef "treble_8"` equivale a impostare `clefGlyph`, `clefPosition` (che regola la posizione verticale della chiave), `middleCPosition` e `clefTransposition`. Viene stampata una chiave quando cambia una di queste proprietà, eccetto `middleCPosition`.

Gli esempi seguenti mostrano le possibilità date dall'impostazione manuale di tali proprietà. Sulla prima linea le modifiche manuali preservano il posizionamento relativo standard di chiavi e note, mentre sulla seconda linea non lo fanno.

```

{
  % The default treble clef.
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = "clefs.F"
  \set Staff.clefPosition = 2
  \set Staff.middleCPosition = 6
  \set Staff.middleCClefPosition = 6
  \key g \major
  c'1
  % The baritone clef.
  \set Staff.clefGlyph = "clefs.C"
  \set Staff.clefPosition = 4
  \set Staff.middleCPosition = 4
  \set Staff.middleCClefPosition = 4
  \key f \major
  c'1
  % The standard choral tenor clef.
  \set Staff.clefGlyph = "clefs.G"
  \set Staff.clefPosition = -2
  \set Staff.clefTransposition = -7
  \set Staff.middleCPosition = 1
  \set Staff.middleCClefPosition = 1
  \key f \major
  c'1
  % A non-standard clef.
  \set Staff.clefPosition = 0
}

```

```

\set Staff.clefTransposition = 0
\set Staff.middleCPosition = -4
\set Staff.middleCClefPosition = -4
\key g \major
c'1 \break

% The following clef changes do not preserve
% the normal relationship between notes, key signatures
% and clefs.
\set Staff.clefGlyph = "clefs.F"
\set Staff.clefPosition = 2
c'1
\set Staff.clefGlyph = "clefs.G"
c'1
\set Staff.clefGlyph = "clefs.C"
c'1
\set Staff.clefTransposition = 7
c'1
\set Staff.clefTransposition = 0
\set Staff.clefPosition = 0
c'1

% Return to the normal clef.
\set Staff.middleCPosition = 0
c'1
}

```



Vedi anche

Guida alla notazione: [Chiavi mensurali], pagina 459, [Chiavi gregoriane], pagina 466, [Intavolature predefinite], pagina 360, [Intavolature personalizzate], pagina 378, [Formattazione delle notine], pagina 226.

File installati: scm/parser-clef.scm.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Clef_engraver” in *Guida al Funzionamento Interno*, Sezione “Clef” in *Guida al Funzionamento Interno*, Sezione “ClefModifier” in *Guida al Funzionamento Interno*, Sezione “clef-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

I numeri di ottavazione assegnati alle chiavi sono trattati come oggetti grafici separati. Quindi qualsiasi `\override` all’oggetto *Clef* dovrà essere applicato, con un altro `\override`, all’oggetto *ClefModifier*.

```

\new Staff \with {
  \override Clef.color = #(universal-color 'blue)
}

```

```
\override ClefModifier.color = #(universal-color 'vermillion)
}
```

```
\clef "treble_8" c'4
```



Armatura di chiave

Nota: I nuovi utenti sono talvolta confusi dalla gestione delle alterazioni e delle armature di chiave. In LilyPond i nomi delle note costituiscono l'input grezzo; le armature e le chiavi determinano come questo venga mostrato. Una nota non alterata come `c` significa 'Do naturale', indipendentemente dall'armatura o dalla chiave. Per maggiori informazioni si veda Sezione "Altezze e armature di chiave" in *Manuale di Apprendimento*.

L'armatura di chiave indica la tonalità di un brano. È costituita da un insieme di alterazioni (bemolle o diesis) all'inizio del rigo. L'armatura di chiave può essere modificata:

```
\key altezza modo
```

`modo` deve essere `\major` o `\minor` per ottenere rispettivamente un'armatura di *altezza-maggiore* o *altezza-minore*. È anche possibile usare i nomi tradizionali dei modi, chiamati anche *modi ecclesiastici*: `\ionian`, `\dorian`, `\phrygian`, `\lydian`, `\mixolydian`, `\aeolian` e `\locrian`.

```
\relative {
  \key g \major
  fis''1
  f
  fis
}
```



Si possono definire ulteriori modi elencando le alterazioni per ogni grado della scala quando il modo inizia col Do.

```
\freygish = #`((0 . ,NATURAL) (1 . ,FLAT) (2 . ,NATURAL)
(3 . ,NATURAL) (4 . ,NATURAL) (5 . ,FLAT) (6 . ,FLAT))
```

```
\relative {
  \key c \freygish c'4 des e f
  \bar "||" \key d \freygish d es fis g
}
```



Le alterazioni dell'armatura di chiave possono essere collocate in posizioni diverse da quelle tradizionali o anche in più di un'ottava, usando le proprietà `flat-positions` e `sharp-positions` di `KeySignature`. I valori di queste proprietà specificano l'estensione delle posizioni del rigo in cui potranno comparire le alterazioni. Se viene specificata una sola posizione, le alterazioni vengono collocate entro l'ottava che finisce in quella posizione del rigo.

```
\override Staff.KeySignature.flat-positions = #'((-5 . 5))
\override Staff.KeyCancellation.flat-positions = #'((-5 . 5))
\clef bass \key es \major es g bes d'
\clef treble \bar "||" \key es \major es' g' bes' d''

\override Staff.KeySignature.sharp-positions = #'(2)
\bar "||" \key b \major b' fis' b'2
```



Frammenti di codice selezionati

Impedire l'inserimento dei segni di bequadro quando cambia l'armatura di chiave

Quando l'armatura di chiave cambia, vengono inseriti automaticamente i segni di bequadro per annullare le alterazioni di precedenti armature. Si può evitare questo comportamento impostando su `f` (falso) la proprietà `printKeyCancellation` nel contesto `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Armature di chiave non tradizionali

Il comando `\key` comunemente usato imposta la proprietà `keyAlterations` del contesto `Staff`. Per creare armature di chiave non standard, tale proprietà va impostata esplicitamente.

Il formato di questo comando è una lista:

```
\set Staff.keyAlterations = #`(((ottava . grado) . alterazione) ((ottava .
grado) . alterazione) ...)
```

dove, per ogni elemento della lista, `ottava` indica l'ottava (0 è l'ottava dal Do centrale al Si precedente), `grado` indica la nota all'interno dell'ottava (0 significa Do e 6 significa Si) e `alterazione` può essere `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc.

Altrimenti, usando il formato breve per ogni elemento della lista, `(grado . alterazione)`, ciò indica che la stessa alterazione deve essere presente in tutte le ottave. Per le scale microtonalidove

un “diesis” non è 100 centesimi, alterazione si riferisce alla proporzione di un duecentesimo di tono intero.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                (1 . ,SEMI-FLAT)
                                (2 . ,FLAT)
                                (5 . ,FLAT)
                                (6 . ,SEMI-FLAT))

  % \set Staff.extraNatural = ##f
  re reb \down reb resd
  dod dob dosd \down dob |
  dobsb dodsd do do |
}
```



Vedi anche

Glossario musicale: Sezione “church mode” in *Glossario Musicale*, Sezione “scordatura” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Altezze e armature di chiave” in *Manuale di Apprendimento*.

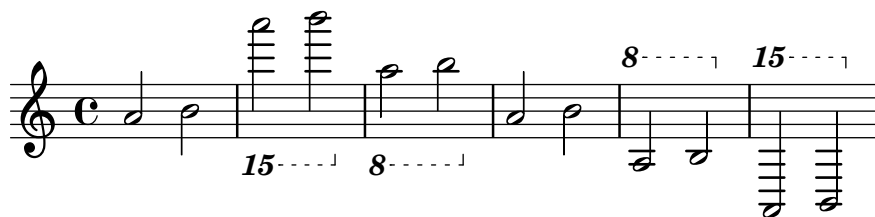
Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “KeyChangeEvent” in *Guida al Funzionamento Interno*, Sezione “Key_engraver” in *Guida al Funzionamento Interno*, Sezione “Key_performer” in *Guida al Funzionamento Interno*, Sezione “KeyCancellation” in *Guida al Funzionamento Interno*, Sezione “KeySignature” in *Guida al Funzionamento Interno*, Sezione “key-signature-interface” in *Guida al Funzionamento Interno*.

Segni di ottavazione

I *segni di ottavazione* introducono un’ulteriore trasposizione di ottava nel rigo:

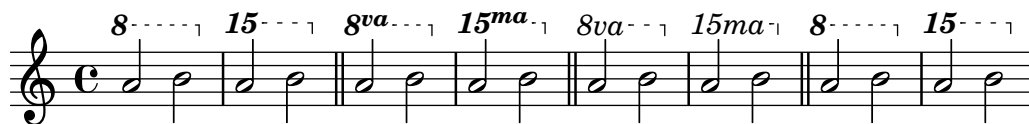
```
\relative c'' {
  a2 b
  \ottava #-2
  a2 b
  \ottava #-1
  a2 b
  \ottava #0
  a2 b
  \ottava #1
  a2 b
  \ottava #2
  a2 b
}
```



Per impostazione predefinita, viene stampato solo un numero all'inizio della parentesi. Questa impostazione può essere cambiata in modo che includa un numero ordinale abbreviato, come apice o come lettere normali (le ultime venivano un tempo usate come predefinite); si può cambiare anche il peso del font grassetto predefinito, come è spiegato in [Scelta del tipo di carattere e della dimensione], pagina 261.

L'esempio seguente mostra varie opzioni, e anche come ritornare al comportamento predefinito corrente:

```
\relative c' ' {
  \ottava #1
  a'2 b
  \ottava #2
  a'2 b
  \bar "||"
  \set Staff.ottavationMarkups = #ottavation-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \override Staff.OttavaBracket.font-series = #'normal
  \set Staff.ottavationMarkups = #ottavation-simple-ordinals
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
  \bar "||"
  \revert Staff.OttavaBracket.font-series
  \set Staff.ottavationMarkups = #ottavation-numbers
  \ottava #1
  a,2 b
  \ottava #2
  a'2 b
}
```



Frammenti di codice selezionati

Changing ottava text

Internally, `\ottava` sets the properties `ottavation` (for example, to `8va` or `8vb`) and `middleCPosition`. To override the text of the bracket, set `ottavation` after invoking `\ottava`.

Short text is especially useful when a brief ottava is used.

```
{
```

```

c'2
\ottava 1
\set Staff.ottavation = "8"
c''2
\ottava 0
c'1
\ottava 1
\set Staff.ottavation = "Text"
c''1
}

```



Aggiungere un segno di ottava a una sola voce

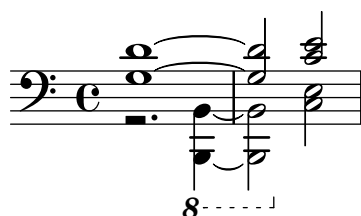
Se il rigo ha più di una voce, l'ottavazione in una voce trasporterà la posizione delle note in tutte le voci per la durata della parentesi dell'ottava. Se si intende applicare l'ottavazione a una sola voce, occorre spostare l'incisore `Ottava_spanner_engraver` nel contesto `Voice`.

```

\layout {
  \context {
    \Staff
    \remove Ottava_spanner_engraver
  }
  \context {
    \Voice
    \consists Ottava_spanner_engraver
  }
}

{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
  {
    r2.
    \ottava -1
    <b,,, b,,,>4 ~ |
    q2
    \ottava 0
    <c e>2
  }
  >>
}

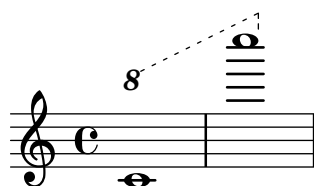
```



Modificare l'inclinazione dell'estensore dell'ottava

È possibile cambiare l'inclinazione dell'estensore dell'ottava.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0)
      (attach-dir . ,LEFT)
      (padding . 0)
      (stencil-align-dir-y . ,CENTER)))
    (right . ((Y . 5.0) ; Change the number here
      (padding . 0)
      (attach-dir . ,RIGHT)
      (text . ,(make-draw-dashed-line-markup
        (cons 0 -1.2)))))
  \override Staff.OttavaBracket.left-bound-info =
    #ly:horizontal-line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
    #ly:horizontal-line-spanner::calc-right-bound-info
  \ottava 1
  c1
  c''1
}
```



Vedi anche

Glossario musicale: Sezione “ottavazione” in *Glossario Musicale*.

Guida alla notazione: [Scelta del tipo di carattere e della dimensione], pagina 261.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Ottava_spanner_engraver” in *Guida al Funzionamento Interno*, Sezione “OttavaBracket” in *Guida al Funzionamento Interno*, Sezione “ottava-bracket-interface” in *Guida al Funzionamento Interno*.

Trasporto strumentale

Quando si scrivono partiture che comprendono strumenti traspositori, alcune parti possono essere scritte a un'altezza diversa dall'*intonazione reale*. In questi casi, è necessario specificare la chiave dello *strumento traspositore*, altrimenti l'output MIDI e le citazioni in altre parti produrranno altezze errate. Per maggiori informazioni sulle citazioni, si veda [Citare altre voci], pagina 222.

```
\transposition altezza
```

L'altezza da usare per `\transposition` deve corrispondere al suono effettivamente prodotto quando un `c'` scritto sul rigo viene suonato dallo strumento traspositore. Tale altezza viene inserita in modalità assoluta; dunque, uno strumento che produce un suono reale un tono sopra la notazione deve usare `\transposition d'`. `\transposition` va usato *soltanto* se le altezze *non* sono scritte in intonazione reale.

Ecco un frammento per violino e clarinetto in Si bemolle, le cui parti sono inserite usando le note e l'armatura di chiave che appaiono nei rispettivi righi sulla partitura del direttore. I due strumenti suonano all'unisono.

```
\new GrandStaff <<
  \new Staff = "violin" \with {
    instrumentName = "Vln"
    midiInstrument = "violin"
  }
  \relative c'' {
    % non strettamente necessario, ma meglio ricordarlo
    \transposition c'
    \key c \major
    g4( c8) r c r c4
  }
  \new Staff = "clarinet" \with {
    instrumentName = \markup { Cl (B\flat) }
    midiInstrument = "clarinet"
  }
  \relative c'' {
    \transposition bes
    \key d \major
    a4( d8) r d r d4
  }
}>>
```



`\transposition` può essere modificato nel corso di un brano. Ad esempio, un clarinettista potrebbe essere costretto a passare da un clarinetto in La a uno in Si bemolle.

```
flute = \relative c'' {
  \key f \major
  \cueDuring "clarinet" #DOWN {
    R1 _\markup\tiny "clarinet"
    c4 f e d
    R1 _\markup\tiny "clarinet"
  }
}

clarinet = \relative c'' {
  \key aes \major
  \transposition a
  aes4 bes c des
  R1~\markup { muta in B\flat }
  \key g \major
  \transposition bes
  d2 g,
}

\addQuote "clarinet" \clarinet
```

```
<<
  \new Staff \with { instrumentName = "Flute" }
    \flute
  \new Staff \with { instrumentName = "Cl (A)" }
    \clarinet
>>
```

[illegible]

Vedi anche

Glossario musicale: Sezione “intonazione reale” in *Glossario Musicale*, Sezione “strumento traspositore” in *Glossario Musicale*.

Guida alla notazione: [Citare altre voci], pagina 222, [Trasposizione], pagina 11.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Alterazioni automatiche

Esistono diverse convenzioni sul modo di scrivere le alterazioni. LilyPond ha una funzione per specificare lo stile di gestione delle alterazioni adottato. Questa funzione viene richiamata nel modo seguente:

```
\new Staff <<
  \accidentalStyle voice
  { ... }
>>
```

La gestione delle alterazioni si applica di norma all'attuale Staff (con l'eccezione degli stili choral piano e piano-cautionary, che sono spiegati dopo). Questa funzione accetta un secondo argomento opzionale che determina in quale ambito debba essere cambiato lo stile. Ad esempio, per usare lo stesso stile in tutti i righi dell'attuale StaffGroup, si usa:

\accidentalStyle StaffGroup.voice

Sono supportati i seguenti modi di gestire le alterazioni. Il seguente esempio mostra tutti gli stili:

```
musicA = {
  <<
    \relative {
      cis''8 fis, bes4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
  \\
  \relative {
    ais'2 cis, |
    fis8 b a4 cis2 |
  }
  >>
}

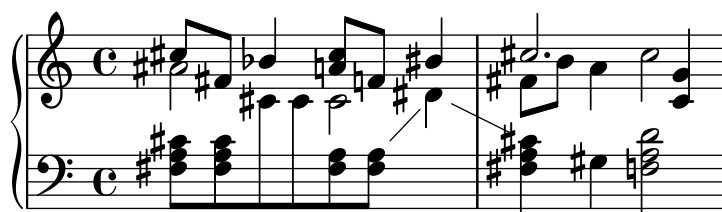
musicB = {
```

```

\clef bass
\new Voice {
  \voiceTwo \relative {
    <fis a cis>8[ <fis a cis>
    \change Staff = up
    cis' cis
    \change Staff = down
    <fis, a> <fis a>]
    \showStaffSwitch
    \change Staff = up
    dis'4 |
    \change Staff = down
    <fis, a cis>4 gis <f a d>2 |
  }
}
}

\new PianoStaff {
  <<
    \new Staff = "up" {
      \accidentalStyle default
      \musicA
    }
    \new Staff = "down" {
      \accidentalStyle default
      \musicB
    }
  >>
}

```



Si noti che le ultime linee di questo esempio possono essere sostituite dal seguente frammento, se si vuole usare lo stesso stile in entrambi i righe.

```

\new PianoStaff {
  <<
    \new Staff = "up" {
      %%% change the next line as desired:
      \accidentalStyle Score.default
      \musicA
    }
    \new Staff = "down" {
      \musicB
    }
  >>
}
default

```

Questo è il comportamento predefinito del compositore tipografico. Corrisponde alla pratica comunemente impiegata dal diciottesimo secolo: le alterazioni vengono ricordate fino alla fine della misura in cui si trovano, limitatamente all’ottava di appartenenza. Quindi, nell’esempio seguente non compare alcun segno di bequadro prima del b nella seconda misura o prima dell’ultimo c:



voice

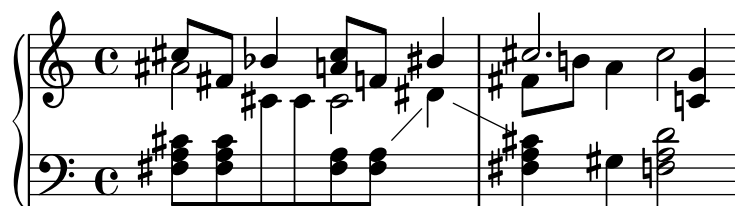
Normalmente le alterazioni mantengono la propria validità a livello di Staff. Tuttavia in questo stile le alterazioni vengono gestite individualmente per ogni voce. Al di fuori di quest’aspetto, lo stile è analogo a default.

Di conseguenza, le alterazioni relative a una voce non vengono cancellate nelle altre voci. Un risultato spesso non desiderabile: nell’esempio seguente è difficile capire se il secondo a sia naturale o diesis. L’opzione `voice` deve essere quindi usata solo se ogni voce è destinata a un esecutore diverso. Se la partitura deve essere letta da un unico musicista (come nel caso della partitura del direttore, o di uno spartito per pianoforte), allora è preferibile usare `modern` o `modern-cautionary`.



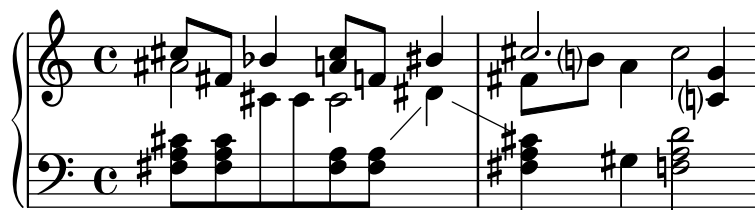
modern

Questa regola corrisponde alla pratica comune del ventesimo secolo. Omette i segni di bequadro supplementari che in passato erano di norma anteposti al diesis che segue un doppio diesis o a un bemolle che segue un doppio bemolle. La regola `modern` presenta le stesse alterazioni di `default`, con due aggiunte che servono a evitare ambiguità: i segni di annullamento delle alterazioni temporanee sono anteposti alle note sulla stessa ottava della misura successiva e alle note in ottave diverse nella stessa misura. In questo esempio, dunque, i bequadri del b e del c nella seconda misura del rigo superiore:



modern-cautionary

Questa regola è simile a `modern`, ma le alterazioni ‘supplementari’ sono segnate come alterazioni di precauzione (con parentesi). La loro dimensione può essere cambiata attraverso la proprietà `font-size` di `AccidentalCautionary`.



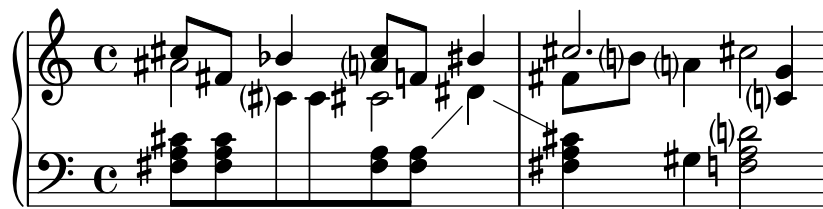
modern-voice

Questa regola viene usata per le alterazioni su più voci destinate sia agli esecutori che suonano una singola voce sia a quelli che suonano tutte le voci. Le alterazioni sono mostrate su tutte le voci, ma *sono annullate* su ogni voce dello stesso rigo (Staff). Quindi, l'alterazione dell' a nell'ultima misura viene annullata perché l'annullamento precedente si trovava in una voce diversa, mentre quella del d nel rigo inferiore viene annullata a causa dell'alterazione in un'altra voce della misura precedente:



modern-voice-cautionary

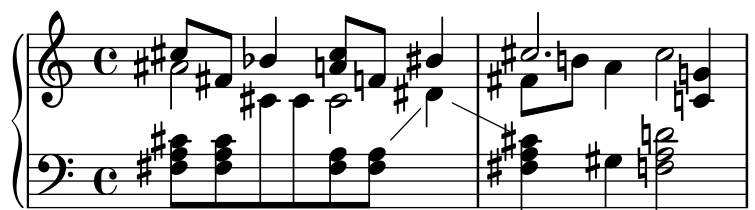
Questa regola è analoga a modern-voice, ma con le alterazioni supplementari (quelle non mostrate da voice) segnate come alterazioni di precauzione. Tutte le alterazioni mostrate da default *sono* mostrate con questa regola, ma alcune di esse sono indicate come alterazioni di precauzione.



piano

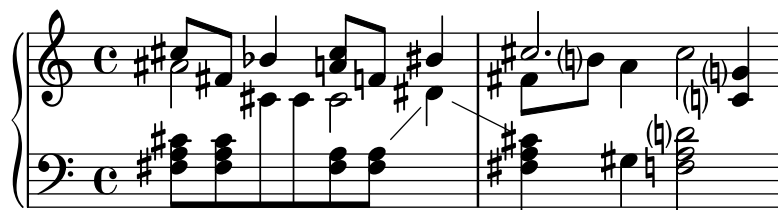
Questa regola riflette la pratica del ventesimo secolo per la notazione per pianoforte. Il suo comportamento è molto simile allo stile modern, ma in questo caso le alterazioni vengono annullate in tutti i righe che si trovano nello stesso GrandStaff o PianoStaff, dunque tutte gli annullamenti delle note finali.

È lo stile che si applica agli attuali GrandStaff e PianoStaff a meno che non sia qualificato con un secondo argomento.



piano-cautionary

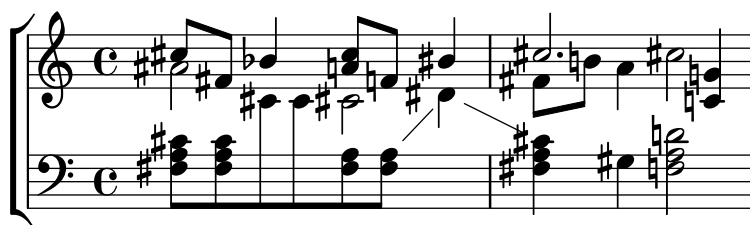
È uguale a piano ma con le alterazioni supplementari mostrate come alterazioni di precauzione.



choral

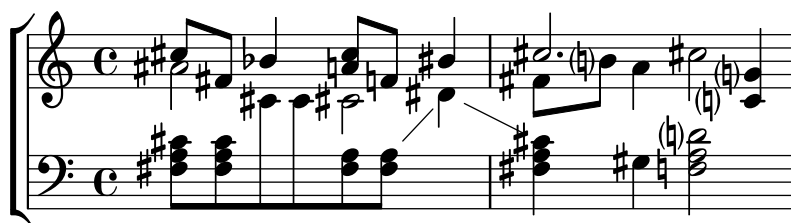
Questa regola è una combinazione degli stili modern-voice e piano. Mostra tutte le alterazioni richieste dai cantanti che seguono solo la propria voce, e mostra anche le ulteriori alterazioni per tutti i lettori che seguono tutte le voci di un intero ChoirStaff contemporaneamente.

Questo stile di alterazioni si applica al ChoirStaff corrente a meno che non sia qualificato con un secondo argomento.



choral-cautionary

Questo stile è identico a choral ma con le alterazioni extra in forma di alterazioni di cortesia.



neo-modern

Questa regola si riferisce a una pratica tipica della musica contemporanea: le alterazioni sono mostrate come in modern, ma vengono ripetute se la stessa nota appare in seguito nella stessa misura – a meno che la seconda occorrenza non segua direttamente la prima.



neo-modern-cautionary

Questa regola è simile a neo-modern, ma le alterazioni ‘supplementari’ sono mostrate come alterazioni di precauzione (con parentesi). La loro dimensione può essere modificata attraverso la proprietà font-size di AccidentalCautionary.



neo-modern-voice

Questa regola viene usata per le alterazioni su più di una voce che devono essere lette sia da musicisti che suonano una singola voce sia da musicisti che suonano tutte le voci. Le alterazioni per ogni voce sono mostrate come nello stile neo-modern, ma vengono annullate attraverso le voci nello stesso rigo (Staff).



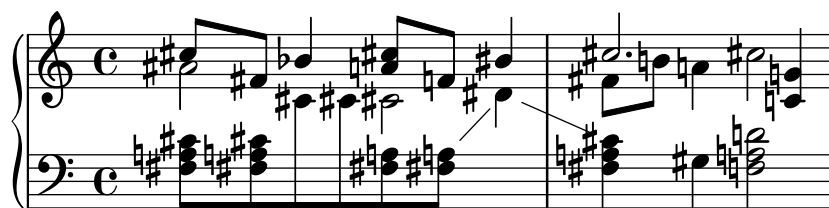
neo-modern-voice-cautionary

Questa regola è simile a neo-modern-voice, ma le alterazioni supplementari sono indicate come alterazioni di precauzione.



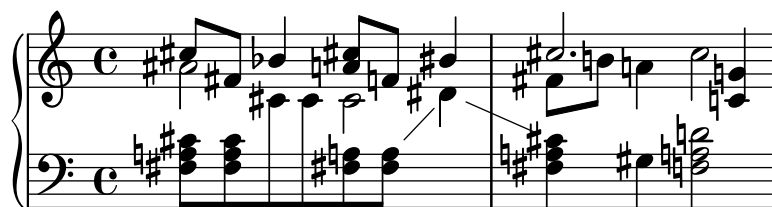
dodecaphonic

Questa regola riflette una regola introdotta dai compositori all'inizio del ventesimo secolo nel tentativo di abolire la gerarchia tra suoni naturali e non naturali. Con questo stile, *ogni* nota presenta un segno di alterazione, anche i suoni naturali.



dodecaphonic-no-repeat

Come nello stile delle alterazioni dodecafonico *ogni* nota ha un segno di alterazione, ma le alterazioni sono soppresse per tutte le altezze ripetute immediatamente nello stesso rigo.



dodecaphonic-first

In modo analogo allo stile delle alterazioni dodecafonico *ogni* altezza ha un segno di alterazione, ma solo la prima volta che si incontra in una misura. Le alterazioni vengono ricordate solo per l'ottava corrente ma in tutte le voci.



teaching

Questa regola è pensata per gli studenti: permette di generare facilmente degli spartiti di scale con le alterazioni di precauzione inserite in modo automatico. Alle alterazioni, indicate come nello stile modern, vengono aggiunte ulteriori segni di precauzione per tutti i diesis e bemolles specificati dall'armatura di chiave, fuorché nel caso di ripetizioni immediatamente successive di una stessa nota.



no-reset

È identico a default, ma le alterazioni mantengono la propria validità 'per sempre', non solo all'interno della singola misura:



forget

È il contrario di no-reset: le alterazioni non vengono ricordate affatto – pertanto, tutte le alterazioni si riferiscono all'armatura di chiave, indipendentemente dal materiale musicale precedente.



Vedi anche

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Accidental” in *Guida al Funzionamento Interno*, Sezione “Accidental_engraver” in *Guida al Funzionamento Interno*, Sezione “GrandStaff” in *Guida al Funzionamento Interno*, Sezione “PianoStaff” in *Guida al Funzionamento Interno*,

Sezione “Staff” in *Guida al Funzionamento Interno*, Sezione “AccidentalSuggestion” in *Guida al Funzionamento Interno*, Sezione “AccidentalPlacement” in *Guida al Funzionamento Interno*, Sezione “accidental-suggestion-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le note simultanee non vengono considerate nell’individuazione automatica delle alterazioni; vengono prese come riferimento solo le note precedenti e l’armatura di chiave. Se la stessa nota occorre simultaneamente con alterazioni diverse, può essere necessario forzare le alterazioni con ! o ?: ‘<f! fis!>’.

Nei finali alternativi, l’annullamento di precauzione dovrebbe basarsi sulla precedente misura *eseguita*, ma si basa sulla precedente misura *stampata*. Nell’esempio seguente il Do naturale della seconda volta non richiede il segno di bequadro:



Si può usare il seguente espediente: si definisce una funzione che imposti localmente lo stile delle alterazioni su forget:

```
forget = #(define-music-function (music) (ly:music?) #{
  \accidentalStyle forget
  #music
  \accidentalStyle modern
  #})
{
  \accidentalStyle modern
  \time 2/4
  \repeat volta 2 {
    c'2
  }
  \alternative {
    \volta 1 { cis' }
    \volta 2 { \forget c' }
  }
}
```



Glifi di alterazione alternativi

I sistemi di notazione non occidentali e antichi hanno le loro alterazioni. I glifi sono regolati dalla proprietà `alterationGlyphs` del contesto `Staff` e da tipi di contesto simili. I valori predefiniti di questa proprietà sono elencati in Sezione A.10 [Gruppi di glifi di alterazione], pagina 708.

```
\layout {
  \context {
    \Staff
    alterationGlyphs = #alteration-vaticana-glyph-name-alist
  }
}
```

```
{ ces' c' cis' }
```



La proprietà può essere impostata anche su una lista associativa personalizzata che associa le alterazioni ai nomi dei glifi. Le alterazioni si specificano come frazioni in toni. I glifi sono elencati in [Glifi delle alterazioni], pagina 695.

```
\layout {
  \context {
    \Staff
    alterationGlyphs =
      #'((-1/2 . "accidentals.flat.arrowdown")
        (0 . "accidentals.natural.arrowup")
        (1/2 . "accidentals.sharp.arrowup"))
  }
}
```

```
{ ces' c' cis' }
```

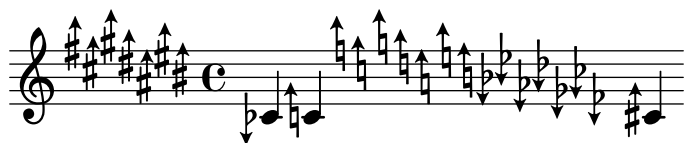


La proprietà `padding-pairs` degli oggetti `KeySignature` e `KeyCancellation` è una lista associativa che mappa coppie di glifi al padding che deve essere aggiunto tra questi glifi nelle armature di chiave.

```
\layout {
  \context {
    \Staff
    alterationGlyphs =
      #'((-1/2 . "accidentals.flat.arrowdown")
        (0 . "accidentals.natural.arrowup")
        (1/2 . "accidentals.sharp.arrowup"))
    \override KeySignature.padding-pairs =
      #'(((("accidentals.sharp.arrowup" . "accidentals.sharp.arrowup")
          . 0.25)
          (("accidentals.flat.arrowdown" . "accidentals.flat.arrowdown")
          . 0.3))
    \override KeyCancellation.padding-pairs =
      #'(((("accidentals.natural.arrowup" . "accidentals.natural.arrowup")
          . 0.7))
  }
}

{
  \key cis \major
  ces' c'
  \key ces \major
  cis'
}
```

}



Vedi anche

Guida alla notazione: Sezione A.10 [Gruppi di glifi di alterazione], pagina 708, [Glifi delle alterazioni], pagina 695.

Guida al funzionamento interno: Sezione “accidental-switch-interface” in *Guida al Funzionamento Interno*, Sezione “Alteration_glyph-engraver” in *Guida al Funzionamento Interno*, Sezione “key-signature-interface” in *Guida al Funzionamento Interno*.

Ambitus

Il termine *ambitus* (pl. *ambitus*) indica l’ambito di altezze di una determinata voce all’interno di una composizione musicale. Può indicare anche l’estensione di uno strumento musicale, ovvero l’intera gamma di suoni che può produrre. L’ambitus viene usato nelle parti vocali in modo che gli esecutori possano capire facilmente se siano adeguate alle loro possibilità.

L’ambitus viene indicato all’inizio del brano, prima della chiave iniziale. L’intervallo è individuato graficamente da due teste di nota che rappresentano l’altezza più bassa e più alta. Le alterazioni sono mostrate solo se non fanno parte dell’armatura di chiave.

```
\layout {
  \context {
    \Voice
    \consists Ambitus_engraver
  }
}

\relative {
  aes' c e2
  cis,1
}
```



Frammenti di codice selezionati

Un ambitus per voce

L’ambitus può essere specificato per voce. In tal caso occorre spostarlo manualmente per evitare collisioni.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = 2.0
    \voiceOne
    c4 a d e
    f1
  }
```

```

}
\new Voice \with {
  \consists "Ambitus_engraver"
} \relative c' {
  \voiceTwo
  es4 f g as
  b1
}
>>

```



Ambitus su più voci

Se si aggiunge l'incisore `Ambitus_engraver` al contesto `Staff` viene creato un solo ambitus per il rigo, anche nel caso di righe che hanno più voci.

```

\new Staff \with {
  \consists "Ambitus_engraver"
}
<<
\new Voice \relative c'' {
  \voiceOne
  c4 a d e
  f1
}
\new Voice \relative c' {
  \voiceTwo
  es4 f g as
  b1
}
>>

```



Modifica dell'intervallo dell'ambitus

È possibile cambiare le impostazioni predefinite dell'intervallo tra le teste di nota dell'ambitus e la linea che le collega.

```

\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

```

```

}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = 1.5
  c'4 g''
}

```



Ambitus dopo armatura di chiave

Per impostazione predefinita, gli ambitus sono posizionati a sinistra della chiave. La funzione `\ambitusAfter` permette di cambiare questo posizionamento. La sintassi è `\ambitusAfter grob-interface` (vedi Graphical Object Interfaces (<http://lilypond.org/doc/v2.22/Documentation/internals/graphical-object-interfaces>) per un elenco dei possibili valori per `grob-interface`.)

Un caso d'uso comune è il posizionamento dell'ambitus tra l'armatura di chiave e l'indicazione di tempo.

```

\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}

```



Vedi anche

Glossario musicale: Sezione “ambitus” in *Glossario Musicale*.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Ambitus-engraver” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*, Sezione “Staff” in *Guida al Funzionamento Interno*, Sezione “Ambitus” in *Guida al Funzionamento Interno*, Sezione “AmbitusAccidental” in *Guida al Funzionamento Interno*, Sezione “AmbitusLine” in *Guida al Funzionamento Interno*, Sezione “AmbitusNoteHead” in *Guida al Funzionamento Interno*, Sezione “ambitus-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le collisioni non vengono gestite in presenza di un ambitus multiplo su più di una voce.

1.1.4 Teste di nota

Questa sezione suggerisce i modi in cui modificare la testa di una nota.

Teste di nota speciali

L’aspetto delle teste delle note può essere modificato:

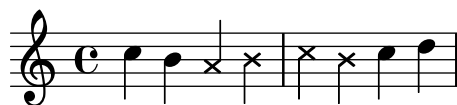
```
\relative c' ' {
  c4 b
  \override NoteHead.style = #'cross
  c4 b
  \revert NoteHead.style
  a b
  \override NoteHead.style = #'harmonic
  a b
  \revert NoteHead.style
  c4 d e f
}
```



L’elenco di tutti gli stili per le teste di nota è in Sezione A.9 [Stili delle teste di nota], pagina 707.

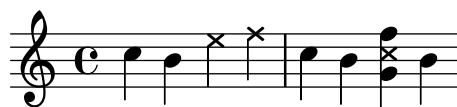
Lo stile barrato (cross) viene usato per rappresentare varie intenzioni musicali. I seguenti comandi generici predefiniti modificano la testa della nota nei contesti del rigo e dell’intavolatura e possono essere usati per rappresentare qualsiasi significato musicale:

```
\relative {
  c' '4 b
  \xNotesOn
  a b c4 b
  \xNotesOff
  c4 d
}
```



Questo comando può essere usato all'interno e all'esterno degli accordi per generare teste barrate sia nel contesto del rigo che in quello dell'intavolatura:

```
\relative {
  c' '4 b
  \xNote { e f }
  c b < g \xNote c f > b
}
```



Potete utilizzare, al posto di `\xNote`, `\xNotesOn` e `\xNotesOff`, i comandi `\deadNote`, `\deadNotesOn` e `\deadNotesOff`. Il termine *dead note* è di uso comune tra i chitarristi.

Esiste anche una scorciatoia simile per le forme a diamante:

```
\relative c' ' {
  <c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic> f\harmonic
}
```



Comandi predefiniti

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

Vedi anche

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida alla notazione: Sezione A.9 [Stili delle teste di nota], pagina 707, [Note in un accordo], pagina 178, [Indicazione di armonici e note smorzate], pagina 408.

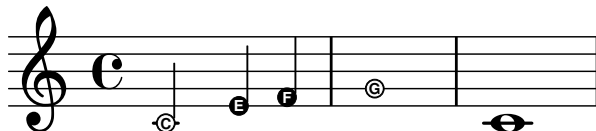
Guida al funzionamento interno: Sezione “note-event” in *Guida al Funzionamento Interno*, Sezione “Note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “Ledger_line_engraver” in *Guida al Funzionamento Interno*, Sezione “NoteHead” in *Guida al Funzionamento Interno*, Sezione “LedgerLineSpanner” in *Guida al Funzionamento Interno*, Sezione “note-head-interface” in *Guida al Funzionamento Interno*, Sezione “ledger-line-spanner-interface” in *Guida al Funzionamento Interno*.

Testa di nota con nome della nota

La nota ‘easy play’ inserisce il nome della nota dentro la testa. Viene usata nella musica per principianti. Per rendere le lettere leggibili, occorrerebbe usare un carattere più grande. A questo proposito si veda Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

```
 #(set-global-staff-size 26)
 \relative c' {
   \easyHeadsOn
   c2 e4 f
   g1
   \easyHeadsOff
 }
```

```
c,1
}
```



Comandi predefiniti

`\easyHeadsOn`, `\easyHeadsOff`.

Frammenti di codice selezionati

Numeri dentro le teste di nota

Le teste di nota con nome della nota usano la proprietà `note-names` dell'oggetto `NoteHead` per determinare cosa appaia all'interno della testa. È possibile sovrascrivere questa proprietà e mostrare numeri corrispondenti ai gradi della scala.

Si può creare un semplice incisore che faccia questo per ogni oggetto testa di nota che incontra.

```
#(define Ez_numbers_engraver
  (make-engraver
    (acknowledgers
      ((note-head-interface engraver grob source-engraver)
        (let* ((context (ly:translator-context engraver))
              (tonic-pitch (ly:context-property context 'tonic))
              (tonic-name (ly:pitch-notename tonic-pitch))
              (grob-pitch
                (ly:event-property (event-cause grob) 'pitch))
              (grob-name (ly:pitch-notename grob-pitch))
              (delta (modulo (- grob-name tonic-name) 7))
              (note-names
                (make-vector 7 (number->string (1+ delta))))))
          (ly:grob-set-property! grob 'note-names note-names))))))

#(set-global-staff-size 30)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

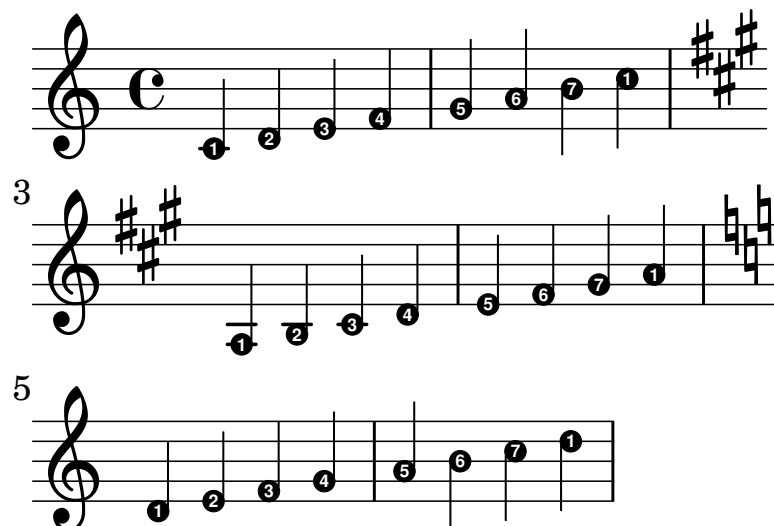
  \key a \major
  a,4 b cis d
  e4 fis gis a \break
```



```

\key d \dorian
d,4 e f g
a4 b c d
}

```



Vedi anche

Guida alla notazione: Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “note-event” in *Guida al Funzionamento Interno*, Sezione “Note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “NoteHead” in *Guida al Funzionamento Interno*, Sezione “note-head-interface” in *Guida al Funzionamento Interno*.

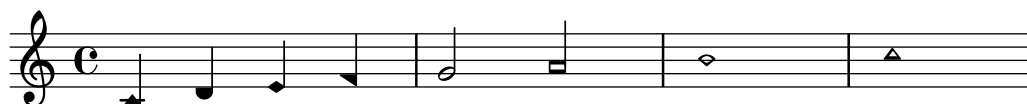
Teste di nota a forma variabile

In alcune notazioni, la forma della testa della nota corrisponde alla funzione armonica di una nota nella scala. Questa notazione era comune nei canzonieri americani del diciannovesimo secolo. Gli stili possibili sono Sacred Harp, Southern Harmony, Funk (Harmonia Sacra), Walker e Aiken (Christian Harmony):

```

\relative c'' {
  \aikenHeads
  c, d e f g2 a b1 c \break
  \aikenThinHeads
  c,4 d e f g2 a b1 c \break
  \sacredHarpHeads
  c,4 d e f g2 a b1 c \break
  \southernHarmonyHeads
  c,4 d e f g2 a b1 c \break
  \funkHeads
  c,4 d e f g2 a b1 c \break
  \walkerHeads
  c,4 d e f g2 a b1 c \break
}

```

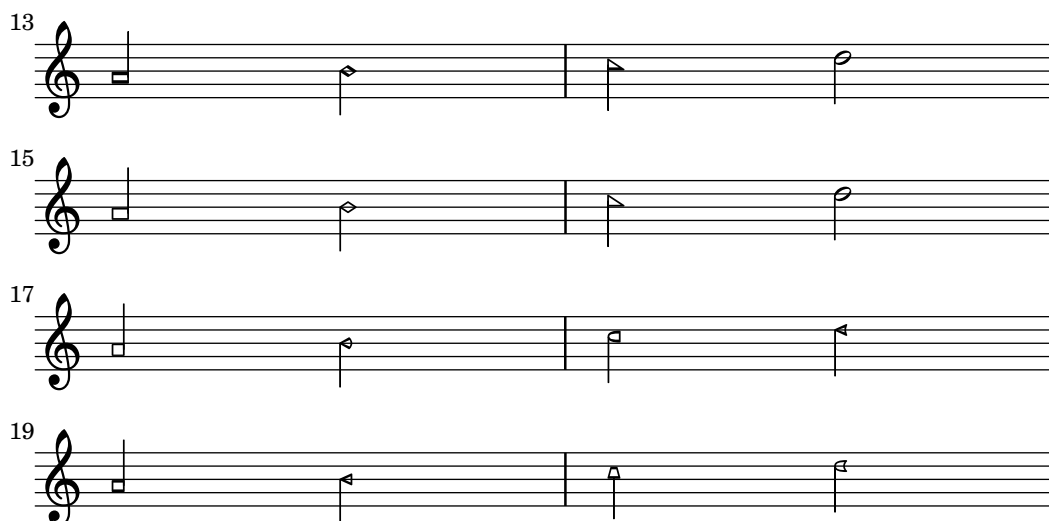




Le forme variano in base al grado della scala; la scala è determinata dal comando `\key`. Se si scrive in tonalità minore, il grado della scala può essere determinato in base alla relativa maggiore:

```
\relative c'' {
  \key a \minor
  \aikenHeads
  a b c d e2 f g1 a \break
  \aikenHeadsMinor
  a,4 b c d e2 f g1 a \break
  \aikenThinHeadsMinor
  a,4 b c d e2 f g1 a \break
  \sacredHarpHeadsMinor
  a,2 b c d \break
  \southernHarmonyHeadsMinor
  a2 b c d \break
  \funkHeadsMinor
  a2 b c d \break
  \walkerHeadsMinor
  a2 b c d \break
}
```





Comandi predefiniti

`\aikenHeads`, `\aikenHeadsMinor`, `\aikenThinHeads`, `\aikenThinHeadsMinor`, `\funkHeads`, `\funkHeadsMinor`, `\sacredHarpHeads`, `\sacredHarpHeadsMinor`, `\southernHarmonyHeads`, `\southernHarmonyHeadsMinor`, `\walkerHeads`, `\walkerHeadsMinor`.

Frammenti di codice selezionati

Teste di nota Aiken variante sottile

Le teste di nota Aiken bianche sono più difficili da leggere a dimensioni del rigo più piccole, specialmente coi tagli addizionali. Perdendo lo spazio bianco interno le fa apparire come note di un quarto.

```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
                          faThin sol laThin tiThin)
    c'' a' c' a
  }
}
```



Applicazione degli stili delle teste di nota in base al grado della scala

La proprietà `shapeNoteStyles` può essere usata per definire vari stili di teste di nota per ogni grado della scala (definita dall'armatura di chiave o dalla proprietà `tonic`). Questa proprietà richiede un insieme di simboli, che può essere puramente arbitrario (sono permesse espressioni geometriche come `triangle`, `cross` e `xcircle`) o basato sull'antica tradizione tipografica americana (sono consentiti anche alcuni nomi di nota latini).

Detto questo, per imitare gli antichi canzonieri americani, ci sono vari stili predefiniti disponibili attraverso dei comodi comandi come `\aikenHeads` o `\sacredHarpHeads`.

Questo esempio mostra modi diversi di ottenere teste di nota di varie forme e illustra la possibilità di trasporre una melodia senza perdere la corrispondenza tra le funzioni armoniche e gli stili delle teste.

```

fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                          #f la ti)

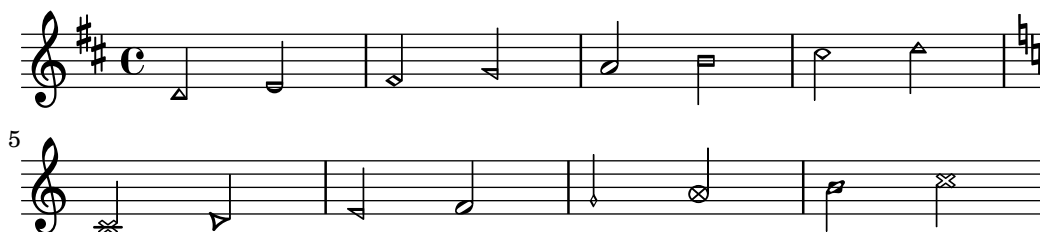
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                          mensural xcircle diamond)

    \fragment
  }
}

```



La lista completa di tutti gli stili delle teste si trova in Sezione A.9 [Stili delle teste di nota], pagina 707.

Vedi anche

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida alla notazione: Sezione A.9 [Stili delle teste di nota], pagina 707.

Guida al funzionamento interno: Sezione “note-event” in *Guida al Funzionamento Interno*, Sezione “Note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “NoteHead” in *Guida al Funzionamento Interno*, Sezione “note-head-interface” in *Guida al Funzionamento Interno*.

Improvvisazione

L'improvvisazione viene talvolta indicata con teste tagliate: l'esecutore può scegliere qualsiasi nota ma deve seguire il ritmo indicato. Si possono creare queste teste:

```

\new Voice \with {
  \consists Pitch_squash_engraver

```

```

} \relative {
  e''8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  2 ~ 8 f4 f8 ~
  2
  \improvisationOff
  a16( bes) a8 g e
}

```



Comandi predefiniti

`\improvisationOn`, `\improvisationOff`.

Vedi anche

Frammenti di codice: Sezione “Altezze” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Pitch_squash_engraver” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*, Sezione “RhythmicStaff” in *Guida al Funzionamento Interno*.

1.2 Ritmi

31 *a tempo*

cantabile

32 *cresc.*

33 *p*



Questa sezione tratta i ritmi, le pause, le durate, la disposizione delle travature e le battute.

1.2.1 Inserimento delle durate

Durata

Le durate delle note si inseriscono usando numeri e punti. Il numero è basato sul valore reciproco della lunghezza della nota. Per esempio, una nota di un quarto si indica col valore numerico di 4 perché il suo valore è $1/4$, mentre una minima si indica col 2, la croma con l'8 e così via. La minor durata esprimibile per una nota indipendente è di 128; è possibile inserire anche valori inferiori, ma solo all'interno di travature. Vedi anche Sezione 1.2.4 [Travature], pagina 88.

Per le note più lunghe di un intero si usano i comandi `\longa` (due volte una breve) e `\breve`. Solo nella notazione per musica antica è possibile specificare una nota che dura quattro volte una breve, attraverso il comando `\maxima`. Vedi Sezione 2.9 [Notazione antica], pagina 454.

```
\relative {
  \time 8/1
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



Ecco gli stessi valori con la disposizione automatica delle travature disabilitata.

```
\relative {
  \time 8/1
  \autoBeamOff
  c''\longa c\breve c1 c2
  c4 c8 c16 c32 c64 c128 c128
}
```



Le durate isolate, ovvero le durate prive di altezza, in una sequenza musicale prenderanno la loro altezza dalla nota o accordo precedenti.

```
\relative { a' a a2 a a4 a a1 a }
```



Le durate isolate prendono l'altezza dalla nota o accordo precedenti.

```
\relative {
  \time 8/1
  c'' \longa \breve 1 2
  4 8 16 32 64 128 128
}
```



Le altezze isolate, ovvero le altezze prive di durata, in una sequenza musicale prenderanno la loro durata dalla nota o accordo precedenti. Se non c'è alcuna durata precedente, il valore predefinito è sempre 4, una nota di un quarto.

```
\relative { a' a a2 a4 a a1 a }
```



Inserire un punto (.) dopo la durata per ottenere note “puntate”. Le figure con doppio punto si indicano aggiungendo due punti, e così via.

```
\relative { a'4 b c4. b8 a4. b4.. c8. }
```



Per evitare collisioni con le linee del rigo, di norma i punti delle note sono spostati in su. Tuttavia all'interno di passaggi polifonici possono essere posizionati manualmente, sopra o sotto il rigo a seconda delle necessità. Vedi Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Alcune durate non possono essere rappresentate usando solo durate e punti ma soltanto tramite legature di valore tra due o più note. Vedi [Legature di valore], pagina 58.

Per sapere come specificare durate che allineino le sillabe del testo alle note, si veda Sezione 2.1 [Musica vocale], pagina 280.

Le note possono essere distanziate in modo proporzionale alla loro durata. Vedi Sezione 4.5.5 [Notazione proporzionale], pagina 595.

Comandi predefiniti

```
\autoBeamOn, \autoBeamOff, \dotsUp, \dotsDown, \dotsNeutral.
```

Frammenti di codice selezionati

Note brevi alternative

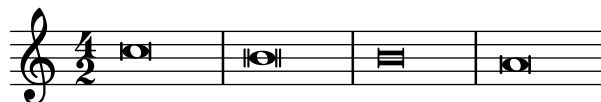
Le note brevi sono disponibili anche con due linee verticali su ciascun lato della testa invece di una sola e in stile barocco.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
```

```

\override Staff.NoteHead.style = #'baroque
b\breve
\revert Staff.NoteHead.style
a\breve
}

```



Modifica del numero di punti di aumentazione per nota

Il numero di punti di aumentazione su una singola nota può essere modificato in modo indipendente dai punti posizionati dopo la nota.

```

\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = 4
  c4.. a16 r2 |
  \override Dots.dot-count = 0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}

```



Vedi anche

Glossario musicale: Sezione “breve” in *Glossario Musicale*, Sezione “longa” in *Glossario Musicale*, Sezione “maxima” in *Glossario Musicale*, Sezione “durata” in *Glossario Musicale*, Sezione “Nomi di durata delle note e delle pause” in *Glossario Musicale*.

Guida alla notazione: Sezione 1.2.4 [Travature], pagina 88, [Legature di valore], pagina 58, [Gambi], pagina 244, Sezione 1.2.1 [Inserimento delle durate], pagina 49, Sezione 1.2.2 [Inserimento delle pause], pagina 63, Sezione 2.1 [Musica vocale], pagina 280, Sezione 2.9 [Notazione antica], pagina 454, Sezione 4.5.5 [Notazione proporzionale], pagina 595.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Dots” in *Guida al Funzionamento Interno*, Sezione “DotColumn” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Non c'è un limite massimo o minimo alla durata di una pausa, ma è il numero dei glifi ad essere limitato, dunque si possono indicare pause comprese tra 128 e \maxima.

Gruppi irregolari

I gruppi irregolari sono costituiti da un'espressione musicale introdotta dal comando \tuplet, che moltiplica la velocità dell'espressione musicale per una frazione:

```

\tuplet frazione { musica }

```

Il numeratore della frazione apparirà sopra o sotto le note; eventualmente, con l'aggiunta opzionale di una parentesi quadra. Il gruppo irregolare più comune è la terzina, in cui 3 note hanno la durata di 2:

```

\relative {

```



```
a'2 \tuplet 3/2 { b4 b b }
c4 c \tuplet 3/2 { b4 a g }
}
```



In caso di lunghi passaggi di gruppi irregolari, dover scrivere un comando `\tuplet` per ogni gruppo è scomodo. È possibile specificare direttamente la durata di un gruppo irregolare prima della musica per far sì che i gruppi siano suddivisi automaticamente:

```
\relative {
  g'2 r8 \tuplet 3/2 8 { cis16 d e e f g g f e }
}
```



Le parentesi dei gruppi irregolari si possono posizionare manualmente sopra o sotto il rigo:

```
\relative {
  \tupletUp \tuplet 3/2 { c''8 d e }
  \tupletNeutral \tuplet 3/2 { c8 d e }
  \tupletDown \tuplet 3/2 { f,8 g a }
  \tupletNeutral \tuplet 3/2 { f8 g a }
}
```



È possibile annidare i gruppi irregolari:

```
\relative {
  \autoBeamOff
  c''4 \tuplet 5/4 { f8 e f \tuplet 3/2 { e[ f g] } } f4
}
```



La modifica di gruppi irregolari annidati che iniziano simultaneamente richiede l'uso di `\tweak`.

Per modificare la durata delle note senza che appaia la parentesi quadra del gruppo irregolare, si veda [Scalare le durate], pagina 57.

Comandi predefiniti

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

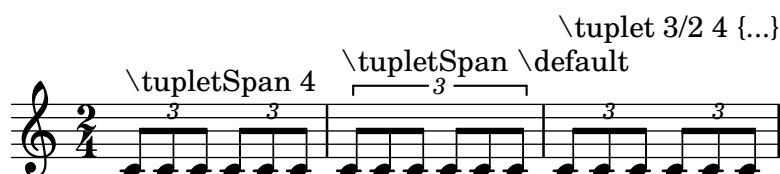
Frammenti di codice selezionati

Inserire vari gruppi irregolari usando una sola volta il comando `\tuplet`

La proprietà `tupletSpannerDuration` imposta la durata di ognuno dei gruppi irregolari compresi tra parentesi dopo il comando `\tuplet`. In questo modo si possono inserire molti gruppi irregolari consecutivi all'interno di una singola espressione `\tuplet`, risparmiando così tempo e spazio.

Ci sono vari modi per impostare `tupletSpannerDuration`. Il comando `\tupletSpan` la imposta su una certa durata e poi la annulla quando invece di una durata viene specificato `\default`. Altrimenti si può usare un argomento opzionale con `\tuplet`.

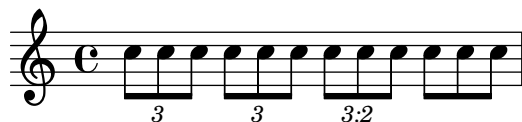
```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\tupletSpan \default" c c c c c }
  \tuplet 3/2 4 { c8^"\tuplet 3/2 4 {...}" c c c c c }
}
```



Modifica del numero del gruppo irregolare

Di norma compare sulla parentesi del gruppo irregolare solo il numeratore del numero del gruppo irregolare, ovvero il numeratore dell'argomento del comando `\tuplet`. Ma è possibile mostrare la frazione *num:den* del numero del gruppo irregolare oppure nascondere del tutto il numero.

```
\relative c' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```



Numeri non predefiniti per i gruppi irregolari

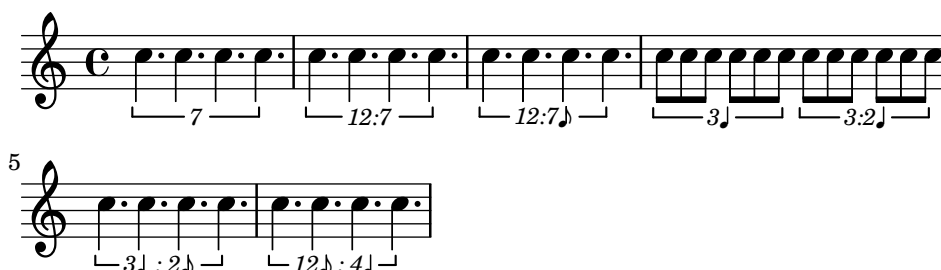
LilyPond fornisce anche funzioni di formattazione che permettono di creare numeri di gruppi irregolari diversi dalla frazione vera e propria, così come di aggiungere un valore di nota al numero o alla frazione di un gruppo irregolare.

```
\relative c' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2 { c4. c4. c4. c4. }
```

```

\once \override TupletNumber.text =
  #(tuplet-number::append-note-wrapper
    (tuplet-number::non-default-tuplet-fraction-text 12 7)
    (ly:make-duration 3 0))
\tuplet 3/2 { c4. c4. c4. c4. }
\once \override TupletNumber.text =
  #(tuplet-number::append-note-wrapper
    tuplet-number::calc-denominator-text
    (ly:make-duration 2 0))
\tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
\once \override TupletNumber.text =
  #(tuplet-number::append-note-wrapper
    tuplet-number::calc-fraction-text
    (ly:make-duration 2 0))
\tuplet 3/2 { c8 c8 c8 c8 c8 c8 }
\once \override TupletNumber.text =
  #(tuplet-number::fraction-with-notes
    (ly:make-duration 2 1) (ly:make-duration 3 0))
\tuplet 3/2 { c4. c4. c4. c4. }
\once \override TupletNumber.text =
  #(tuplet-number::non-default-fraction-with-notes 12
    (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
\tuplet 3/2 { c4. c4. c4. c4. }
}

```



Controllare la visibilità della parentesi del gruppo irregolare

Il comportamento predefinito relativo alla visibilità della parentesi quadra del gruppo irregolare è di mostrare una parentesi a meno che non ci sia una travatura della stessa lunghezza del gruppo.

Per controllare la visibilità di tale parentesi, si imposta la proprietà 'bracket-visibility' su #t (mostra sempre la parentesi), #if-no-beam (mostra la parentesi solo se non c'è una travatura, che è il comportamento predefinito) o #f (non mostrare mai la parentesi). L'ultima opzione equivale a omettere l'oggetto @code{TupletBracket} dall'output.

```

music = \relative c' {
  \tuplet 3/2 { c16[ d e ] f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

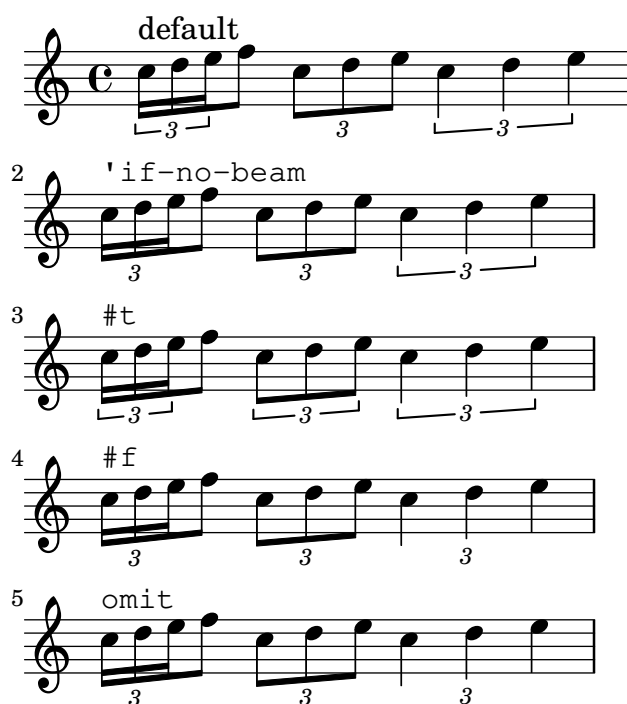
\new Voice {
  \relative c' {
    \override Score.TextMark.non-musical = ##f
    \textMark "default" \music
  }
}

```

```

\override TupletBracket.bracket-visibility = #'if-no-beam
\textMark \markup \typewriter "'if-no-beam" \music
\override TupletBracket.bracket-visibility = ##t
\textMark \markup \typewriter "#t" \music
\override TupletBracket.bracket-visibility = ##f
\textMark \markup \typewriter "#f" \music
\omit TupletBracket
\textMark \markup \typewriter "omit" \music
}
}

```



Consentire l'interruzione del rigo all'interno di gruppi irregolari con travature

Questo esempio artificioso mostra come permettere interruzioni del rigo sia manuali che automatiche all'interno di un gruppo irregolare con travature. Si noti che le travature di questi gruppi irregolari fuori dal ritmo devono essere disposte manualmente.

```

\layout {
  \context {
    \Voice
    % Permit automatic line breaks within tuplets.
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks.
    \override Beam.breakable = ##t
  }
}

\relative c'' {
  <>^"manually forced line break"
  a8
  \repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
  \tuplet 3/2 { c8[ b \break g16 a] }
}

```

```

\repeat unfold 5 { \tuplet 3/2 { c8[ b g16 a] } }
c8 \bar "||"
}

\relative c'' {
  <>^"automatic line break"
  \repeat unfold 28 a16
  \tuplet 11/8 { a16[ b c d e f e d c b a] }
  \repeat unfold 28 a16 \bar "||"
}

```

manually forced line break

automatic line break

Vedi anche

Glossario musicale: Sezione “terzina” in *Glossario Musicale*, Sezione “gruppo irregolare” in *Glossario Musicale*, Sezione “polimetrico” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Metodi di modifica” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Gestione del tempo], pagina 128, [Scalare le durate], pagina 57, Sezione 5.3.4 [Il comando `\tweak`], pagina 632, [Notazione polimetrica], pagina 81.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TupletBracket” in *Guida al Funzionamento Interno*, Sezione “TupletNumber” in *Guida al Funzionamento Interno*, Sezione “TimeScaled-Music” in *Guida al Funzionamento Interno*.

Scalare le durate

La durata di singole note, pause o accordi può essere moltiplicata per una frazione N/M aggiungendo $*N/M$ (o $*N$ se M è 1). Questo non cambierà l'aspetto delle note o delle pause, ma la durata così alterata verrà utilizzata per calcolare la posizione all'interno della misura e per impostare la durata nel file MIDI. Si possono combinare molteplici fattori, come $*L*M/N$. I fattori fanno parte della durata: quindi se non si specifica una durata per le note successive, la durata ripresa dalla nota precedente includerà il fattore di scalatura.

Nell'esempio seguente le prime tre note occupano esattamente due tempi, ma non sono indicate come gruppo irregolare.

```
\relative {
  \time 2/4
  % Trasforma le durate in terzine
  a'4*2/3 gis a
  % Durate normali
  a4 a
  % Raddoppia la durata dell'accordo
  <a d>4*2
  % Durata di un quarto, ma appare come un sedicesimo
  b16*4 c4
}
```



Anche la durata delle pause spaziatrici può essere modificata con un moltiplicatore. Può essere utile per saltare molte misure; per esempio $s1*23$.

Frammenti musicali più lunghi possono essere compressi secondo la stessa proporzione, come moltiplicando ogni nota, accordo o pausa per una medesima frazione. In questo modo, l'aspetto della musica non cambia ma la durata interna delle note viene moltiplicata per la frazione num/den . Ecco un esempio che mostra come la musica possa essere compressa e espansa:

```
\relative {
  \time 2/4
  % Durate normali
  <c' a>4 c8 a
  % Scala la musica di *2/3
  \scaleDurations 2/3 {
    <c a f>4. c8 a f
  }
  % Scala la musica di *2
  \scaleDurations 2/1 {
    <c' a>4 c8 b
  }
}
```



Questo comando torna utile nella notazione polimetrica, si veda [Notazione polimetrica], pagina 81.

Vedi anche

Guida alla notazione: [Gruppi irregolari], pagina 51, [Pause invisibili], pagina 65, [Notazione polimetrica], pagina 81.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Problemi noti e avvertimenti

Il calcolo della posizione in una misura deve considerare tutti i fattori di dimensionamento applicati alle note di quella misura e gli esigui residui delle misure precedenti. Questo calcolo viene fatto con numeri razionali. Se un numeratore o un denominatore intermedi in quel calcolo eccedono di 2^{30} , l'esecuzione e la composizione tipografica si arresteranno in quel punto senza indicare un errore.

Legature di valore

Una legatura di valore connette le teste di due note della stessa altezza successive. Dunque, la legatura di valore prolunga la durata di una nota.

Nota: Le legature di valore non devono essere confuse con le *legature di portamento*, che articolano un passaggio, o con le *legature di frase*, che delimitano una frase musicale. Una legatura di valore serve semplicemente a prolungare la durata di una nota, in modo analogo al punto di valore.

La legatura di valore si inserisce aggiungendo il simbolo tilde (~) alla prima di ogni coppia di note legate. Esso indica che la nota deve essere legata alla nota successiva, che deve essere della stessa altezza.

```
{ a'2~ 4~ 16 r r8 }
```



Le legature di valore possono avvantaggiarsi dell'interpretazione dell' 'ultima altezza esplicita' per le durate isolate:

```
{ a'2~ 4~ 16 r r8 }
```



Le legature di valore si usano per unire due note a cavallo di una stanghetta di battuta, oppure quando non si possono usare i punti per esprimere una particolare durata. Le legature si dovrebbero usare anche per unire note dalle durate superiori all'unità di suddivisione della misura:

```
\relative {  
  r8 c'4.~ 4 r4 |
```

```
r8~"non" c2~ 8 r4
}
```



Per legare una successione di note la cui durata si prolunga per più misure intere, è più semplice ricorrere alla suddivisione automatica delle note, come è spiegato in [Divisione automatica delle note], pagina 84. Questo metodo divide automaticamente le note lunghe e le connette da misura a misura.

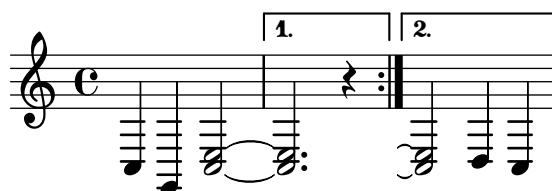
Quando si applica una legatura di valore a degli accordi, vengono legate tutte le teste delle note della stessa altezza. In assenza di altezze corrispondenti, non verrà creata alcuna legatura. Singoli suoni degli accordi possono essere legati inserendo la legatura all'interno dell'accordo stesso.

```
\relative c' {
  <c e g>2~ 2 |
  <c e g>4~ <c e g c>
  <c~ e g~ b> <c e g b> |
}
```



Quando la battuta della "seconda volta" di un ritornello inizia con una nota legata a quella precedente, occorre indicare la legatura nel modo seguente:

```
\relative {
  \repeat volta 2 { c g <c e>2~ }
  \alternative {
    % Prima volta: la nota seguente viene legata in modo normale
    { <c e>2. r4 }
    % Seconda volta: la nota seguente ha una legatura ripetuta
    { <c e>2\repeatTie d4 c }
  }
}
```



Le legature *L.v.* (*laissez vibrer*) indicano che le note non devono essere terminate nettamente. Si usa nella notazione per pianoforte, arpa e altri strumenti a corda e a percussione. Si inseriscono così:

```
<c' f' g'>1\laissezVibrer
```



Le legature di valore possono essere impostate manualmente per avere la curva in su o in giù, come è spiegato in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Le legature di valore possono essere tratteggiate, punteggiate, oppure tracciate secondo una successione di tratti continui e tratti interrotti.

```
\relative c' {
  \tieDotted
  c2~ 2
  \tieDashed
  c2~ 2
  \tieHalfDashed
  c2~ 2
  \tieHalfSolid
  c2~ 2
  \tieSolid
  c2~ 2
}
```



Si possono specificare modelli di tratteggiatura personalizzati:

```
\relative c' {
  \tieDashPattern #0.3 #0.75
  c2~ 2
  \tieDashPattern #0.7 #1.5
  c2~ 2
  \tieSolid
  c2~ 2
}
```



Le definizioni dei modelli di tratteggiatura delle legature di valore hanno la stessa struttura di quelle per le legature di portamento. I dettagli relativi ai modelli complessi di tratteggiatura sono trattati in [Legature di portamento], pagina 142.

Sovrascrivere le proprietà di formattazione *whiteout* e *layer* degli oggetti che devono creare uno spazio vuoto tra le legature di valore.

```
\relative {
  \override Tie.layer = #-2
  \override Staff.TimeSignature.layer = #-1
  \override Staff.KeySignature.layer = #-1
  \override Staff.TimeSignature.whiteout = ##t
  \override Staff.KeySignature.whiteout = ##t
  b'2 b~
  \time 3/4
  \key a \major
  b r4
}
```



Comandi predefiniti

`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`,
`\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Frammenti di codice selezionati

Usare le legature di valore con un arpeggio

Le legature di valore vengono usate talvolta per scrivere un arpeggio. In questo caso, le due note da legare devono non essere consecutive. Per ottenere tale risultato occorre impostare la proprietà `tieWaitForNote` su `#t`. Questa funzionalità serve anche a legare un tremolo a un accordo e in generale qualsiasi coppia di note consecutive.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Disegnare manualmente le legature di valore

Le legature di valore possono essere disegnate a mano cambiando la proprietà `tie-configuration` dell'oggetto `TieColumn`. Il primo numero indica la distanza dal centro del rigo nell'unità di metà spazio rigo, mentre il secondo numero indica la direzione (1 = su, -1 = giù).

Si noti che LilyPond fa una distinzione tra valori precisi e imprecisi per il primo numero. Se si usa un valore preciso (ovvero un intero o una frazione come $(/ 4 5)$), il valore serve come posizione verticale approssimata che viene ulteriormente aggiustata da LilyPond per far sì che la legatura di valore eviti le linee del rigo. Se si usa un valore impreciso, come una virgola mobile, viene usato per la posizione verticale senza ulteriori regolazioni.

```
\relative c' {
  <>^"default"
  g'1 ^~ g

  <>^"0"
  \once \override Tie.staff-position = 0
  g1 ^~ g

  <>^"0.0"
```

```

\once \override Tie.staff-position = 0.0
g1 ^~ g

<>^"reset"
\revert Tie.staff-position
g1 ^~ g
}

\relative c' {
  \override TextScript.outside-staff-priority = ##f
  \override TextScript.padding = 0

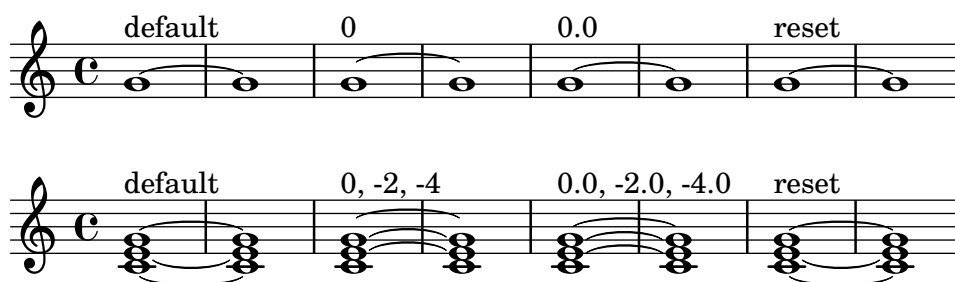
  <>^"default"
  <c e g>1~ <c e g>

  <>^"0, -2, -4"
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>1~ <c e g>

  <>^"0.0, -2.0, -4.0"
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>1~ <c e g>

  <>^"reset"
  \override TieColumn.tie-configuration = ##f
  <c e g>1~ <c e g>
}

```



Vedi anche

Glossario musicale: Sezione “legatura di valore” in *Glossario Musicale*, Sezione “laissez vibrer” in *Glossario Musicale*.

Guida alla notazione: [Legature di portamento], pagina 142, [Divisione automatica delle note], pagina 84.

Frammenti di codice: Sezione “Expressive marks” in *Frammenti di codice*, Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “LaissezVibrerTie” in *Guida al Funzionamento Interno*, Sezione “LaissezVibrerTieColumn” in *Guida al Funzionamento Interno*, Sezione “TieColumn” in *Guida al Funzionamento Interno*, Sezione “Tie” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Cambiare rigo mentre una legatura di valore è attiva non produce una legatura obliqua.

Il cambio di chiave o di ottava durante una legatura di valore non è una situazione ben definita. In questi casi è preferibile usare una legatura di portamento.

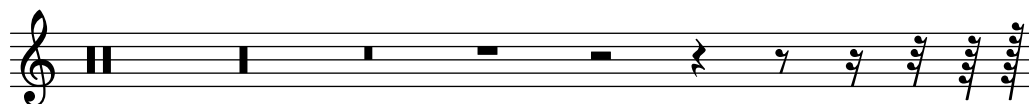
1.2.2 Inserimento delle pause

Le pause si inseriscono insieme alla musica contenuta nelle espressioni musicali.

Pause

Le pause si inseriscono allo stesso modo delle note, ma con il carattere `r`. Le durate più lunghe di un intero usano i seguenti comandi predefiniti:

```
\new Staff {
  % Queste due linee servono solo ad abbellire questo esempio
  \time 16/1
  \omit Staff.TimeSignature
  % Mostra una pausa di maxima, equivalente a quattro brevi
  r\maxima
  % Mostra una pausa di longa, equivalente a due brevi
  r\longa
  % Mostra una pausa di breve
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Le pause d'intero, poste al centro della misura, devono essere inserite come pause multiple. Si possono usare sia per una sola misura sia su più misure, come è spiegato in [Pause d'intero], pagina 66.

Per indicare esplicitamente la posizione verticale di una pausa, si scrive la nota corrispondente seguita da `\rest`. Una pausa della durata della nota verrà collocata nella posizione della nota sul rigo. Questo permette una precisa formattazione manuale della musica polifonica, dato che il formattatore automatico che gestisce le collisioni tra pause non interviene su questo tipo di pause.

```
\relative { a'4\rest d4\rest }
```



Frammenti di codice selezionati

Stili di pausa

Esistono vari stili di pausa.

```
restsA = {
  r\maxima r\longa r\breve r1 r2 r4 r8 r16 s32
  s64 s128 s256 s512 s1024 s1024
}
restsB = {
```

```

r\maxima r\longa r\breve r1 r2 r4 r8 r16 r32
r64 r128 r256 r512 r1024 s1024
}

\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

  \override Staff.Rest.style = #'mensural
  <>^\markup \typewriter { mensural } \restsA \bar "" \break

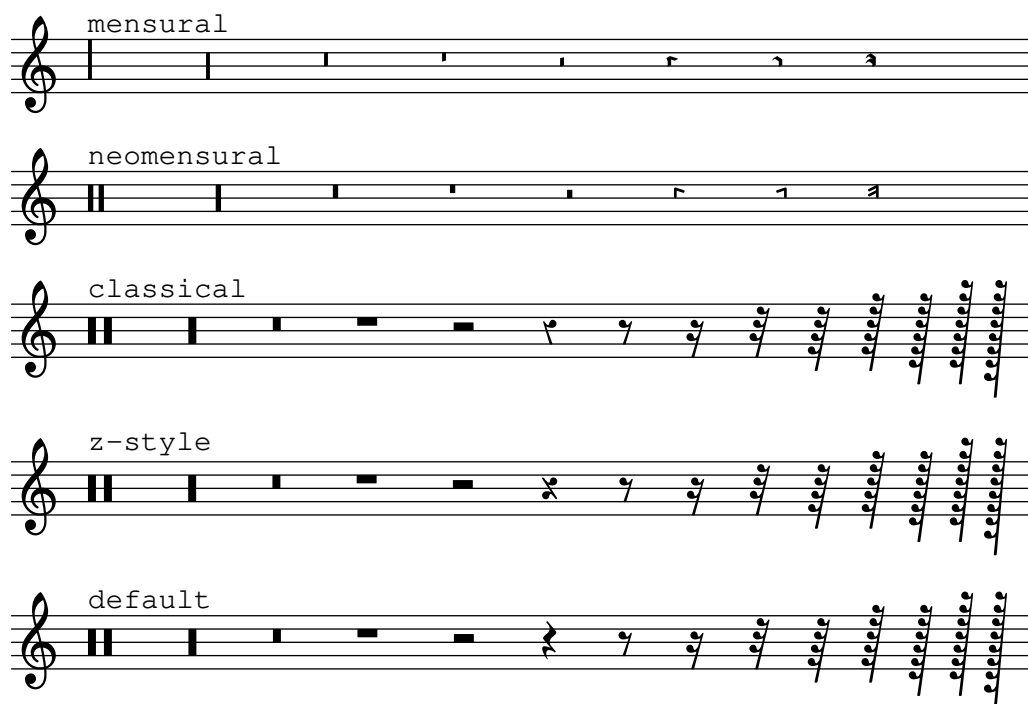
  \override Staff.Rest.style = #'neomensural
  <>^\markup \typewriter { neomensural } \restsA \bar "" \break

  \override Staff.Rest.style = #'classical
  <>^\markup \typewriter { classical } \restsB \bar "" \break

  \override Staff.Rest.style = #'z
  <>^\markup \typewriter { z-style } \restsB \bar "" \break

  \override Staff.Rest.style = #'default
  <>^\markup \typewriter { default } \restsB \bar "" \break
}

```



Vedi anche

Glossario musicale: Sezione “breve” in *Glossario Musicale*, Sezione “longa” in *Glossario Musicale*, Sezione “maxima” in *Glossario Musicale*.

Guida alla notazione: [Pause d’intero], pagina 66.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Rest” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Non c'è un limite massimo o minimo alla durata di una pausa, ma è il numero dei glifi ad essere limitato: si possono indicare pause da un centoventottesimo fino alla maxima (otto volte una semibreve).

Pause invisibili

Una pausa invisibile (chiamata anche ‘pausa spaziatrice’) si inserisce come una nota col nome s:

```
\relative c'' {
  c4 c s c |
  s2 c |
}
```



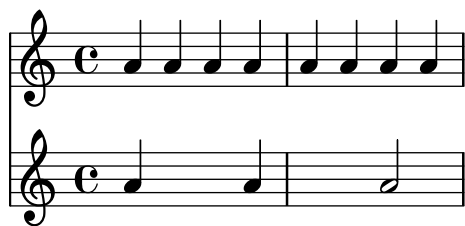
Le pause spaziatrici possono essere usate soltanto nella modalità note e nella modalità accordi. In altre situazioni, ad esempio quando si inserisce il testo vocale, si usa il comando `\skip` per saltare un valore musicale. `\skip` richiede una durata esplicita, ma questo requisito viene ignorato se il testo desume le proprie durate dalle note presenti in una melodia ad esso associata attraverso `\addlyrics` o `\lyricsto`.

```
<<
{
  a'2 \skip2 a'2 a'2
}
\new Lyrics {
  \lyricmode {
    foo2 \skip 1 bla2
  }
}
>>
```



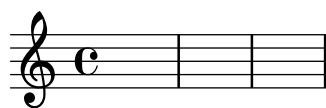
Dato che `\skip` è un comando, non modifica la durata predefinita delle note che seguono, diversamente da s.

```
<<
{
  \repeat unfold 8 { a'4 }
}
{
  a'4 \skip 2 a' |
  s2 a'
}
>>
```



Una pausa spaziatrice crea implicitamente i contesti Staff e Voice se non esistono già, proprio come accade per le note e le pause:

```
{ s1 s s }
```



`\skip` si limita a saltare un valore musicale, non crea nessun tipo di output.

```
% Questo input è corretto, ma non produce niente
\skip 1 \skip1 \skip 1
```

Vedi anche

Manuale di apprendimento: Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: [Note nascoste], pagina 240, Sezione 5.4.7 [Visibilità degli oggetti], pagina 652.

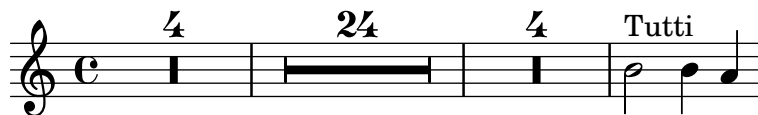
Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “SkipMusic” in *Guida al Funzionamento Interno*.

Pause d'intero

Le pause per una o più misure d'intero si inseriscono, come le note, col carattere maiuscolo R:

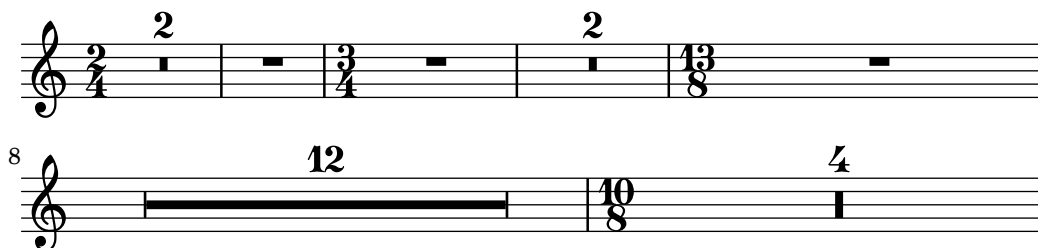
```
% L'insieme delle misure di pausa vengono riportate in una sola misura
\compressMMRests {
  R1*4
  R1*24
  R1*4
  b'2~"Tutti" b'4 a'4
}
```



La durata delle pause multiple è identica alla notazione di durata usata per le note e deve essere sempre un numero intero di misure/lunghezze, quindi occorre spesso usare dei punti di aumentazione o delle frazioni:

```
\compressMMRests {
  \time 2/4
  R1 | R2 |
  \time 3/4
  R2. | R2.*2 |
  \time 13/8
  R1*13/8 | R1*13/8*12 |
}
```

```
\time 10/8
R4*5*4 |
}
```



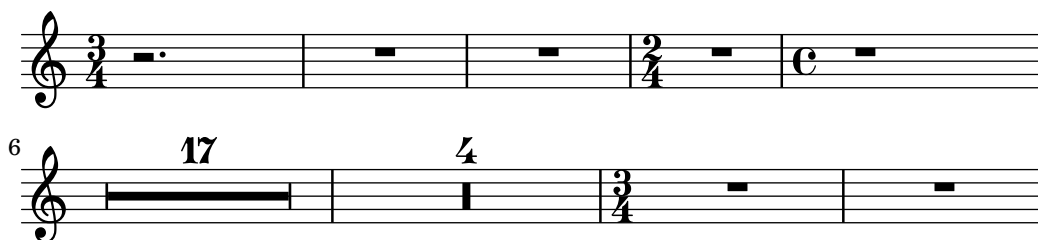
Una pausa d'intero appare al centro della misura con la durata di una semibreve o di una breve, in base all'indicazione di tempo.

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



Di norma una pausa multipla viene scorporata sul pentagramma in modo da mostrare esplicitamente tutte le misure per cui si prolunga. Altrimenti, è possibile indicarla collocando in una sola misura un simbolo di pausa multipla, col numero di misure per cui la pausa si prolunga posto al di sopra della misura stessa:

```
% Comportamento predefinito
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Tutte le misure di pausa sono riportate in una singola misura
\compressMMRests {
  r1 | R1*17 | R1*4 |
}
% Le misure della pausa multipla sono scorporate di nuovo
\time 3/4
R2.*2 |
```



Si possono aggiungere delle annotazioni alle pause multiple.

```
\compressMMRests {
  \time 3/4
  R2.*10~\markup { \italic "ad lib." }
}
```


}



Quando una pausa multipla segue immediatamente un comando `\partial`, potrebbero non apparire i relativi avvertimenti del controllo battuta.

Comandi predefiniti

`\textLengthOn`, `\textLengthOff`, `\compressMMRests`,

Frammenti di codice selezionati

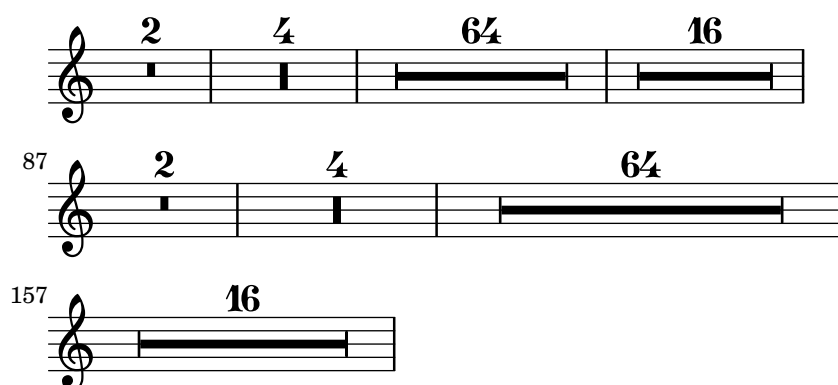
Regolazione della lunghezza delle pause multiple

Le pause multiple hanno una lunghezza che dipende dalla loro durata totale e tale lunghezza è regolata da `MultiMeasureRest.space-increment`. Nota che il valore predefinito è 2.0.

```
\relative c' {
  \omit Staff.TimeSignature
  \compressEmptyMeasures

  R1*2 R1*4 R1*64 R1*16 \break
  \override MultiMeasureRest.space-increment = 4
  R1*2 R1*4 R1*64 R1*16
}

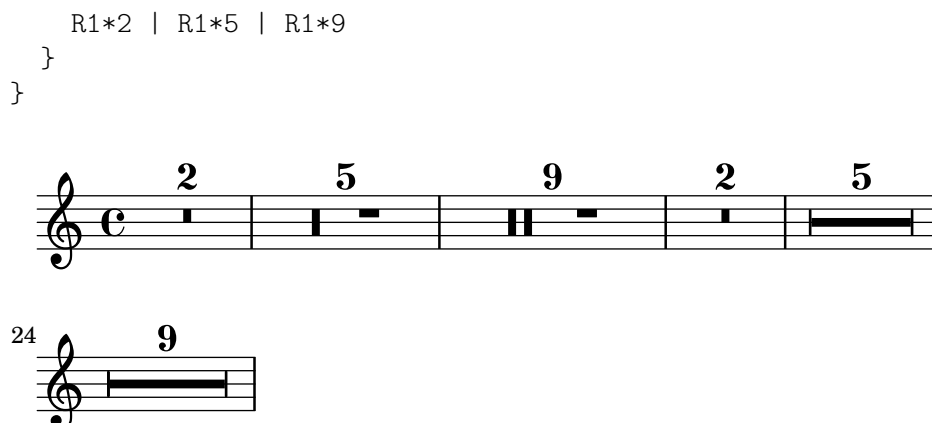
\layout {
  ragged-right = ##t
}
```



Modificare la forma delle pause multiple

Se la pausa multipla dura dieci misure o un numero inferiore a dieci, nel rigo apparirà una serie di pause di lunga e di breve (chiamate in tedesco “Kirchenpausen” - pause ecclesiastiche); altrimenti apparirà una semplice linea. Il numero predefinito di dieci può essere cambiato sovrascrivendo la proprietà `expand-limit`.

```
\relative c' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
  }
  \override MultiMeasureRest.expand-limit = 3
}
```



Posizionamento delle pause multiple

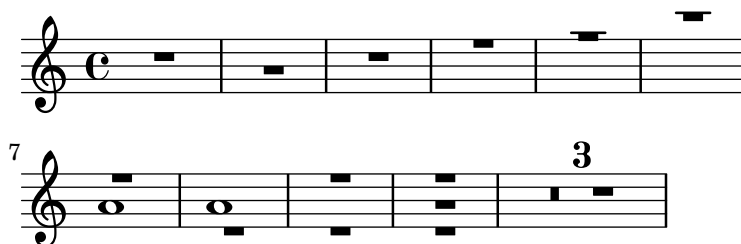
Diversamente dalle pause normali, non esiste un comando predefinito per cambiare la posizione sul rigo di un simbolo di pausa multipla di qualsiasi tipo connettendolo a una nota. Tuttavia, nella musica polifonica le pause multiple nelle voci dispari e pari sono separate verticalmente. Il posizionamento delle pause multiple si controlla nel modo seguente:

```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line.
  R1
  % They can be moved using an override or tweak.
  \tweak staff-position -2 R1
  \tweak staff-position 0 R1
  \tweak staff-position 2 R1
  \override MultiMeasureRest.staff-position = 3 R1
  \override MultiMeasureRest.staff-position = 6 R1
  \revert MultiMeasureRest.staff-position
  \break

  % Odd-numbered voices are under the top line.
  << { R1 } \\\ { a1 } >>
  % Even-numbered voices are under the bottom line.
  << { a1 } \\\ { R1 } >>
  % Multi-measure rests in both voices remain separate.
  << { R1 } \\\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override or tweak.
  << { R1 } \\\ { R1 } \\\ { \tweak staff-position -2 R1 } >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed.
  \compressMMRests
  <<
    \revert MultiMeasureRest.direction
    { R1*3 } \\\
    \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```

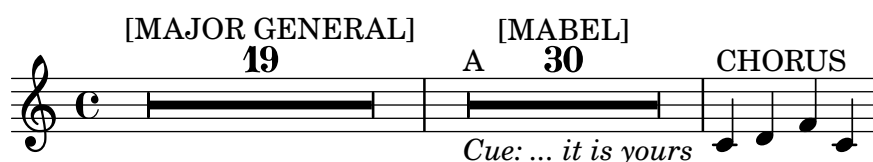


Testo a margine delle pause multiple

Il testo a margine di una pausa multipla viene centrato sopra o sotto di essa. Se il testo è lungo, la misura non si espanderà. Per espandere la pausa multipla in modo che si allinei col testo, conviene usare un accordo vuoto con del testo attaccato prima della pausa multipla.

Il testo così attaccato a una nota spaziatrice viene allineato a sinistra della posizione in cui la nota sarebbe posta nella misura, ma se la lunghezza della misura è determinata dalla lunghezza del testo, il testo verrà centrato.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_^\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



Vedi anche

Glossario musicale: Sezione “pausa multipla” in *Glossario Musicale*.

Guida alla notazione: [Durata], pagina 49, Sezione 1.8 [Testo], pagina 253, Sezione 1.8.2 [Formattazione del testo], pagina 260, [Scritte], pagina 253.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “MultiMeasureRest” in *Guida al Funzionamento Interno*, Sezione “MultiMeasureRestNumber” in *Guida al Funzionamento Interno*, Sezione “MultiMeasureRestScript” in *Guida al Funzionamento Interno*, Sezione “MultiMeasureRest-Text” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Se una ditekkiatura viene posta su una pausa multipla (ad esempio R1*10-4), il numero della ditekkiatura può collidere col numero del contatore delle battute.

Non è possibile condensare automaticamente molteplici pause normali in in una singola pausa multipla.

Le pause multiple non considerano le collisioni di pausa.

1.2.3 Aspetto dei ritmi

Indicazione di tempo

L'indicazione di tempo si imposta così:

```
\time 2/4 c''2
\time 3/4 c''2.
```



I cambi di indicazione di tempo a metà misura sono trattati in [Anacrusi], pagina 78.

Le indicazioni di tempo appaiono all'inizio di un brano e ogni volta che l'indicazione cambia. Se il cambio ha luogo alla fine di un rigo, appare un'indicazione di tempo di precauzione. Si può modificare questo comportamento predefinito, come è spiegato in Sezione 5.4.7 [Visibilità degli oggetti], pagina 652.

```
\relative c'' {
  \time 2/4
  c2 c
  \break
  c c
  \break
  \time 4/4
  c c c c
}
```



Il simbolo di indicazione di tempo usato nei tempi 2/2 e 4/4 può essere sostituito da un numero:

```
\relative c'' {
  % Stile predefinito
  \time 4/4 c1
  \time 2/2 c1
  % Passaggio allo stile numerico
  \numericTimeSignature
  \time 4/4 c1
  \time 2/2 c1
  % Ritorno allo stile predefinito
  \defaultTimeSignature
  \time 4/4 c1
  \time 2/2 c1
}
```

}



Le indicazioni di tempo mensurali sono trattate in [Indicazioni di tempo mensurali], pagina 460.

Oltre a impostare l'indicazione di tempo che appare nel pentagramma, il comando `\time` imposta anche i valori delle proprietà basate sull'indicazione di tempo, ovvero `beatBase`, `beatStructure` e `beamExceptions`. I valori predefiniti di queste proprietà si trovano in `scm/time-signature-settings.scm`.

Si può sovrascrivere il valore predefinito di `beatStructure` nel comando `\time` stesso specificandolo come primo argomento opzionale:

```
\score {
  \new Staff {
    \relative {
      \time 2,2,3 7/8
      \repeat unfold 7 { c'8 } |
      \time 3,2,2 7/8
      \repeat unfold 7 { c8 } |
    }
  }
}
```



Oppure si possono impostare tutti i valori predefiniti di queste variabili relative all'indicazione di tempo, incluse `beatBase` e `beamExceptions`. I valori possono essere impostati in modo indipendente per diverse indicazioni di tempo. I nuovi valori hanno effetto appena viene eseguito un nuovo comando `\time` che abbia lo stesso valore dell'indicazione di tempo specificata nelle nuove impostazioni:

```
\score {
  \new Staff {
    \relative c' {
      \overrideTimeSignatureSettings
      4/4          % timeSignature
      #1/4         % beatBase
      3,1          % beatStructure
      #'()         % beamExceptions
      \time 4/4
      \repeat unfold 8 { c8 } |
    }
  }
}
```



`\overrideTimeSignatureSettings` prende quattro argomenti:

1. *timeSignature*, una frazione che indica l'indicazione di tempo a cui questi valori si riferiscono.
2. *beatBase*, una frazione che contiene il numeratore e il denominatore dell'unità di tempo.
3. *beatStructure*, una lista Scheme che indica la struttura dei battiti nella misura, nell'unità di *beatBase*.
4. *beamExceptions*, una lista di associazione (*alist*) che contiene regole di disposizione delle travature che vanno oltre la fine ad ogni battito, come descritto in [Impostare il comportamento delle travature automatiche], pagina 91.

I valori modificati delle proprietà predefinite dell'indicazione di tempo possono essere ripristinati ai valori originali:

```
\score {
  \relative {
    \repeat unfold 8 { c'8 } |
    \overrideTimeSignatureSettings
      4/4      % timeSignature
      #1/4     % beatBase
      3,1      % beatStructure
      #'()     % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings 4/4
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}
```



Si possono stabilire valori diversi delle proprietà predefinite dell'indicazione di tempo per righe diversi spostando `Timing_translator` e `Default_bar_line_engraver` dal contesto `Score` al contesto `Staff`.

```
\score {
  \new StaffGroup <<
    \new Staff {
      \overrideTimeSignatureSettings
        4/4      % timeSignature
        #1/4     % beatBase
        3,1      % beatStructure
        #'()     % beamExceptions
      \time 4/4
      \repeat unfold 8 {c''8}
    }
    \new Staff {
      \overrideTimeSignatureSettings
        4/4      % timeSignature
        #1/4     % beatBase
        1,3      % beatStructure
        #'()     % beamExceptions
    }
  }
}
```

```

        \time 4/4
        \repeat unfold 8 {c''8}
    }
>>
\layout {
  \context {
    \Score
    \remove Timing_translator
  }
  \context {
    \Staff
    \consists Timing_translator
  }
}

```



Un ulteriore metodo per modificare queste variabili relative all'indicazione di tempo, che evita di mostrare di nuovo l'indicazione di tempo al momento del cambio, è descritto in [Impostare il comportamento delle travature automatiche], pagina 91.

Comandi predefiniti

`\numericTimeSignature`, `\defaultTimeSignature`.

Frammenti di codice selezionati

Indicazione di tempo che mostra solo il numeratore (invece della frazione)

Talvolta un'indicazione di tempo non deve mostrare la frazione intera (ad esempio 7/4), ma solo il numeratore (numero 7 in questo caso). Si può ottenere facilmente con `\override Staff.TimeSignature.style = #'single-number`, che cambia lo stile in modo permanente. Con `\revert Staff.TimeSignature.style`, questa impostazione può essere annullata. Per applicare lo stile a cifra singola (single-number) a una sola indicazione di tempo, si usa il comando `\override` preceduto da `\once`.

```

\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-number
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
}

```

```

c4 c
% single-number style only for the next time signature
\tweak style #'single-number \time 5/4
c4 c c c c
\time 2/4
c4 c
}

```



Vedi anche

Glossario musicale: Sezione “indicazione di tempo” in *Glossario Musicale*

Guida alla notazione: [Indicazioni di tempo mensurali], pagina 460, [Impostare il comportamento delle travature automatiche], pagina 91, [Gestione del tempo], pagina 128.

File installati: scm/time-signature-settings.scm.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TimeSignature” in *Guida al Funzionamento Interno*, Sezione “Timing-translator” in *Guida al Funzionamento Interno*.

Indicazioni metronomiche

Un’indicazione metronomica è semplice da scrivere:

```

\relative {
  \tempo 4 = 120
  c'2 d
  e4. d8 c2
}

```



Le indicazioni metronomiche si possono rappresentare anche come una gamma di due numeri:

```

\relative {
  \tempo 4 = 40 - 46
  c'4. e8 a4 g
  b,2 d4 r
}

```

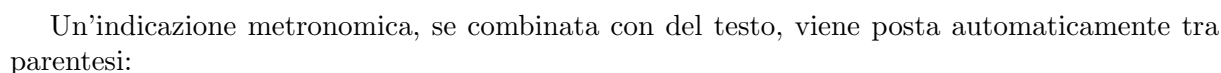


Al loro posto si possono usare delle indicazioni di tempo testuali:

```

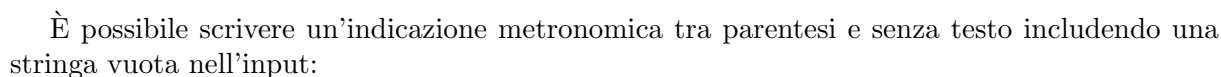
\relative {
  \tempo "Allegretto"
  c'4 e d c
  b4. a16 b c4 r4
}

```

Allegro (♩ = 160)

```
\relative {
  \tempo \markup { \i Faster } 4 = 132
  a'8-. r8 b-. r gis-. r a-. r
}
```

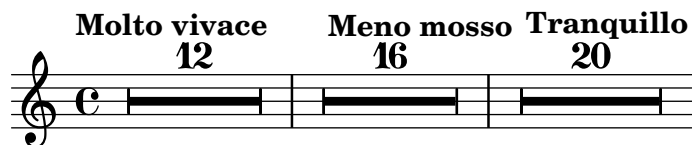


(♩ = 96)



```
\compressMMRests {
  \markLengthOn
  \tempo "Molto vivace"
  R1*12
  \tempo "Meno mosso"
  R1*16
  \markLengthOff
}
```

```
\tempo "Tranquillo"
R1*20
}
```



Frammenti di codice selezionati

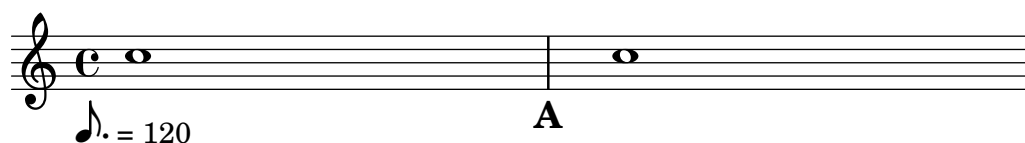
Posizionare il metronomo e i numeri di chiamata sotto il rigo

Di norma, il metronomo e i numeri di chiamata vengono posizionati sopra il rigo. Per metterli sotto il rigo basta impostare correttamente la proprietà `direction` di `MetronomeMark` o `RehearsalMark`.

```
\layout {
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Modificare il tempo senza mostrare l'indicazione metronomica

Per cambiare il tempo del file MIDI senza che appaia l'indicazione metronomica, basta renderla invisibile.

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Creare indicazioni metronomiche in modalità testuale

Si possono creare nuove indicazioni metronomiche in modalità testuale, ma non modificheranno il tempo del file MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
        \smaller \general-align #Y #DOWN \note { 16. } #UP
        " = "
        \smaller \general-align #Y #DOWN \note { 8 } #UP
      )
    }
  }
  c1
  c4 c' c,2
}
```



I dettagli si trovano in Sezione 1.8.2 [Formattazione del testo], pagina 260.

Vedi anche

Glossario musicale: Sezione “metronomo” in *Glossario Musicale*, Sezione “indicazione di tempo” in *Glossario Musicale*, Sezione “indicazione metronomica” in *Glossario Musicale*.

Guida alla notazione: Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione 3.5 [Creazione dell’output MIDI], pagina 537.

Frammenti di codice: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “MetronomeMark” in *Guida al Funzionamento Interno*.

Anacrusi

Le misure parziali, come l’*anacrusi* o la battuta in levare, si inseriscono col comando `\partial`:

```
\partial durata
```

Quando si usa `\partial` all’inizio di una partitura, la *durata* è la lunghezza della musica che precede la prima battuta.

```
\relative {
  \time 3/4
  \partial 4.
  r4 e'8 | a4 c8 b c4 |
}
```



Quando si usa `\partial` dopo l'inizio di una partitura, la *durata* è la lunghezza *rimanente* della misura corrente. Non crea una battuta con un nuovo numero.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
    #end-of-line-invisible
  \time 9/8
  d' '4.~ 4 d8 d( c) b | c4.~ 4. \bar "||"
  \time 12/8
  \partial 4.
  c8( d) e | f2.~ 4 f8 a,( c) f |
}
```



Il comando `\partial` è *obbligatorio* quando l'indicazione di tempo cambia in mezzo a una misura, ma si può usare anche da solo.

```
\relative {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \override Score.BarNumber.break-visibility =
    #end-of-line-invisible
  \time 6/8
  \partial 8
  e'8 | a4 c8 b[ c b] |
  \partial 4
  r8 e,8 | a4 \bar "||"
  \partial 4
  r8 e8 | a4
  c8 b[ c b] |
}
```



Il comando `\partial` imposta la proprietà `Timing.measurePosition`, che è un numero razionale che indica quanto tempo della misura è trascorsa.

Vedi anche

Glossario musicale: Sezione “anacrusi” in *Glossario Musicale*.

Guida alla notazione: [Abbellimenti], pagina 122.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Timing-translator” in *Guida al Funzionamento Interno*.

Musica in tempo libero

Nella musica in un tempo determinato l'inserimento delle stanghette e dei numeri di battuta è calcolato automaticamente. Nella musica in tempo libero (per esempio, la cadenza), un simile

comportamento non è desiderabile, e può essere ‘disabilitato’ col comando `\cadenzaOn` e poi ‘riabilitato’ quando necessario con `\cadenzaOff`.

```
\relative c'' {
  c4 d e d
  \cadenzaOn
  c4 c d8[ d d] f4 g4.
  \cadenzaOff
  \bar "|"
  d4 e d c
}
```



La numerazione delle battute riprende dopo la cadenza.

```
\relative c'' {
  % Mostra tutti i numeri di battuta
  \override Score.BarNumber.break-visibility = #all-visible
  c4 d e d
  \cadenzaOn
  c4 c d8[ d d] f4 g4.
  \cadenzaOff
  \bar "|"
  d4 e d c
}
```



Se si inserisce un comando `\bar` dentro una cadenza non viene iniziata una nuova misura, anche se appare una stanghetta nell’output. Quindi qualsiasi alterazione, che di solito si considera sempre attiva fino alla fine della misura, sarà ancora valida dopo la stanghetta stampata da `\bar`. Se si desidera che le alterazioni successive appaiano, si dovranno inserire manualmente delle alterazioni forzate o di precauzione, come è spiegato in [Alterazioni], pagina 6.

```
\relative c'' {
  c4 d e d
  \cadenzaOn
  cis4 d cis d
  \bar "|"
  % Il primo cis viene stampato senza alterazione anche se si trova dopo \bar
  cis4 d cis! d
  \cadenzaOff
  \bar "|"
}
```



La disposizione automatica delle travature viene disabilitata da `\cadenzaOn`. Quindi tutte le travature nelle cadenze devono essere inserite manualmente. Si veda [Travature manuali], pagina 101.

```

\relative {
  \repeat unfold 8 { c''8 }
  \cadenzaOn
  cis8 c c c c
  \bar""|
  c8 c c
  \cadenzaOff
  \repeat unfold 8 { c8 }
}

```



Questi comandi predefiniti hanno effetto su tutti i rigi di una partitura, anche quando inseriti in un solo contesto Voice. Per modificare questo comportamento, si sposta `Timing_translator` dal contesto Score al contesto Staff. Si veda [Notazione polimetrica], pagina 81.

Comandi predefiniti

`\cadenzaOn`, `\cadenzaOff`.

Vedi anche

Glossario musicale: Sezione “cadenza” in *Glossario Musicale*.

Guida alla notazione: Sezione 5.4.7 [Visibilità degli oggetti], pagina 652, [Notazione polimetrica], pagina 81, [Travature manuali], pagina 101, [Alterazioni], pagina 6.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Problemi noti e avvertimenti

Le interruzioni automatiche di linea e di pagina possono aver luogo solo dopo una stanghetta di battuta; quindi, per consentire delle interruzioni nei lunghi passaggi di musica in tempo libero è necessario inserire manualmente delle stanghette ‘invisibili’:

```
\bar ""
```

Notazione polimetrica

La notazione polimetrica è supportata esplicitamente o tramite la modifica manuale del simbolo d’indicazione di tempo (e la trasformazione della durata delle note).

Diverse indicazioni di tempo con misure di uguale lunghezza

Si sceglie una normale indicazione di tempo per ogni rigo e si imposta `timeSignature` sulla frazione desiderata. Quindi si usa la funzione `\scaleDurations` per scalare la durata delle note di ogni rigo in modo che rientrino nella comune indicazione di tempo.

L’esempio seguente presenta simultaneamente musica con indicazioni di tempo di $3/4$, $9/8$ e $10/8$. Nel secondo rigo le durate appaiono come moltiplicate per $2/3$ (perché $2/3 * 9/8 = 3/4$), mentre nel terzo rigo le durate appaiono come moltiplicate per $3/5$ (perché $3/5 * 10/8 = 3/4$). È possibile che si debbano inserire a mano le travature, perché la scalatura delle durate influenzerà le regole della disposizione automatica delle travature.

```

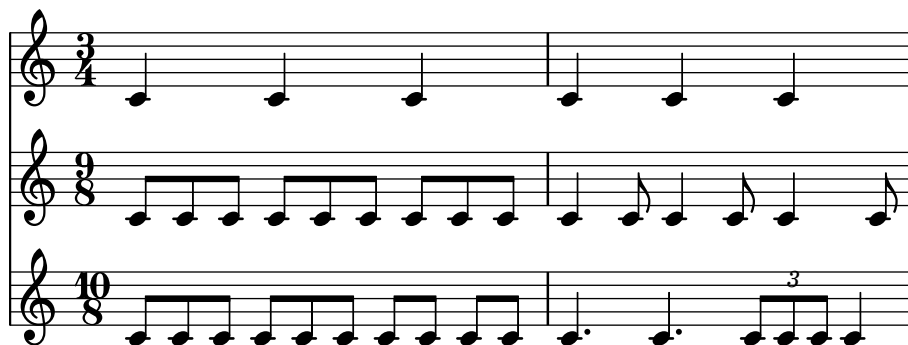
\relative <<
  \new Staff {
    \time 3/4
    c'4 c c |
    c4 c c |
  }

```

```

}
\new Staff {
  \time 3/4
  \set Staff.timeSignature = 9/8
  \scaleDurations 2/3 {
    \repeat unfold 3 { c8[ c c] }
    \repeat unfold 3 { c4 c8 }
  }
}
\new Staff {
  \time 3/4
  \set Staff.timeSignature = 10/8
  \scaleDurations 3/5 {
    \repeat unfold 2 { c8[ c c] }
    \repeat unfold 2 { c8[ c] } |
    c4. c \tuplet 3/2 { c8[ c c] } c4
  }
}
>>

```



Diverse indicazioni di tempo con misure di lunghezza differenti

Si può dare a ogni rigo la sua indicazione di tempo spostando `Timing_translator` e `Default_bar_line_engraver` nel contesto `Staff`.

```

\layout {
  \context {
    \Score
    \remove Timing_translator
  }
  \context {
    \Staff
    \consists Timing_translator
  }
}

```

% Ora ogni rigo ha la sua indicazione di tempo.

```

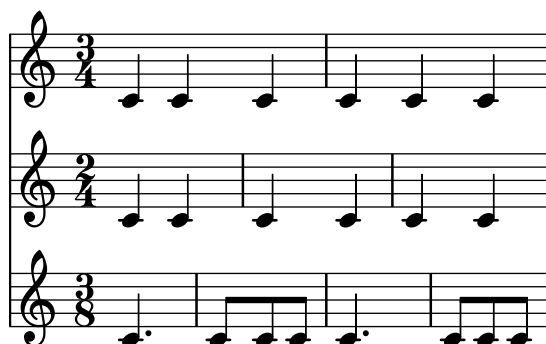
\relative <<
\new Staff {
  \time 3/4
  c'4 c c |
  c4 c c |
}

```

```

}
\new Staff {
  \time 2/4
  c4 c |
  c4 c |
  c4 c |
}
\new Staff {
  \time 3/8
  c4. |
  c8 c c |
  c4. |
  c8 c c |
}
}
>>

```



Indicazioni di tempo composto

Si creano con la funzione `\timeAbbrev`. La sintassi è:

```
\timeAbbrev #'(lista di liste)
```

La struttura più semplice è una singola lista, dove l'*ultimo* numero indica il numero inferiore dell'indicazione di tempo e i numeri precedenti indicano i numeri superiori del segno di tempo.

```

\relative {
  \timeAbbrev #'((2 2 2 8))
  \repeat unfold 6 c'8 \repeat unfold 12 c16
}

```



Si possono costruire tempi più complessi tramite ulteriori liste. Le modalità di disposizione automatica delle travature varieranno a seconda di questi valori.

```

\relative {
  \timeAbbrev #'((1 4) (3 8))
  \repeat unfold 5 c'8 \repeat unfold 10 c16
}

```

```

\relative {
  \timeAbbrev #'((1 2 3 8) (3 4))
  \repeat unfold 12 c'8
}

```




Vedi anche

Glossario musicale: Sezione “polimetrico” in *Glossario Musicale*, Sezione “indicazione di tempo polimetrico” in *Glossario Musicale*, Sezione “tempo” in *Glossario Musicale*.

Guida alla notazione: [Travature automatiche], pagina 88, [Travature manuali], pagina 101, [Indicazione di tempo], pagina 71, [Scalare le durate], pagina 57.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TimeSignature” in *Guida al Funzionamento Interno*, Sezione “Timing_translator” in *Guida al Funzionamento Interno*, Sezione “Staff” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Sebbene le note che si presentano nello stesso momento in ciascuno dei vari righe siano poste nello stesso punto orizzontale, le stanghette (in ciascun rigo) potrebbero causare una spaziatura difforme in ciascuna delle diverse indicazioni di tempo.

L’uso di un blocco midi con notazione polimetrica potrebbe causare avvertimenti indesiderati di controllo di battuta. In questo caso, spostare `Timing_translator` dal contesto `Score` al contesto `Staff` all’interno del blocco midi.

```
\midi {
  \context {
    \Score
    \remove Timing_translator
  }
  \context {
    \Staff
    \consists Timing_translator
  }
}
```

Divisione automatica delle note

Le note le cui durate eccedono il valore della battuta possono essere convertite automaticamente in note con legature di valore a cavallo delle stanghette sostituendo l’incisore `Note_heads_engraver` con `Completion_heads_engraver`. Analogamente, le pause le cui durate eccedono il valore della battuta possono essere divise automaticamente sostituendo `Rest_engraver` con `Completion_rest_engraver`. Nell’esempio seguente, le note e le pause che eccedono la durata di battuta vengono divise e le note sono anche connesse con legature di valore a cavallo della stanghetta.

```
\new Voice \with {
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
  \remove Rest_engraver
  \consists Completion_rest_engraver
}
```

```
\relative {
  c'2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 r1*2
}
```



Questi incisori dividono tutte le note e le pause in corrispondenza della stanghetta e inseriscono le legature di valore. Uno dei suoi usi possibili è la verifica di partiture complesse: se le misure non sono riempite interamente, le legature di valore mostrano esattamente di quanto è ecceduta ogni misura.

La proprietà `completionUnit` imposta la durata preferita per le note divise.

```
\new Voice \with {
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
} \relative {
  \time 9/8 g\breve. d''4. \bar "||"
  \set completionUnit = #3/8
  g\breve. d4.
}
```



Questi incisori dividono le note che hanno una durata ridimensionata, come quelle dei gruppi irregolari, in note con lo stesso fattore di ridimensionamento della nota di input.

```
\new Voice \with {
  \remove Note_heads_engraver
  \consists Completion_heads_engraver
} \relative {
  \time 2/4 r4
  \tuplet 3/2 {g'4 a b}
  \scaleDurations 2/3 {g a b}
  g4*2/3 a b
  \tuplet 3/2 {g4 a b}
  r4
}
```



Vedi anche

Glossario musicale: Sezione “legatura di valore” in *Glossario Musicale*

Manuale di apprendimento: Sezione “Gli incisori” in *Manuale di Apprendimento*, Sezione “Aggiungere e togliere gli incisori” in *Manuale di Apprendimento*.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “Completion_heads_engraver” in *Guida al Funzionamento Interno*, Sezione

“Rest_engraver” in *Guida al Funzionamento Interno*, Sezione “Completion_rest_engraver” in *Guida al Funzionamento Interno*, Sezione “Forbid_line_break_engraver” in *Guida al Funzionamento Interno*.


Problemi noti e avvertimenti

In accordo col comportamento precedente, note e pause la cui durata è più lunga di una misura, come `c1*2`, sono divise in note prive di fattore di ridimensionamento, `{ c1 c1 }`. La proprietà `completionFactor` controlla questo comportamento e impostandola su `#f` fa sì che le note e le pause divise abbiano il fattore di ridimensionamento delle durate di input.

Mostrare i ritmi della melodia

È possibile mostrare soltanto il ritmo di una melodia usando il rigo ritmico. Tutte le altezze delle note su tale rigo sono appiattite e il rigo stesso ha una sola linea

```
<<
\new RhythmicStaff {
  \new Voice = "myRhythm" \relative {
    \time 4/4
    c'4 e8 f g2
    r4 g g f
    g1
  }
}
\new Lyrics {
  \lyricsto "myRhythm" {
    This is my song
    I like to sing
  }
}
>>
```

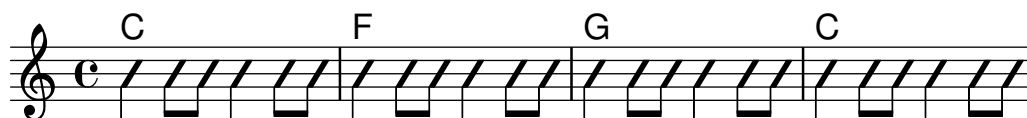


This is my song I like to sing

I diagrammi degli accordi per chitarra di solito mostrano i ritmi di accompagnamento. Si possono visualizzare usando l'incisore `Pitch_squash_engraver` e il comando `\improvisationOn`.

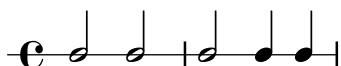
```
<<
\new ChordNames {
  \chordmode {
    c1 f g c
  }
}
\new Voice \with {
  \consists Pitch_squash_engraver
} \relative c'' {
  \improvisationOn
  c4 c8 c c4 c8 c
  f4 f8 f f4 f8 f
  g4 g8 g g4 g8 g
  c4 c8 c c4 c8 c
}
>>
```

>>



La musica contenente accordi può essere usata anche come input per `RhythmicStaff` e per essere usata con l'incisore `Pitch_squash_engraver` se gli accordi vengono prima ridotti a note singole con la funzione musicale `\reduceChords`:

```
\new RhythmicStaff {
  \time 4/4
  \reduceChords {
    <c>2
    <e>2
    <c e g>2
    <c e g>4
    <c e g>4
  }
}
```



Comandi predefiniti

`\improvisationOn`, `\improvisationOff`, `\reduceChords`.

Frammenti di codice selezionati

Ritmi di accompagnamento per chitarra

Per la musica per chitarra, è possibile mostrare i ritmi di accompagnamento, insieme alle note della melodia e ai nomi e ai diagrammi degli accordi.

```
\include "predefined-guitar-fretboards.ly"
```

<<

```
\new ChordNames \chordmode {
  c1 | f | g | c
}
\new FretBoards \chordmode {
  c1 | f | g | c
}
\new Voice \with {
  \consists "Pitch_squash_engraver"
} \relative c'' {
  \improvisationOn
  c4 c8 c c4 c8 c
  f4 f8 f f4 f8 f
  g4 g8 g g4 g8 g
  c4 c8 c c4 c8 c
}
\new Voice = "melody" \relative c'' {
  c2 e4 e4
  f2. r4
```

```

g2. a4
e4 c2.
}
\new Lyrics \lyricsto "melody" {
  This is my song.
  I like to sing.
}
>>

```

The image displays a musical score for the lyrics "This is my song. I like to sing." The score is written in C major and 2/4 time. It features a melody line in the treble clef and a guitar accompaniment in the bass clef. The guitar part includes three chords: C (3 2 1), F (1 3 4 2 1 1), and G (2 1 3). The melody line consists of eighth notes and quarter notes. The lyrics are placed below the notes.

Vedi anche

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “RhythmicStaff” in *Guida al Funzionamento Interno*, Sezione “Pitch_squash_engraver” in *Guida al Funzionamento Interno*.

1.2.4 Travature

Travature automatiche

Le travature sono inserite automaticamente:

```

\relative c'' {
  \time 2/4 c8 c c c
  \time 6/8 c8 c c c8. c16 c8
}

```

The image shows a musical score with two measures. The first measure is in 2/4 time and contains four eighth notes. The second measure is in 6/8 time and contains a quarter note, an eighth note, and a dotted quarter note. Automatic bar lines (travature) are placed at the end of each measure.

Se queste impostazioni automatiche non sono soddisfacenti, si può definire esplicitamente la disposizione delle travature, come è spiegato in [Travature manuali], pagina 101. Le travature *devono* essere inserite manualmente se devono estendersi oltre le pause.

La disposizione automatica delle travature, se non necessaria, può essere disabilitata con `\autoBeamOff` e riabilitata con `\autoBeamOn`:

```
\relative c' {
  c4 c8 c8. c16 c8. c16 c8
  \autoBeamOff
  c4 c8 c8. c16 c8.
  \autoBeamOn
  c16 c8
}
```



Nota: Se si usano le travature per indicare i melismi nelle parti vocali, occorre disabilitare la disposizione automatica delle travature con `\autoBeamOff` e le travature devono essere indicate manualmente. L'uso di `\partCombine` insieme a `\autoBeamOff` può produrre risultati imprevisti. Si vedano i frammenti di codice per avere maggiori informazioni.

Si possono creare dei modelli di disposizione delle travature diversi da quelli automatici predefiniti, come è spiegato in [Impostare il comportamento delle travature automatiche], pagina 91.

Comandi predefiniti

`\autoBeamOff`, `\autoBeamOn`.

Frammenti di codice selezionati

Travature che attraversano le interruzioni di linea

Le interruzioni di linea sono di norma proibite quando le travature attraversano la stanghetta di una battuta. Si può cambiare questo comportamento nel modo seguente:

```
music = {
  \repeat unfold 8 c8
  c8 \repeat unfold 7 { c[ c] } c
  \repeat unfold 8 c8
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable set to \typewriter "#t" }
  \override Beam.breakable = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Beam.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}
```

}

Beam.breakable set to #t

3

Beam.breakable not set

2

4

Modificare la distanza delle travature angolari

Le travature angolari vengono inserite automaticamente quando viene rilevata un'ampia distanza tra le teste di nota. Questo comportamento può essere regolato attraverso la proprietà `auto-knee-gap`. Viene disegnata una travatura angolare se la distanza è più grande del valore di `auto-knee-gap` più la larghezza della travatura (che dipende dalla durata delle note e dall'inclinazione della travatura). Il valore predefinito di `auto-knee-gap` è 5.5 spazi rigo.

```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```

Partcombine and \autoBeamOff

The function of `\autoBeamOff` when used with `\partCombine` can be difficult to understand. It may be preferable to use

```
\set Staff.autoBeaming = ##f
```

instead to ensure that auto-beaming is turned off for the entire staff. Use this at a spot in your score where no beam generated by the auto-beamer is still active.

Internally, `\partCombine` works with four voices – up-stem single, down-stem single, combined, and solo. In order to use `\autoBeamOff` to stop all auto-beaming when used with `\partCombine`, it is necessary to use *four* calls to `\autoBeamOff`.

```
{
  % \set Staff.autoBeaming = ##f % turns off all auto-beaming

  \partCombine {
```

```

\autoBeamOff % applies to split up-stems
\repeat unfold 4 a'16
% \autoBeamOff % applies to combined stems
\repeat unfold 4 a'8
\repeat unfold 4 a'16
% \autoBeamOff % applies to solo
\repeat unfold 4 a'16
r4
} {
% \autoBeamOff % applies to split down-stems
\repeat unfold 4 f'8
\repeat unfold 8 f'16 |
r4
\repeat unfold 4 a'16
}
}

```



Vedi anche

Guida alla notazione: [Travature manuali], pagina 101, [Impostare il comportamento delle travature automatiche], pagina 91.

File installati: scm/auto-beam.scm.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Auto_beam_engraver” in *Guida al Funzionamento Interno*, Sezione “Beam_engraver” in *Guida al Funzionamento Interno*, Sezione “Beam” in *Guida al Funzionamento Interno*, Sezione “BeamEvent” in *Guida al Funzionamento Interno*, Sezione “BeamForbidEvent” in *Guida al Funzionamento Interno*, Sezione “beam-interface” in *Guida al Funzionamento Interno*, Sezione “unbreakable-spanner-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le proprietà di una travatura sono determinate all’inizio della sua costruzione e qualsiasi ulteriore modifica alle sue proprietà che venga fatta prima che la travatura sia stata completata non avrà effetto finché non inizia la *successiva*, nuova travatura.

Impostare il comportamento delle travature automatiche

Quando la disposizione automatica delle travature è abilitata, la disposizione delle travature è determinata da tre proprietà di contesto: `beatBase`, `beatStructure` e `beamExceptions`. I valori predefiniti di queste variabili possono essere sovrascritti, come vedremo tra breve, oppure si possono anche cambiare i valori predefiniti stessi, come è spiegato in [Indicazione di tempo], pagina 71.

Se è definita una regola `beamExceptions` per l’indicazione di tempo corrente, tale regola soltanto determina la disposizione delle travature; i valori di `beatBase` e `beatStructure` vengono ignorati. Se non è definita alcuna regola `beamExceptions` per l’indicazione di tempo corrente, la disposizione delle travature è determinata dai valori di `beatBase` e `beatStructure`.

Disposizione delle travature basata su beatBase e beatStructure

Dato che le indicazioni di tempo più comuni hanno delle regole `beamExceptions` già definite, occorre disabilitarle se la disposizione automatica deve basarsi su `beatBase` e `beatStructure`. Le regole `beamExceptions` si disabilitano con questo comando

```
\set Timing.beamExceptions = #'()
```

Quando `beamExceptions` è impostato su `#'()`, o per impostazione esplicita o perché non sono state definite internamente le `beamExceptions` per l'indicazione di tempo corrente, le estremità delle travature si trovano sulle suddivisioni come specificato dalle proprietà di contesto `beatBase` e `beatStructure`. `beatStructure` è una lista *Scheme* che definisce la lunghezza di ogni suddivisione in rapporto alla misura in unità di `beatBase`. Per impostazione predefinita, `beatBase` è uno fratto il denominatore dell'indicazione di tempo e ogni unità di `beatBase` corrisponde a una singola suddivisione.

Per ogni indicazione di tempo esistono valori separati per `beatStructure` e `beatBase`. Le modifiche di queste variabili hanno effetto solo sulle indicazioni di tempo attive, dunque tali modifiche devono essere poste dopo il comando `\time` che inizia una nuova indicazione di tempo, non prima. I nuovi valori assegnati a una certa indicazione di tempo sono mantenuti e reintrodotti ogni volta che quell'indicazione di tempo viene ristabilita.

```
\relative c' {
  \time 5/16
  c16^"predefinito" c c c c |
  % È improbabile che per un tempo di 5/16 sia stata definita beamExceptions,
  % ma disabilitiamola lo stesso per sicurezza
  \set Timing.beamExceptions = #'()
  \set Timing.beatStructure = 2,3
  c16^(2+3)" c c c c |
  \set Timing.beatStructure = 3,2
  c16^(3+2)" c c c c |
}
```



```
\relative {
  \time 4/4
  a'8^"predefinito" a a a a a a
  % Disabilita beamExceptions perché è senz'altro definita
  % per il tempo 4/4
  \set Timing.beamExceptions = #'()
  \set Timing.beatBase = #1/4
  \set Timing.beatStructure = 1,1,1,1
  a8^"cambiato" a a a a a a
}
```



Le modifiche alle impostazioni delle travature possono essere limitate a contesti specifici. Se non si specifica alcuna impostazione in un contesto di livello più basso, verrà applicata l'impostazione del contesto che lo contiene.

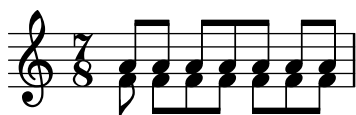
```
\new Staff {
```

```

\time 7/8
% Nessun bisogno di disabilitare beamExceptions perché non è definita per il tempo 7/8

\set Staff.beatStructure = 2,3,2
<<
  \new Voice = one {
    \relative {
      a'8 a a a a a a
    }
  }
  \new Voice = two {
    \relative {
      \voiceTwo
      \set Voice.beatStructure = 1,3,3
      f'8 f f f f f f
    }
  }
>>
}

```



Quando si usano più voci, occorre specificare il contesto Staff se si vuole applicare la disposizione delle travature a tutte le voci del rigo:

```

\time 7/8
% ritmo 3-1-1-2
% Se non si specifica il contesto, la modifica viene applicata a Voice e quindi non funziona
% Dato che le voci sono autogenerate, tutto il ritmo avrà come beatBase #1/8
\set beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>

% Funziona correttamente se si specifica il contesto Staff
\set Staff.beatStructure = 3,1,1,2
<< \relative {a'8 a a a16 a a a a8 a} \\ \relative {f'4. f8 f f f} >>

```



Il valore di beatBase può essere regolato in modo da cambiare il comportamento delle travature, se si vuole. In questo caso occorre cambiare anche il valore di beatStructure così che sia compatibile col nuovo valore di beatBase.

```

\time 5/8
% Nessun bisogno di disabilitare beamExceptions perché non è definita per il tempo 5/8
\set Timing.beatBase = #1/16
\set Timing.beatStructure = 7,3
\repeat unfold 10 { a16 }

```



Per impostazione predefinita, `beatBase` ha un valore di uno fratto il denominatore dell'indicazione di tempo. Le eccezioni a questa regola si trovano in `scm/time-signature-settings.scm`.

Disposizione delle travature con beamExceptions

Le regole speciali di disposizione automatica delle travature (diverse da quelle che determinano la corrispondenza della travatura alla suddivisione) sono definite nella proprietà `beamExceptions`.

Il valore di `beamExceptions`, una struttura dati Scheme piuttosto complessa, è più facile da generare con la funzione `\beamExceptions`. A tale funzione viene passato uno o più schemi ritmici della misura, specificati con travature manuali. Le misure devono essere separate da un controllo di battuta | dato che la funzione non ha altro modo per determinare la lunghezza della misura. Ecco un semplice esempio:

```
\relative c' {
  \time 3/16
  \set Timing.beatStructure = 2,1
  \set Timing.beamExceptions =
    \beamExceptions { 32[ 32] 32[ 32] 32[ 32] }
  c16 c c |
  \repeat unfold 6 { c32 } |
}
```



Nota: Il valore di `beamExceptions` deve essere una lista *completa* di eccezioni, ovvero bisogna includere tutte le eccezioni che si vogliono applicare. Non è possibile aggiungere, rimuovere o modificare soltanto una eccezione. Anche se questo può sembrare scomodo, significa anche che non c'è bisogno di conoscere le attuali impostazioni delle travature per poter specificare un nuovo modello di disposizione delle travature.

Quando cambia l'indicazione di tempo, vengono impostati i valori predefiniti di `Timing.beatBase`, `Timing.beatStructure` e `Timing.beamExceptions`. L'impostazione dell'indicazione di tempo ripristina le impostazioni automatiche delle travature del contesto `Timing` ai valori predefiniti.

```
\relative a' {
  \time 6/8
  \repeat unfold 6 { a8 }
  % raggruppamento (4 + 2)
  \set Timing.beatStructure = 4,2
  \repeat unfold 6 { a8 }
  % ritorno al comportamento predefinito
  \time 6/8
  \repeat unfold 6 { a8 }
}
```



Le impostazioni predefinite della disposizione automatica delle travature per ogni tempo sono definite in `scm/time-signature-settings.scm`. La loro modifica è descritta in [Indicazione di tempo], pagina 71.

Molte impostazioni di travature automatiche per le indicazioni di tempo hanno un elemento `beamExceptions`. Ad esempio, il tempo 4/4 cerca di creare due travature nella misura se ci sono solo note di un ottavo. La regola `beamExceptions` può sovrascrivere l'impostazione di `beatStructure` se `beamExceptions` non viene annullato.

```
\time 4/4
\set Timing.beatBase = #1/8
\set Timing.beatStructure = 3,3,2
% Le travature non saranno raggruppate in (3 3 2) a causa di beamExceptions
\repeat unfold 8 {c8} |
% Il raggruppamento delle travature è (3 3 2) perché abbiamo tolto le impostazioni predefinite
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c8}
```



Analogamente, le note di un ottavo in un tempo 3/4 sono raggruppate in un'unica travatura. Per raggrupparle secondo le suddivisioni, azzera `beamExceptions`.

```
\time 3/4
% il comportamento predefinito è un gruppo di (6) a causa di beamExceptions
\repeat unfold 6 {a8} |
% Le travature saranno raggruppate in (1 1 1) a causa dei valori predefiniti di beatBase
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a8}
```



Spesso, nelle partiture di età classica e romantica, le travature iniziano a metà della misura in un tempo 3/4; ma la pratica moderna preferisce evitare l'impressione ingannevole di un tempo 6/8 (vedi Gould, p. 153). Situazioni simili si incontrano anche per il tempo 3/8. Questo comportamento è controllato dalla proprietà di contesto `beamHalfMeasure`, che ha effetto soltanto sulle indicazioni di tempo che hanno 3 come numeratore:

```
\relative a' {
  \time 3/4
  r4. a8 a a |
  \set Timing.beamHalfMeasure = ##f
  r4. a8 a a |
}
```



Come funziona la disposizione automatica delle travature

Quando la disposizione automatica delle travature è abilitata, la disposizione delle travature è determinata dalle proprietà di contesto `beatBase`, `beatStructure` e `beamExceptions`.

Nel determinare l'aspetto delle travature vengono applicate le seguenti regole, in ordine di priorità:

- Se si specifica una travatura manuale con [...] imposta la travatura in quel modo, altrimenti
- se è definita una regola di fine della travatura in `beamExceptions` per il tipo di travatura in questione, la usa per determinare i punti corretti in cui le travature possono terminare, altrimenti
- se è definita una regola di fine della travatura in `beamExceptions` per un tipo di travatura più lunga, la usa per determinare i punti corretti in cui le travature possono terminare, altrimenti
- usa i valori di `beatBase` e `beatStructure` per determinare l'estensione delle suddivisioni della misura e terminare le travature alla fine delle suddivisioni.

Nelle regole precedenti, il *tipo di travatura* è la durata della nota più corta nel gruppo della travatura.

Le regole predefinite per le travature si trovano in `scm/time-signature-settings.scm`.

Frammenti di codice selezionati

Suddividere le travature

Le travature di note consecutive di un sedicesimo (o più brevi) non vengono suddivise, ovvero i tre (o più) tratti della travatura si estendono, senza spezzarsi, sugli interi gruppi di note. Questo comportamento può essere modificato in modo da suddividere le travature in sottogruppi attraverso la proprietà `subdivideBeams`. Se impostata, le travature che comprendono più sottogruppi verranno suddivise a intervalli definiti dal valore attuale di `beatBase`, riducendo le travature multiple al numero di travature che indica il valore metrico della suddivisione. Se il gruppo successivo alla suddivisione è più breve del valore metrico corrente (di solito perché la travatura è incompleta), il numero di travature riflette il gruppo di suddivisione più lungo possibile. Tuttavia, se rimane una sola nota dopo la divisione, questa restrizione non viene applicata. Si noti che `beatBase`, se non impostata esplicitamente, equivale a uno fratto il denominatore dell'attuale indicazione di tempo. Deve quindi essere impostata su una frazione che stabilisca la durata del sottogruppo di travature; lo si può fare usando la funzione `ly:make-moment`, come è mostrato in questo frammento di codice. Inoltre quando `beatBase` cambia, anche `beatStructure` deve essere modificato per accordarsi con `beatBase`:

```
\relative c' ' {
  \time 1/4

  <>^"default"
  c32 c c c c c c c

  <>^"with subdivision"
  \set subdivideBeams = ##t
  c32 c c c c c c c

  <>^"min 1/8"
  \once \set beamMinimumSubdivision = #1/8
  c32 c c c c c c c
```

```

<>~"max 1/16"
\once \set beamMaximumSubdivision = #1/16
c32 c c c c c c c

<>~"max 3/8"
\once \set beamMaximumSubdivision = #3/8
\repeat unfold 16 c64

<>~"min 1/32, max 1/64"
% Set maximum beam subdivision interval to 1/64 to limit
% subdivision depth, despite not being metrically correct.
\once \set beamMinimumSubdivision = #1/32
\once \set beamMaximumSubdivision = #1/64
\repeat unfold 32 c128
\break

<>~"beams with incomplete subdivisions"
c32 c c c c c c r32
c32 c c c c c r16.

<>~\markup { "the same with"
      \typewriter { "respectIncomplete=#t" } }
\set respectIncompleteBeams = ##t
% The incomplete subgroup extends the completed subgroup.
c32 c c c c c c r32
% No visual change since we have only two stems in the
% incomplete subgroup.
c32 c c c c c r16.
}

```

The image displays six musical staves in 4/4 time, each illustrating a different beam subdivision setting. The staves are numbered 1 through 6 on the left margin.

- Staff 1:** Labeled "default", "with subdivision", and "min 1/8". It shows a sequence of eighth notes beamed in groups of four.
- Staff 2:** Labeled "max 1/16" and "max 3/8". It shows a sequence of sixteenth notes beamed in groups of four, followed by a sequence of eighth notes beamed in groups of four.
- Staff 3:** Labeled "min 1/32, max 1/64". It shows a sequence of thirty-two sixteenth notes beamed in groups of four.
- Staff 4:** Labeled "beams with incomplete subdivisions". It shows a sequence of eighth notes beamed in groups of four, followed by a sequence of eighth notes beamed in groups of four, with the final note of the second group being a dotted quarter note.
- Staff 5:** Labeled "the same with respectIncomplete=#t". It shows a sequence of eighth notes beamed in groups of four, followed by a sequence of eighth notes beamed in groups of four, with the final note of the second group being a dotted quarter note.

Travatura che segue strettamente il battito

Si possono impostare i tratti di suddivisione della travatura in modo che siano rivolti verso la relativa pulsazione. La prima travatura fa sì che non spuntino i tratti di suddivisione (comportamento predefinito); la seconda travatura è orientata verso la pulsazione.

```
\relative c'' {
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



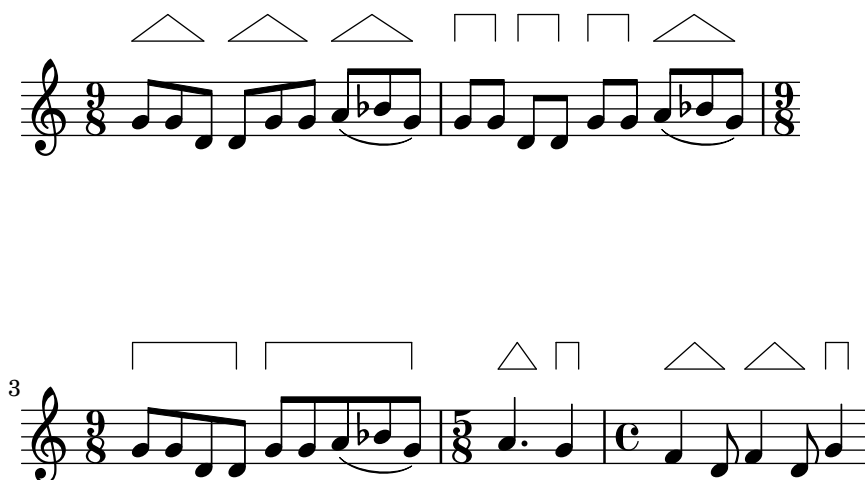
Segni per la conduzione, segni di raggruppamento della misura

Il raggruppamento delle pulsazioni all'interno della misura è regolato dalla proprietà di contesto `beatStructure`. I valori di `beatStructure` per varie indicazioni di tempo vengono stabiliti in `scm/time-signature-settings.scm`. Questi valori possono essere impostati o modificati con `\set`. Altrimenti, si può usare `\time` per impostare sia l'indicazione di tempo che la struttura delle pulsazioni. Per farlo si specifica il raggruppamento interno delle pulsazioni in una misura in una lista di numeri (nella sintassi di Scheme) prima dell'indicazione di tempo.

`\time` agisce nel contesto `Timing`, dunque non reimposterà i i valori di `beatStructure` e `beatBase` che sono impostati in altri contesti di più basso livello, come `Voice`.

Se si include l'incisore `Measure_grouping_engraver` in uno dei contesti che regolano l'aspetto, appariranno i segni di raggruppamento della misura. Tali segni facilitano la lettura di musica moderna ritmicamente complessa. Nell'esempio la misura di 9/8 è raggruppata in due diversi schemi usando due metodi differenti, mentre la misura di 5/8 è raggruppata in base alle impostazioni predefinite in `scm/time-signature-settings.scm`:

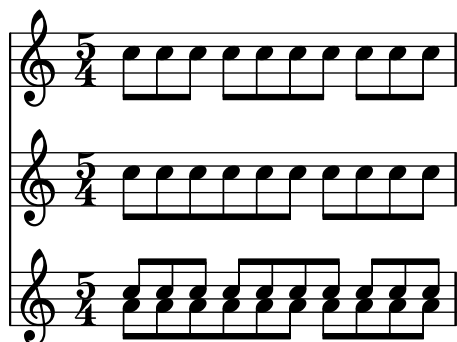
```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
    \time 3,3,2 4/4
    \set Timing.beatBase = #1/8
    f4 d8 f4 d8 g4
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Estremità delle travature nel contesto Score

Le regole relative alle estremità delle travature definite nel contesto Score si applicano a tutti i rigi, ma possono essere modificate anche ai livelli Staff e Voice:

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.beatBase = #1/8
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
```

Vedi anche

Guida alla notazione: [Indicazione di tempo], pagina 71.

File installati: scm/time-signature-settings.scm.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

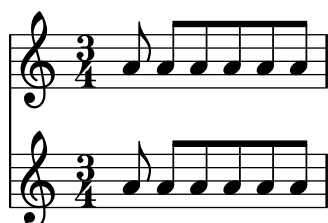
Guida al funzionamento interno: Sezione “Auto_beam_engraver” in *Guida al Funzionamento Interno*, Sezione “Beam” in *Guida al Funzionamento Interno*, Sezione “BeamForbidEvent” in *Guida al Funzionamento Interno*, Sezione “beam-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Se una partitura finisce prima del termine di una travatura automatica, cui mancano ancora delle note, quest’ultima travatura non apparirà. Lo stesso vale per le voci polifoniche, inserite con << ... \ \ ... >>. Una voce polifonica non apparirà se termina quando una travatura automatica è ancora in attesa di note. Per aggirare questi problemi occorre impostare manualmente l’ultima travatura della voce o della partitura.

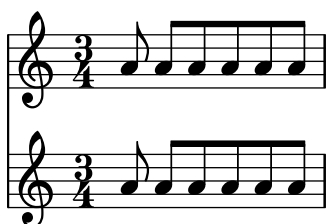
Timing è un alias del contesto Score. Questo significa che la modifica della disposizione delle travature in un rigo avrà effetto anche sugli altri rigi. Quindi un’impostazione di tempo in un rigo successivo reimposterà la disposizione personalizzata delle travature definita in un rigo precedente. Per evitare questo problema si può impostare l’indicazione di tempo su un solo rigo.

```
<<
  \new Staff {
    \time 3/4
    \set Timing.beatBase = #1/8
    \set Timing.beatStructure = 1,5
    \set Timing.beamExceptions = #'()
    \repeat unfold 6 { a'8 }
  }
  \new Staff {
    \repeat unfold 6 { a'8 }
  }
>>
```



Si possono cambiare anche le impostazioni predefinite delle travature, in modo che sia usata sempre la disposizione delle travature desiderata. Le modifiche nelle impostazioni della travatura automatica per le indicazioni di tempo sono descritte in [Indicazione di tempo], pagina 71.

```
<<
\new Staff {
  \overrideTimeSignatureSettings
    3/4          % timeSignature
    #1/8         % beatBase
    1,5         % beatStructure
    #'() % beamExceptions
  \time 3/4
  \repeat unfold 6 { a'8 }
}
\new Staff {
  \time 3/4
  \repeat unfold 6 { a'8 }
}
>>
```



Travature manuali

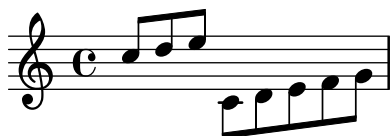
In alcuni casi potrebbe essere necessario scavalcare l'algoritmo di disposizione automatica delle travature. Ad esempio, questo algoritmo non inserirà delle travature tra le pause o tra le stanghet-
te; e nelle partiture corali la disposizione delle travature è spesso determinato dall'articolazione del testo piuttosto che da quella musicale. Tali travature possono essere specificate manualmente indicandone l'inizio e la fine con [e] .

```
\relative { r4 r8[ g' a r] r g[ | a] r }
```



La direzione delle travature può essere impostata manualmente attraverso gli indicatori di direzione:

```
\relative { c''8^[ d e] c,_[ d e f g] }
```



Le note individuali possono essere contrassegnate con \noBeam per impedire che vengano inserite in una travatura:

```
\relative {
  \time 2/4
  c''8 c\noBeam c c
```

}



Le travature degli abbellimenti e quelle delle note normali possono coesistere simultaneamente. Gli abbellimenti privi di travatura non vengono inseriti nella travatura delle note normali.

```
\relative {
  c' '4 d8[
    \grace { e32 d c d }
  e8] e[ e
    \grace { f16 }
  e8 e]
}
```



Si può ottenere un controllo manuale delle travature ancora più preciso agendo sulle proprietà `stemLeftBeamCount` e `stemRightBeamCount`, che specificano il numero di travature da creare a sinistra e a destra della nota successiva. Se una di queste proprietà viene impostata, il suo valore verrà usato una volta sola, e la proprietà sarà poi cancellata. In questo esempio, l'ultima nota `f` ha una sola travatura a sinistra: la travatura corrispondente alla sottodivisione di un ottavo all'interno dell'intero raggruppamento.

```
\relative a' {
  a8[ r16 f g a]
  a8[ r16
    \set stemLeftBeamCount = #2
    \set stemRightBeamCount = #1
  f16
    \set stemLeftBeamCount = #1
  g16 a]
}
```



Comandi predefiniti

`\noBeam.`

Frammenti di codice selezionati

Beam nibs

Beam nibs at the start and end of beams together with beams attached to solitary notes that look like flat flags are possible with a combination of `stemLeftBeamCount`, `stemRightBeamCount`, and paired `[]` beam indicators.

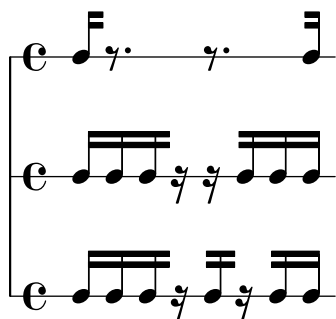
For imitating right-pointing flat flags on lone notes, use paired `[]` beam indicators and set `stemLeftBeamCount` to zero. For imitating left-pointing flat flags on lone notes, set `stemRightBeamCount` to zero instead (line one).

For right-pointing nibs at the end of a run of beamed notes, set `stemRightBeamCount` to a positive value. For left-pointing nibs at the start of a run of beamed notes, set `stemLeftBeamCount` instead (line two).

Sometimes it may make sense for a lone note surrounded by rests to carry both a left- and right-pointing nib. Do this with paired `[]` beam indicators alone (line three).

Note that `\set stemLeftBeamCount` is always equivalent to `\once \set`. In other words, the beam count settings are not “sticky”, so the pair of nibs attached to the lone 16th note in the last example has nothing to do with the `\set` command for the beam before.

```
\score {
  <<
    \new RhythmicStaff {
      \set stemLeftBeamCount = 0
      c16[] r8.
      r8.
      \set stemRightBeamCount = 0
      16[]
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r r
      \set stemLeftBeamCount = 2
      16 16 16
    }
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = 2
      16 r16
      16[] r16
      \set stemLeftBeamCount = 2
      16 16
    }
  >>
}
```



Vedi anche

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Abbellimenti], pagina 122.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Beam” in *Guida al Funzionamento Interno*, Sezione “BeamEvent” in *Guida al Funzionamento Interno*, Sezione “Beam_engraver” in *Guida al*

Funzionamento Interno, Sezione “beam-interface” in *Guida al Funzionamento Interno*, Sezione “Stem_engraver” in *Guida al Funzionamento Interno*.

Travature a raggiera

Le travature a raggiera servono a indicare che un gruppo di note determinato deve essere eseguito a un tempo progressivamente accelerato (o rallentato), senza cambiare l’andamento complessivo del brano. L’estensione della travatura a raggiera deve essere indicato a mano con [e], e la convergenza o divergenza delle travature si determina specificando la la direzione della proprietà Beam di grow-direction.

Perché il *ritardando* o l’*accelerando* indicati dalla travatura a raggiera trovino riscontro nella disposizione delle note e nell’esecuzione del file MIDI, le note devono essere raggruppate in un’espressione musicale delimitata da parentesi graffe e preceduta dal comando `\featherDurations`, che specifica il rapporto tra le durate delle prime e delle ultime note del gruppo.

Le parentesi quadre indicano l’estensione della travatura, mentre quelle graffe indicano quali note devono avere una durata modificata. Di norma queste parentesi delimitano lo stesso gruppo di note, ma questo non è tassativo: i due comandi sono indipendenti.

Nell’esempio seguente le otto note da un sedicesimo occupano esattamente lo stesso tempo di una nota di due quarti, ma la prima nota dura la metà dell’ultima e le note intermedie si allungano gradualmente. Le prime quattro note da un trentaduesimo sono progressivamente più veloci, mentre le ultime quattro presentano lo stesso tempo.

```
\relative c' {
  \override Beam.grow-direction = #LEFT
  \featherDurations 2/1
  { c16[ c c c c c c c ] }
  \override Beam.grow-direction = #RIGHT
  \featherDurations 2/3
  { c32[ d e f ] }
  % ripristina le travature normali
  \override Beam.grow-direction = #'()
  { g32[ a b c ] }
}
```



La spaziatura rappresenta la durata effettiva delle note solo in modo approssimato, mentre il tempo nel file MIDI è esatto.

Comandi predefiniti

`\featherDurations`.

Vedi anche

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Problemi noti e avvertimenti

Il comando `\featherDurations` funziona solamente con frammenti di musica molto brevi e quando i numeri della frazione sono piccoli.

1.2.5 Battute

Stanghette

Le stanghette delimitano le misure e sono usate anche per indicare i ritornelli. Di norma, le stanghette semplici sono inserite automaticamente in base all'indicazione di tempo.

Si possono inserire altri tipi di stanghette col comando `\bar`. Ad esempio, di solito si usa una stanghetta finale al termine di un brano:

```
\relative { e'4 d c2 \bar "|" }
```



Se l'ultima nota di una misura non termina entro la stanghetta inserita automaticamente, non viene segnalato un errore: si presuppone che la nota continui nella misura successiva. Ma se ci sono tante misure simili in sequenza, la musica potrebbe apparire compressa oppure scorrere fuori dalla pagina. Questo accade perché le interruzioni di linea automatiche si verificano solo al termine di misure complete, ovvero quando tutte le note terminano prima dell'inizio di una misura.

Nota: Una durata errata può impedire un'interruzione di linea, causando una linea di musica altamente compressa oppure a musica che prosegue fuori dalla pagina.

Le interruzioni di linea sono permesse anche in caso si stanghette inserite a mano anche all'interno di misure incomplete. Per permettere un'interruzione di linea senza che appaia una stanghetta si usa:

```
\bar ""
```

Questo comando inserirà una stanghetta invisibile e consentirà (senza però forzarla) un'interruzione di linea in questo punto. Il conteggio dei numeri di battuta non incrementa. Per forzare un'interruzione di linea si veda Sezione 4.3.1 [Interruzioni di linea], pagina 567.

Si possono inserire questa e altre stanghette speciali in qualsiasi punto. Quando coincidono con la fine di una misura, sostituiscono la stanghetta semplice che sarebbe stata posta automaticamente. Quando non coincidono con la fine di una misura, la stanghetta specificata viene inserita in quel punto.

Si noti che le stanghette manuali hanno una funzione puramente visiva. Non hanno alcun effetto sulle proprietà di una normale stanghetta, come i numeri della misura, le alterazioni, le interruzioni di linea, etc. Non influiscono nemmeno sul conteggio e sulla posizione delle stanghette automatiche successive. Quando una stanghetta manuale è posta nel punto in cui si trova già una normale stanghetta, le caratteristiche della stanghetta originale non sono alterate.

Sono disponibili per l'inserimento manuale due tipi di stanghette semplici e cinque tipi di stanghette doppie:

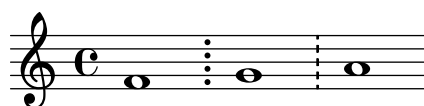
```
\relative {
  f'1 \bar "|"
  f1 \bar "."
  g1 \bar "||"
  a1 \bar ".|"
  b1 \bar ".."
  c1 \bar "|.|"
  d1 \bar "|."
```

```
e1
}
```



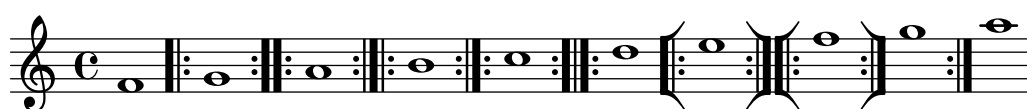
oltre alle stanghette puntate e tratteggiate:

```
\relative {
  f'1 \bar ";"
  g1 \bar "!"
  a1
}
```



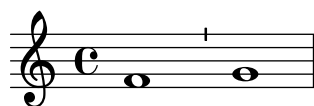
e a nove tipi di stanghette per le ripetizioni:

```
\relative {
  f'1 \bar ".|:"
  g1 \bar ":\.:"
  a1 \bar ":\.|:"
  b1 \bar ":\.|"
  c1 \bar ":\.|\.:"
  d1 \bar "[|:"
  e1 \bar ":\]|[:]"
  f1 \bar ":\]|]"
  g1 \bar ":\.|"
  a1
}
```



Inoltre, una stanghetta può apparire come un semplice segno di spunta:

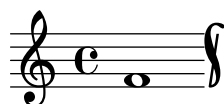
```
f'1 \bar "" g'1
```



Tuttavia, dato che questi segni di spunta sono tipicamente usati nella notazione gregoriana, è preferibile usare `\divisioMinima`, come è descritto nella sezione [Divisiones], pagina 467, della parte dedicata al canto gregoriano.

LilyPond supporta la notazione gregoriana russa e fornisce una stanghetta speciale per questo tipo di notazione:

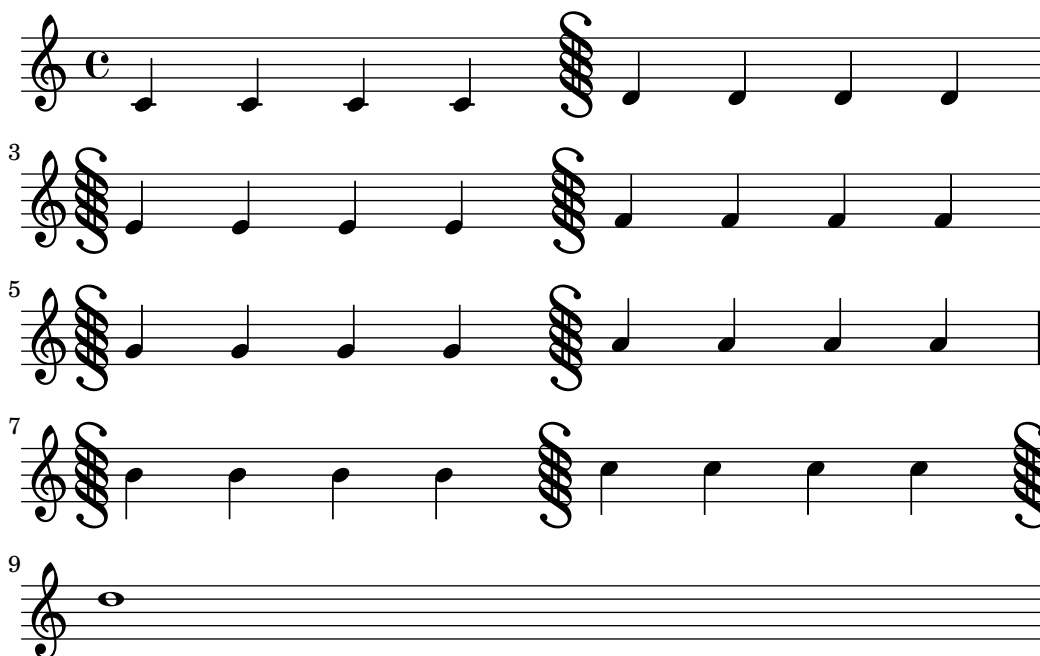
```
f'1 \bar "k"
```



I dettagli di questo tipo di notazione sono spiegati in Sezione 2.9.5 [Scrivere in notazione quadrata di Kiev], pagina 475.

Per i segni di tipo segno interni al rigo, ci sono tre tipi di stanghette che differiscono nel comportamento quando incontrano un'interruzione di linea:

```
\fixed c' {
  c4 4 4 4
  \bar "S"
  d4 4 4 4 \break
  \bar "S"
  e4 4 4 4
  \bar "S-|"
  f4 4 4 4 \break
  \bar "S-|"
  g4 4 4 4
  \bar "S-||"
  a4 4 4 4 \break
  \bar "S-||"
  b4 4 4 4
  \bar "S-S"
  c'4 4 4 4 \break
  \bar "S-S"
  d'1
}
```



Sebbene LilyPond preveda l'inserimento manuale delle stanghette che indicano i ritornelli, ciò non consente il riconoscimento della musica come una sezione da ripetere. Tali sezioni devono essere inserite con i vari comandi di ripetizione (vedi Sezione 1.4 [Ripetizioni], pagina 160), che creano automaticamente le stanghette appropriate.

Inoltre si può specificare ".|:-||", che è equivalente a ".|:" tranne in presenza di un'interruzione di linea, dove crea una doppia stanghetta alla fine della linea e una stanghetta di inizio ripetizione all'inizio della linea successiva.

```
\fixed c' {
```



```

c4 4 4 4
\bar ".|:"
d4 4 4 4 \break
\bar ".|:"
e4 4 4 4
\bar ".|:-|"
f4 4 4 4 \break
\bar ".|:-|"
g4 4 4 4
\bar ".|:-||"
a4 4 4 4 \break
\bar ".|:-||"
b4 4 4 4
\bar ".|:-|. "
c'4 4 4 4 \break
\bar ".|:-|. "
d'4 4 4 4
}

```



Esistono sei diverse combinazioni di ripetizioni e indicazioni di segno:

```

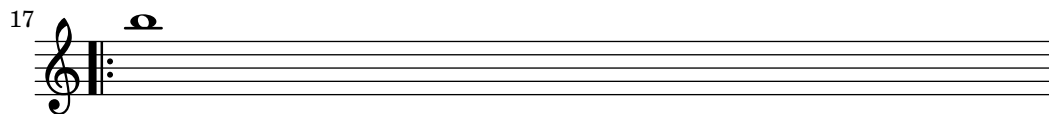
\fixed c' {
  g,4 4 4 4
  \bar ":|.S"
  a,4 4 4 4 \break
  \bar ":|.S"
  b,4 4 4 4
  \bar ":|.S-S"
  c4 4 4 4 \break
  \bar ":|.S-S"
  d4 4 4 4
  \bar "S.|:-S"
  e4 4 4 4 \break
  \bar "S.|:-S"
}

```

```

f4 4 4 4
\bar "S. | : "
g4 4 4 4 \break
\bar "S. | : "
a4 4 4 4
\bar "S. | : - | "
b4 4 4 4 \break
\bar "S. | : - | "
c'4 4 4 4
\bar "S. | : - | | "
d'4 4 4 4 \break
\bar "S. | : - | | "
e'4 4 4 4
\bar " : | .S. | : "
f'4 4 4 4 \break
\bar " : | .S. | : "
g'4 4 4 4
\bar " : | .S. | : -S"
a'4 4 4 4 \break
\bar " : | .S. | : -S"
b'1
}

```



Esiste inoltre un comando `\inStaffSegno` che crea una stanghetta con segno in congiunzione con un'appropriata stanghetta di ripetizione se usata con un comando `\repeat volta`, vedi [Ripetizioni normali], pagina 161..

Si possono definire nuovi tipi di stanghette con `\defineBarLine`:

```
\defineBarLine tipo-stanghetta #'(fine inizio span)
```

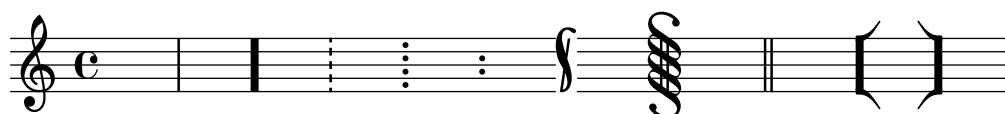
Le variabili di `\defineBarLine` possono includere la stringa 'vuota' "", che è equivalente a una stanghetta invisibile. Oppure possono essere impostate su #f, che fa sì che non appaia alcuna stanghetta.

Dopo averla definita, si può richiamare la nuova stanghetta col comando `\bar tipo-stanghetta`.

Attualmente sono disponibile dieci tipi di stanghetta:

```
\defineBarLine ":" #'(" " ":" " ")
\defineBarLine "=" #'(" " "=" " " " ")
\defineBarLine "[" #'(" " "[" " " " ")
\defineBarLine "]" #'(" " "]" " " " ")
```

```
\new Staff {
  s1 \bar "|"
  s1 \bar "."
  s1 \bar "!"
  s1 \bar ";"
  s1 \bar ":"
  s1 \bar "k"
  s1 \bar "S"
  s1 \bar "="
  s1 \bar "["
  s1 \bar "]"
  s1 \bar ""
}
```



La stanghetta "=" crea una stanghetta doppia da combinare con il segno. Non va usata per creare una stanghetta doppia indipendente; in questo caso è preferibile usare `\bar "||"`.

Il segno "-" introduce le annotazioni alle stanghette che servono a distinguere quelle che hanno aspetto identico ma un diverso comportamento in corrispondenza delle interruzioni di linea e/o un diverso modo di connettere le stanghette tra i rigi. La parte che segue il segno "-" non viene usato per costruire la stanghetta.

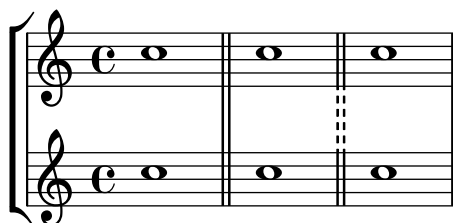
```
\defineBarLine "||-dashedSpan" #'("||" " " "!!")
```

```
\new StaffGroup <<
  \new Staff \relative c'' {
    c1 \bar "||"
    c1 \bar "||-dashedSpan"
    c1
  }
}
```

```

\new Staff \relative c'' {
  c1
  c1
  c1
}
>>

```



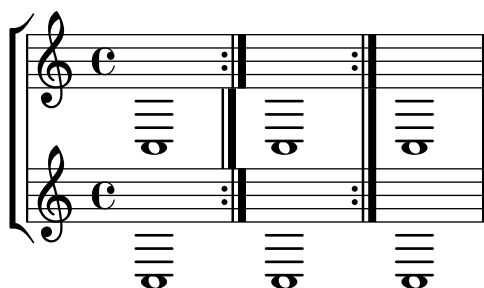
Inoltre, lo spazio " " fa da spaziatore e fa sì che le stanghette tra i righi siano allineate correttamente alle stanghette principali:

```

\defineBarLine ":|.-sbagliata" #'(" :|. " " " |.")
\defineBarLine ":|.-giusta" #'(" :|. " " " |.")

\new StaffGroup <<
  \new Staff {
    c1 \bar ":|.-sbagliata"
    c1 \bar ":|.-giusta"
    c1
  }
  \new Staff {
    c1
    c1
    c1
  }
}
>>

```



Se servono ulteriori elementi, LilyPond fornisce un modo semplice per definirli. Maggiori informazioni sulla modifica e l'aggiunta delle stanghette sono presenti nel file `scm/bar-line.scm`.

Nelle partiture con molti rigi, un comando `\bar` inserito in un rigo viene applicato automaticamente a tutti i rigi. Le stanghette risultanti sono connesse tra i diversi rigi di un `StaffGroup`, `PianoStaff` o `GrandStaff`.

```

<<
\new StaffGroup <<
  \new Staff \relative {
    e'4 d
    \bar "||"
    f4 e
  }
}
>>

```

```

    }
    \new Staff \relative { \clef bass c'4 g e g }
  >>
  \new Staff \relative { \clef bass c'2 c2 }
  >>

```



Il comando ‘`\bar tipo-stanghetta`’ è una scorciatoia di ‘`\set Timing.whichBar = tipo-stanghetta`’. Una stanghetta viene creata ogni volta che si imposta la proprietà `whichBar`.

Il tipo di stanghetta predefinita per le stanghettoni inserite automaticamente è “|”. Si può modificare in qualsiasi momento con ‘`\set Timing.measureBarType = tipo-stanghetta`’.

Vedi anche

Guida alla notazione: Sezione 4.3.1 [Interruzioni di linea], pagina 567, Sezione 1.4 [Ripetizioni], pagina 160, [Raggruppare i rigli], pagina 203.

File installati: `scm/bar-line.scm`.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “BarLine” in *Guida al Funzionamento Interno* (creata al livello Staff), Sezione “SpanBar” in *Guida al Funzionamento Interno* (tra i rigli), Sezione “Timing_translator” in *Guida al Funzionamento Interno* (per le proprietà di Timing).

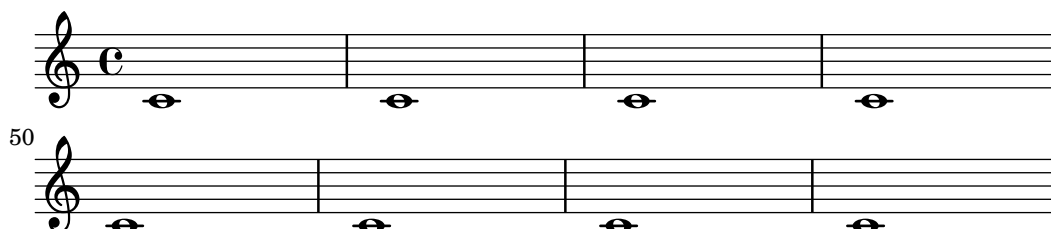
Numeri di battuta

I numeri di battuta appaiono all’inizio di ogni linea tranne la prima. Il numero viene salvato nella proprietà `currentBarNumber`, che viene aggiornata automaticamente per ogni misura. Può anche essere impostata a mano:

```

\relative c' {
  c1 c c c
  \break
  \set Score.currentBarNumber = #50
  c1 c c c
}

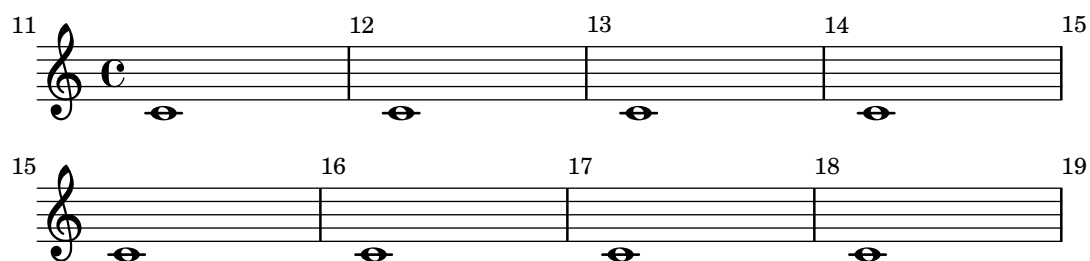
```



I numeri di battuta possono essere mostrati a intervalli regolari anziché solo all’inizio di ogni linea. Per farlo occorre sovrascrivere il comportamento predefinito e permettere ai numeri

di battuta di apparire anche in punti diversi dall'inizio della linea. Questo comportamento è regolato dalla proprietà `break-visibility` di `BarNumber`, che considera tre valori impostabili su `#t` o `#f`, i quali indicano se il numero di battuta corrispondente debba essere visibile o no. L'ordine dei tre valori è `end of line visible`, `middle of line visible`, `beginning of line visible`. Nell'esempio seguente i numeri di battuta compaiono in tutti i punti possibili:

```
\relative c' {
  \override Score.BarNumber.break-visibility = ##(#t #t #t)
  \set Score.currentBarNumber = #11
  % Permette la visualizzazione del primo numero di battuta
  \bar ""
  c1 | c | c | c
  \break
  c1 | c | c | c
}
```



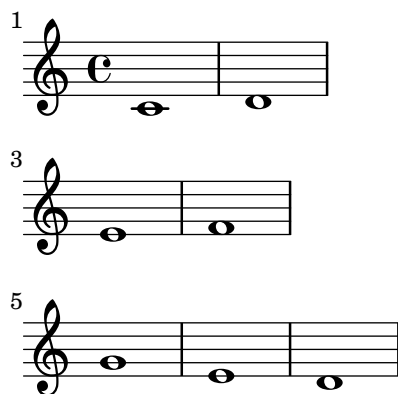
Frammenti di codice selezionati

Mostrare il numero di battuta nella prima misura

Il primo numero di battuta di una partitura viene soppresso se è inferiore o uguale a '1'. Se si imposta `barNumberVisibility` su `all-bar-numbers-visible`, verrà mostrato il numero di battuta della prima misura e di tutte quelle successive. Si noti che perché funzioni è necessario inserire una stanghetta invisibile prima della prima nota.

```
\paper {
  line-width = 50\mm
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  c1 | d | e | f \break
  g1 | e | d | c
}
```

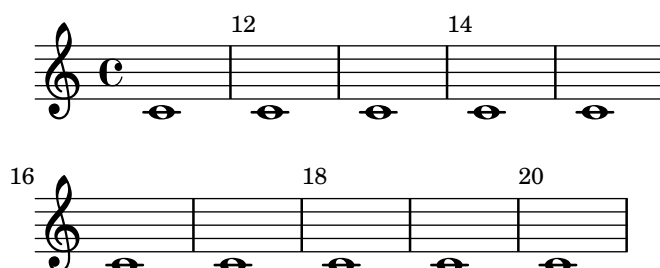




Mostrare i numeri di battuta a intervalli regolari

Impostando la proprietà `barNumberVisibility` si possono far apparire i numeri di battuta a intervalli regolari. In questo esempio vengono mostrati ogni due misure eccetto alla fine della linea.

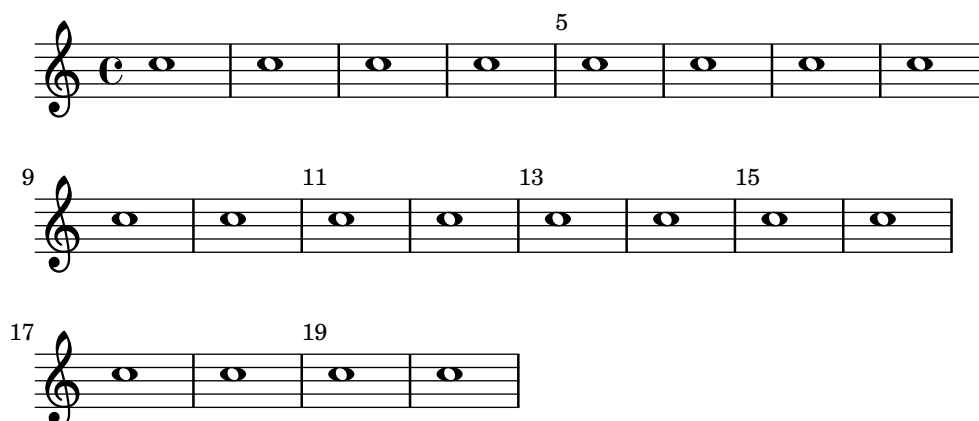
```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = 11
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



Stampare i numeri di battuta a intervalli regolari variabili

Tramite la funzione di contesto `{set-bar-number-visibility}` si possono modificare gli intervalli dei numeri di battuta.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```



Numeri di battuta racchiusi in rettangoli o cerchi

I numeri di battuta possono apparire anche all'interno di rettangoli o cerchi.

```
\relative c' {
  % Center bar numbers except at the beginning of a staff.
  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER CENTER 0.3)

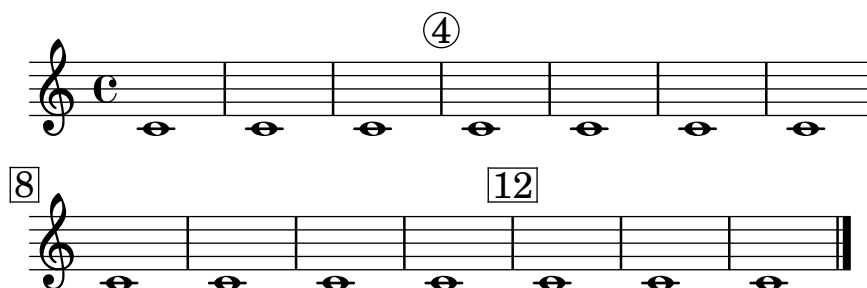
  % Prevent bar numbers at the end of a line and permit them elsewhere.
  \override Score.BarNumber.break-visibility = #end-of-line-invisible

  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2.
  \override Score.BarNumber.font-size = 2

  % Draw a circle round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 7 { c1 } \break

  % Draw a box round the following bar number(s).
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 7 { c1 } \bar "|"
}
```



Numeri di battuta alternativi

Si possono impostare due metodi alternativi di numerazione della battuta, utili specialmente per le ripetizioni.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "|"
}

{
  \textMark \markup \large "default"
  \music
}
```



```

}

{
  \textMark \markup \large \typewriter "numbers"
  \set Score.alternativeNumberingStyle = #'numbers
  \music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}

```

default

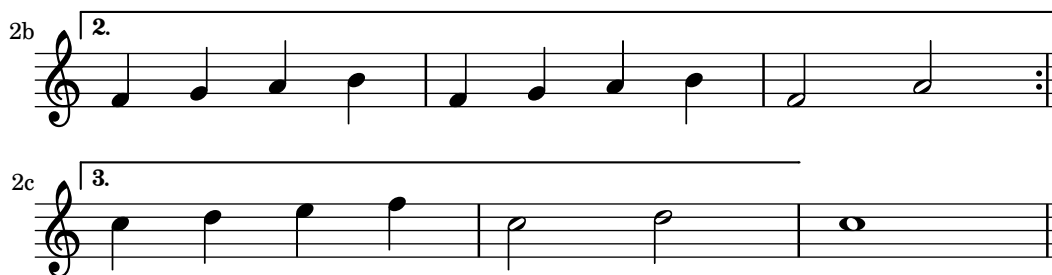


numbers



numbers-with-letters





Allineare i numeri di battuta

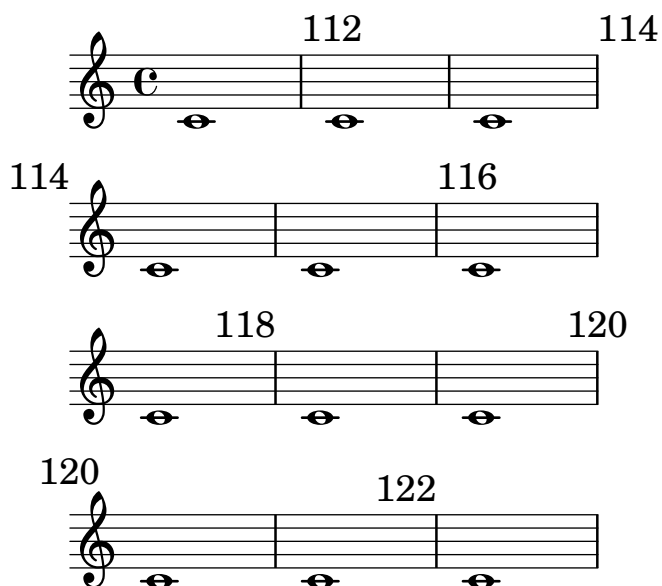
Per impostazione predefinita i numeri di battuta sono allineati a destra rispetto al loro oggetto genitore. Di solito si tratta del margine sinistro della linea oppure, se i numeri appaiono all'interno della linea, del lato sinistro della stanghetta. I numeri possono essere posizionati anche direttamente sopra la stanghetta oppure allineati a sinistra della stanghetta.

```
\relative c' {
  \set Score.currentBarNumber = 111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = 2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)

  c1 | c1 | c1 | \break
  c1 | c1 | c1 | \break

  \override Score.BarNumber.self-alignment-X =
    #(break-alignment-list CENTER RIGHT CENTER)
  c1 | c1 | c1 | \break
  c1 | c1 | c1 |
}

\paper {
  line-width = 70\mm
}
```



Togliere i numeri di battuta da uno spartito

I numeri di battuta possono essere tolti rimuovendo l'incisore `Bar_number_engraver` dal contesto `Score`.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    % \remove "Bar_number_engraver"
  }
}

\relative c' {
  c4 c c c \break
  c4 c c c
}
```



Vedi anche

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “BarNumber” in *Guida al Funzionamento Interno*, Sezione “Bar_number_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

I numeri di battuta possono collidere con la parte superiore della parentesi quadra di `StaffGroup`, se presente. Per evitare la collisione, si può usare la proprietà `padding` di `BarNumber` per posizionare correttamente il numero. Si veda Sezione “StaffGroup” in *Guida al Funzionamento Interno* e Sezione “BarNumber” in *Guida al Funzionamento Interno* per maggiori informazioni.

Controlli di battuta e del numero di battuta

I controlli di battuta aiutano a rilevare gli errori di durata. Il controllo di battuta si inserisce col simbolo della barra verticale, `|`, in un qualsiasi punto in cui è previsto l'inserimento di una stanghetta. Se vengono trovati controlli di battuta in punti diversi, viene creata una lista di avvisi nel file di log che mostra i numeri di linea e le linee in cui il controllo è fallito. Nell'esempio seguente il secondo controllo di battuta segnerà un errore.

```
\time 3/4 c2 e4 | g2 |
```

Una durata non corretta può generare uno spartito completamente alterato, specialmente nel caso di brani polifonici. Quindi il primo passo da compiere per correggere l'input è la verifica dei controlli di battuta e delle durate errate.

Se i controlli di battuta successivi sono spostati dello stesso intervallo musicale, viene mostrato solo il primo messaggio di avviso. Così l'avvertimento si concentra sulla causa dell'errore di tempo.

I controlli di battuta possono essere usati anche all'interno del testo vocale:

```
\lyricmode {
```

```
\time 2/4
Twin -- kle | Twin -- kle |
}
```

Attenzione: i segni di controllo di ottava nel testo vocale sono elaborati nel momento musicale in cui la sillaba *che segue* il segno di controllo viene elaborata. Se il testo è associato alle note di una voce che ha una pausa all’inizio di una battuta, non è possibile individuare alcuna sillaba all’inizio di quella battuta e apparirà un avvertimento se viene posto un controllo di battuta in quel punto del testo vocale.

È anche possibile ridefinire l’azione da prendere quando si incontra un controllo di battuta o simbolo di barra verticale, |, nell’input, in modo che avvenga qualcosa di diverso dal controllo di battuta. Si può fare assegnando un’espressione musicale a "|". Nell’esempio seguente |, invece di controllare la fine di una battuta, viene usato per inserire una stanghetta doppia ovunque appaia nell’input.

```
"|" = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
}
```



Quando si copiano brani di una certa ampiezza, può essere d’aiuto verificare che i numeri di battuta di LilyPond corrispondano all’originale a partire dal quale si sta scrivendo il brano. Si può abilitare con `\barNumberCheck`, ad esempio,

```
\barNumberCheck #123
```

genererà un avvertimento se `currentBarNumber` non è 123 nel momento in cui viene elaborato.

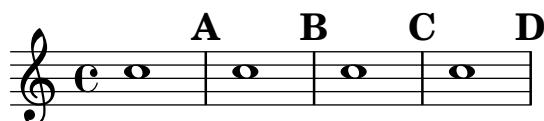
Vedi anche

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Segni di chiamata

Per creare un segno di chiamata si usa il comando `\mark`.

```
\relative c' ' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
}
```



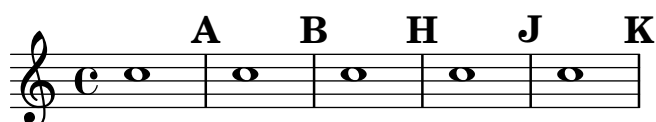
Il segno viene incrementato automaticamente se si usa `\mark \default`, ma è possibile usare anche un numero intero come argomento in modo da impostare il segno manualmente. Il valore da usare viene salvato nella proprietà `rehearsalMark`.

```
\relative c' ' {
```

```

c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
}

```

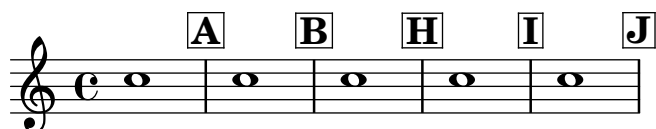


La lettera 'I' viene saltata, come vuole la tradizione tipografica. Se si desidera includere la lettera 'I', si può usare uno dei seguenti comandi, a seconda dello stile che si vuole (solo lettere, lettere in un quadrato o lettere in un cerchio).

```

\set Score.rehearsalMarkFormatter = #format-mark-alphabet
\set Score.rehearsalMarkFormatter = #format-mark-box-alphabet
\set Score.rehearsalMarkFormatter = #format-mark-circle-alphabet
\relative c'' {
  \set Score.rehearsalMarkFormatter = #format-mark-box-alphabet
  c1 \mark \default
  c1 \mark \default
  c1 \mark #8
  c1 \mark \default
  c1 \mark \default
}

```

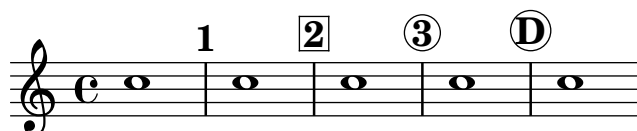


Lo stile viene definito dalla proprietà `rehearsalMarkFormatter`. È una funzione che accoglie come argomenti il segno corrente (un numero intero) e il contesto corrente. Dovrebbe restituire un oggetto testuale. Nell'esempio seguente, `rehearsalMarkFormatter` viene prima impostato su una procedura predefinita e dopo alcune misure su una procedura che produce un numero racchiuso in un quadrato.

```

\relative c'' {
  \set Score.rehearsalMarkFormatter = #format-mark-numbers
  c1 \mark \default
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-box-numbers
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-circle-numbers
  c1 \mark \default
  \set Score.rehearsalMarkFormatter = #format-mark-circle-letters
  c1
}

```



Il file `scm/translation-functions.scm` contiene le definizioni di `format-mark-letters` (il formato predefinito), `format-mark-box-letters`, `format-mark-numbers` e `format-mark-box-numbers`. Possono essere usate come fonte di ispirazione per creare altre funzioni di formattazione.

Si possono usare `format-mark-barnumbers`, `format-mark-box-barnumbers` e `format-mark-circle-barnumbers` per ottenere i numeri di battuta invece di numeri o lettere crescenti.

Si possono specificare manualmente altri stili di segni di chiamata:

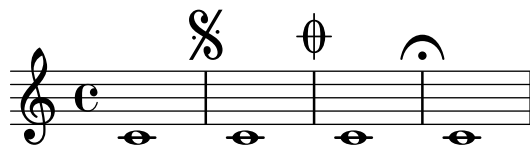
```
\mark "A1"
```

Si noti che `Score.rehearsalMarkFormatter` non ha effetto sui segni specificati in questo modo. Tuttavia, è possibile applicare un `\markup` alla stringa.

```
\mark \markup { \box A1 }
```

I glifi musicali (come il Segno) possono essere posti dentro il comando `\mark`

```
\relative c' {
  c1 \mark \markup { \musicglyph "scripts.segno" }
  c1 \mark \markup { \musicglyph "scripts.coda" }
  c1 \mark \markup { \musicglyph "scripts.ufermata" }
  c1
}
```



L'elenco dei simboli che possono essere prodotti con `\musicglyph` si trova in Sezione A.8 [Il font Emmentaler], pagina 693.

Per le più comuni modifiche relative al posizionamento dei segni di chiamata, si veda Sezione 1.8.2 [Formattazione del testo], pagina 260. Per ottenere un controllo più preciso si consiglia di studiare il funzionamento della proprietà `break-alignable-interface` descritta in Sezione 5.5.1 [Allineamento degli oggetti], pagina 660.

Il file `scm/translation-functions.scm` contiene le definizioni di `format-mark-numbers` e `format-mark-letters`, che possono essere usate come fonte di ispirazione per creare altre funzioni di formattazione.

Vedi anche

Guida alla notazione: Sezione A.8 [Il font Emmentaler], pagina 693, Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione 5.5.1 [Allineamento degli oggetti], pagina 660.

File installati: `scm/translation-functions.scm`.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “AdHocMarkEvent” in *Guida al Funzionamento Interno*, Sezione “RehearsalMark” in *Guida al Funzionamento Interno*, Sezione “RehearsalMarkEvent” in *Guida al Funzionamento Interno*.

1.2.6 Questioni ritmiche particolari

Abbellimenti

Gli abbellimenti sono degli ornamenti musicali che hanno un carattere in corpo più piccolo e non alterano la durata della misura.

```
\relative {
  c''4 \grace b16 a4(
  \grace { b16 c16 } a2)
}
```



Esistono altri tre tipi di abbellimenti possibili; l'*acciaccatura* – un abbellimento in tempo libero indicato da una nota con legatura di portamento e un gambo barrato – e l'*appoggiatura*, che sottrae un valore determinato della nota principale a cui corrisponde e ha un gambo non barrato. È anche possibile creare un abbellimento con gambo barrato, come l'*acciaccatura*, ma privo di legatura di portamento, in modo da collocarla tra note già poste sotto una legatura: si usa il comando `\slashedGrace`.

```
\relative {
  \acciaccatura d''8 c4
  \appoggiatura e8 d4
  \acciaccatura { g16 f } e2
  \slashedGrace a,8 g4
  \slashedGrace b16 a4(
  \slashedGrace b8 a2)
}
```



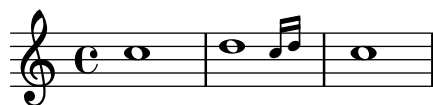
Il posizionamento degli abbellimenti è sincronizzato sui diversi righi. Nell'esempio seguente, ci sono due abbellimenti da un sedicesimo ogni abbellimento da un ottavo

```
<<
  \new Staff \relative { e''2 \grace { c16 d e f } e2 }
  \new Staff \relative { c''2 \grace { g8 b } c2 }
>>
```



Se si desidera risolvere una nota su un abbellimento, si usa il comando `\afterGrace`. Considera due argomenti: la nota principale e gli abbellimenti che la seguono.

```
\relative { c''1 \afterGrace d1 { c16[ d] } c1 }
```



In questo modo, gli abbellimenti sono collocati *dopo* l'inizio della nota principale. Il momento temporale in cui sono posti gli abbellimenti è una certa frazione della durata della nota principale. L'impostazione predefinita di

```
afterGraceFraction = 3/4
```

può essere ridefinita nel livello superiore. Ma è anche possibile specificare la frazione per ogni singolo comando `\afterGrace`.

L'esempio seguente mostra le diverse spaziature che si ottengono con la frazione predefinita, impostandola a 15/16 e infine a 1/2 della nota principale.

```
<<
  \new Staff \relative {
    c'1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff \relative {
    c'1 \afterGrace 15/16 d1 { c16[ d] } c1
  }
  \new Staff \relative {
    c'1 \afterGrace 1/2 d1 { c16[ d] } c1
  }
>>
```



Lo spazio tra la nota principale e l'abbellimento può essere definito anche attraverso delle pause spaziatrici. L'esempio seguente sposta l'abbellimento di uno spazio equivalente ai 7/8 della nota principale.

```
\new Voice \relative {
  <<
    { d'1^{\trill_( }
    { s2 s4. \grace { c16 d } }
  >>
  c1)
}
```



L'espressione musicale introdotta dal comando `\grace` avrà delle impostazioni tipografiche speciali; per esempio, per rimpicciolire il tipo di carattere e impostare le direzioni. Dunque le

modifiche che sovrascrivono tali impostazioni speciali devono essere poste all'interno del blocco `\grace`. Lo stesso vale per le modifiche che ripristinano i valori predefiniti. Nell'esempio seguente la direzione predefinita del gambo viene prima sovrascritta e poi ripristinata.

```
\new Voice \relative {
  \acciaccatura {
    \stemDown
    f''16->
    \stemNeutral
  }
  g4 e c2
}
```



Frammenti di codice selezionati

Usare il gambo barrato degli abbellimenti con le teste normali

Il gambo barrato presente nelle acciaccature può essere applicato in altre situazioni.

```
\relative c'' {
  \override Flag.stroke-style = "grace"
  c8( d2) e8( f4)
}
```



Modificare l'aspetto degli abbellimenti di un intero brano

L'aspetto di tutte le espressioni contenute nei blocchi `\grace` di un brano può essere modificato con le funzioni `add-grace-property` e `remove-grace-property`. L'esempio seguente toglie la definizione della direzione di Stem nell'abbellimento, in modo che gli abbellimenti non siano sempre rivolti in su, e barra le teste di nota.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
      \acciaccatura { f16 } g4
      \grace { d16 e } f4
      \appoggiatura { f,32 g a } e2
    }
  }
}
```



Ridefinire le impostazioni predefinite globali degli abbellimenti

Le impostazioni globali predefinite degli abbellimenti sono salvate nei seguenti identificatori:

```
startGraceMusic
stopGraceMusic
startAcciaccaturaMusic
stopAcciaccaturaMusic
startAppoggiaturaMusic
stopAppoggiaturaMusic
```

Sono definiti nel file `ly/grace-init.ly`. Ridefinendoli si possono ottenere effetti diversi.

```
startAcciaccaturaMusic = {
  <>(
    \override Flag.stroke-style = "grace"
    \slurDashed
  )

  stopAcciaccaturaMusic = {
    \revert Flag.stroke-style
    \slurSolid
    <>)
  }

  \relative c'' {
    \acciaccatura d8 c1
  }
```



Posizionare gli abbellimenti con dello spazio fluttuante

Se si imposta la proprietà `'strict-grace-spacing`, le colonne musicali degli abbellimenti 'fluttuano', ovvero si scollegano dalle note normali: prima vengono spaziate le note normali, poi le colonne musicali degli abbellimenti vengono messe a sinistra delle colonne delle note principali.

```
shiftedGrace =
#(define-music-function (offset music) (number? ly:music?)
  #{
    \override NoteHead.X-offset = #(- offset 0.85)
    \override Stem.X-offset = #offset
    \grace { $music }
    \revert NoteHead.X-offset
    \revert Stem.X-offset
  })

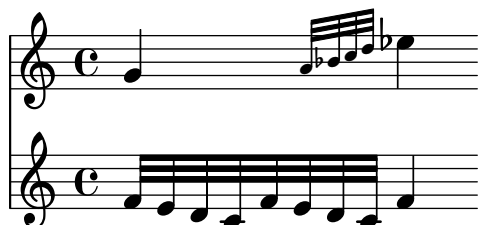
\relative c'' <<
  { g4 \shiftedGrace #-1.3 a32 \shiftedGrace #-0.5 { bes c d } es4 }
  { f,32 e d c f e d c f4 }
>>

\layout {
  \context {
    \Score
```

```

\override SpacingSpanner.strict-grace-spacing = ##t
}
}

```



Vedi anche

Glossario musicale: Sezione “acciaccatura” in *Glossario Musicale*, Sezione “acciaccatura” in *Glossario Musicale*, Sezione “appoggiatura” in *Glossario Musicale*.

Guida alla notazione: [Scalare le durate], pagina 57, [Travature manuali], pagina 101.

File installati: `ly/grace-init.ly`.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “GraceMusic” in *Guida al Funzionamento Interno*, Sezione “Grace_beam_engraver” in *Guida al Funzionamento Interno*, Sezione “Grace_auto_beam_engraver” in *Guida al Funzionamento Interno*, Sezione “Grace_engraver” in *Guida al Funzionamento Interno*, Sezione “Grace-spacing_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Una *acciaccatura* con molte note raggruppate sotto una travatura è priva della barra trasversale e ha il medesimo aspetto di una *appoggiatura* composta da varie note raggruppate sotto una travatura.

La sincronizzazione degli abbellimenti può nascondere delle sorprese, perché vengono sincronizzati anche altri elementi della notazione del rigo, come le armature di chiave, le stanghette, etc. Fai attenzione quando metti insieme rigi che hanno degli abbellimenti con rigi che non ne hanno. Ad esempio

```

<<
\new Staff \relative { e''4 \section \grace c16 d2. }
\new Staff \relative { c''4 \section d2. }
>>

```



Si può ovviare a questo problema inserendo degli abbellimenti della durata corrispondente negli altri rigi. Riprendendo l'esempio precedente

```

<<
\new Staff \relative { e''4 \section \grace c16 d2. }
\new Staff \relative { c''4 \section \grace s16 d2. }
>>

```



L'uso del comando `\grace` nella parte con le pause spaziatrici è obbligatorio, anche se la parte visibile usa `\acciaccatura` o `\appoggiatura`, perché altrimenti apparirà un orribile frammento di legatura di portamento che connette la nota di abbellimento invisibile alla nota seguente.

Le sezioni con abbellimenti devono essere usate solamente all'interno di espressioni musicali sequenziali. Non è permesso annidare o affiancare gruppi di abbellimenti; potrebbero verificarsi blocchi del programma o altri errori se non si rispetta questa limitazione.

Ogni abbellimento generato nell'output MIDI ha una durata di $1/4$ della sua vera durata. Se la durata complessiva degli abbellimenti è maggiore della durata della nota che li precede, verrà generato l'errore "Going back in MIDI time". A meno che non si diminuisca la durata degli abbellimenti, ad esempio:

```
c'8 \acciaccatura { c'8[ d' e' f' g'] }
```

diventa:

```
c'8 \acciaccatura { c'16[ d' e' f' g'] }
```

Oppure si cambia esplicitamente la durata musicale:

```
c'8 \acciaccatura { \scaleDurations 1/2 { c'8[ d' e' f' g'] } }
```

Vedi [Scalare le durate], pagina 57.

Allineamento sulle cadenze

Nell'ambito di una partitura per orchestra, le cadenze presentano un problema peculiare: quando si scrive una partitura che include una cadenza o un altro passaggio solistico, tutti gli altri strumenti devono saltare esattamente la durata complessiva delle note del passaggio, altrimenti inizieranno troppo presto o troppo tardi.

Una possibile soluzione a questo problema consiste nell'uso delle funzioni `mmrest-of-length` e `skip-of-length`. Queste funzioni Scheme prendono come argomento una sezione di musica salvata in una variabile e generano una pausa multipla o `\skip` della lunghezza esatta del brano.

```
MyCadenza = \relative {
  c'4 d8 e f g g4
  f2 g4 g
}

\new GrandStaff <<
  \new Staff {
    \MyCadenza c'1
    \MyCadenza c'1
  }
  \new Staff {
    #(mmrest-of-length MyCadenza)
    c'1
    #(skip-of-length MyCadenza)
    c'1
  }
>>
```



Vedi anche

Glossario musicale: Sezione “cadenza” in *Glossario Musicale*.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Gestione del tempo

Il tempo è gestito da `Timing_translator`, che si trova nel contesto `Score`. Un suo alias, `Timing`, viene aggiunto nel contesto nel quale si trova `Timing_translator`. Per assicurarsi che l’alias `Timing` sia disponibile, occorre istanziare esplicitamente il contesto che lo contiene (come `Voice` o `Staff`).

Si usano le seguenti proprietà di `Timing` per tenere traccia del tempo in una partitura.

`currentBarNumber`

Il numero di battuta corrente. Un esempio che mostra l’uso di questa proprietà si trova in [Numeri di battuta], pagina 112.

`measureLength`

La durata delle misure nel tempo corrente. Per un tempo di 4/4 è 1, per un tempo di 6/8 è 3/4. Il suo valore determina quando debbano essere inserite le stanghette e come debbano essere generate le travature automatiche.

`measurePosition`

Il punto della misura in cui ci si trova. Questa quantità viene reimpostata sottraendo `measureLength` ogni volta che `measureLength` viene raggiunto o superato. Quando questo accade, `currentBarNumber` viene incrementato.

`timing`

Se impostato su `#t`, le variabili precedenti sono aggiornate ad ogni momento temporale. Se impostato su `#f`, l’incisore rimane nella misura corrente per un tempo indefinito.

Si può cambiare il tempo impostando esplicitamente una qualsiasi di queste variabili. Nel prossimo esempio, viene visualizzata l’indicazione di tempo predefinita di 4/4, ma `measureLength` è impostato su 5/4. A 4/8 della terza misura, `measurePosition` si sposta in avanti di 1/8 fino a 5/8, diminuendo quella misura di 1/8. Quindi la stanghetta successiva si troverà a 9/8 invece che a 5/4.

```
\new Voice \relative {
  \set Timing.measureLength = #5/4
  c'1 c4 |
  c1 c4 |
  c4 c
  \set Timing.measurePosition = \musicLength 8*5
  b4 b b8 |
  c4 c1 |
}
```



Come mostra l'esempio, `ly:make-moment n/m` definisce una durata di n/m della nota intera. Ad esempio, `ly:make-moment 1/8` corrisponde alla durata di un ottavo mentre `ly:make-moment 7/16` a quella di sette sedicesimi.

Vedi anche

Guida alla notazione: [Numeri di battuta], pagina 112, [Musica in tempo libero], pagina 79.

Frammenti di codice: Sezione “Rhythms” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Timing_translator” in *Guida al Funzionamento Interno*, Sezione “Score” in *Guida al Funzionamento Interno*.

1.3 Segni di espressione

RONDO
Allegro

Questa sezione elenca vari segni di espressione che si possono usare in una partitura.

1.3.1 Segni di espressione collegati alle note

Questa sezione spiega come creare segni di espressione collegati alle note: articolazioni, abbellimenti e dinamiche. Sono trattati anche i metodi per creare nuove indicazioni dinamiche.

Articolazioni e abbellimenti

I diversi simboli che rappresentano articolazioni, ornamenti e altre indicazioni esecutive possono essere collegati a una nota con questa sintassi:

`nota\nome`

I valori possibili per *nome* sono elencati in Sezione A.15 [Elenco delle articolazioni], pagina 793. Ad esempio:

```
\relative {
  c'4\staccato c\mordent b2\turn
  c1\fermata
```

}



Alcune di queste articolazioni hanno delle abbreviazioni che ne semplificano l’inserimento. Le abbreviazioni sono attaccate al nome della nota e la loro sintassi è composta da un trattino - seguito da un simbolo che indica l’articolazione. Esistono abbreviazioni predefinite per *marcato*, *chiuso*, *tenuto*, *staccatissimo*, *accento*, *staccato* e *portato*. L’output corrispondente è:

```
\relative {
  c' '4-~ c-+ c-- c-!
  c4-> c-. c2-_
}
```



Le regole per il posizionamento predefinito delle articolazioni sono definite in `scm/script.scm`. Articolazioni e ornamenti possono essere posizionati manualmente sopra o sotto il rigo; si veda Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Le articolazioni sono oggetti Script. Le loro proprietà sono descritte in dettaglio in Sezione “Script” in *Guida al Funzionamento Interno*.

Le articolazioni possono essere attaccate alle pause normali, alle pause multiple e alle note. Attaccando un’articolazione a una pausa multipla si crea un oggetto `MultiMeasureRestScript`.

```
\override Script.color = #red
\override MultiMeasureRestScript.color = #blue
a'2\fermata r\fermata
R1\fermata
```



Oltre alle articolazioni, si può attaccare alle note anche un testo, posto tra virgolette o in un blocco `\markup{}`. Si veda [Scritte], pagina 253.

Ulteriori informazioni sull’ordine degli oggetti Script e TextScript collegati alle note si trovano in Sezione “Posizionamento degli oggetti” in *Manuale di Apprendimento*.

Frammenti di codice selezionati

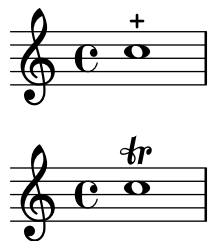
Modificare i valori predefiniti per le abbreviazioni delle articolazioni

Le abbreviazioni sono definite in ‘`ly/script-init.ly`’, dove sono assegnati valori predefiniti alle variabili `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` e `dashUnderscore`. Questi valori predefiniti possono essere modificati. Ad esempio, per associare l’abbreviazione `++` (`dashPlus`) al simbolo del *trillo* invece che al simbolo `+` predefinito, si assegna il valore `trill` alla variabile `dashPlus`:

```
\relative c' ' { c1-+ }

dashPlus = \trill
```

```
\relative c'' { c1-+ }
```



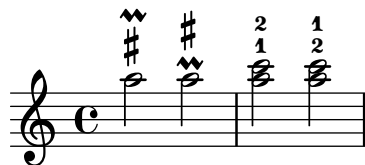
Controllo dell'ordine verticale degli script

L'ordine verticale degli script è determinato dalla proprietà `'script-priority`. Più il numero è piccolo, più sarà posto vicino alla nota. In questo esempio, il simbolo di *diesis* (oggetto `TextScript`) ha prima la priorità più bassa, dunque è posto più in basso nel primo esempio. Nel secondo, il *trillo* (oggetto `Script`) ha la priorità più bassa, quindi si trova all'interno. Quando due oggetti hanno la stessa priorità, l'ordine in cui sono inseriti determina quale viene prima.

```
\relative c''' {
  \once \override TextScript.script-priority = -100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = -100
  a2^\prall^\markup { \sharp }

  \set fingeringOrientations = #'(up)
  <c-2 a-1>2
  <a-1 c\ tweak script-priority -100 -2>2
}
```



Creare un gruppetto ritardato

Creare un gruppetto ritardato, dove la nota più bassa del gruppetto usa l'alterazione, richiede vari `\override`. La proprietà `outside-staff-priority` deve essere impostata su `#f`, perché altrimenti questa avrebbe la precedenza sulla proprietà `avoid-slur`. Cambiando la frazione $2/3$ si aggiusta la posizione orizzontale.

```
\relative c'' {
  \after 2*2/3 \turn c2( d4) r |
  \after 4 \turn c4.( d8)
  \after 4
  {
    \once \set suggestAccidentals = ##t
    \once \override AccidentalSuggestion.outside-staff-priority = ##f
    \once \override AccidentalSuggestion.avoid-slur = #'inside
    \once \override AccidentalSuggestion.font-size = -3
    \once \override AccidentalSuggestion.script-priority = -1
    \once \hideNotes
    cis8\turn \noBeam
  }
}
```



```
d4.( e8)
}
```



Vedi anche

Glossario Musicale: Sezione “tenuto” in *Glossario Musicale*, Sezione “accento” in *Glossario Musicale*, Sezione “staccato” in *Glossario Musicale*, Sezione “portato” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Posizionamento degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: [Scritte], pagina 253, Sezione 5.4.2 [Direzione e posizionamento], pagina 643, Sezione A.15 [Elenco delle articolazioni], pagina 793, [Trilli], pagina 158.

File installati: scm/script.scm.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Script” in *Guida al Funzionamento Interno*, Sezione “TextScript” in *Guida al Funzionamento Interno*.

Dinamiche

Le indicazioni dinamiche assolute si indicano con un comando che segue una nota, come ad esempio `c4\ff`. Le indicazioni dinamiche disponibili sono `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\ffffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, `\rfz` e `\n`. Le indicazioni dinamiche possono essere posizionate manualmente sopra o sotto il rigo, come è spiegato in dettaglio in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

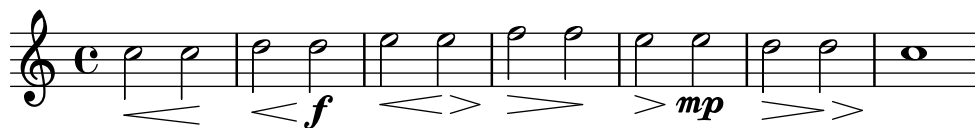
```
\relative c'' {
  c2\ppp c\mp
  c2\rfz c^\mf
  c2\spp c^\ff
}
```



Un’indicazione di *crescendo* inizia con `\<` e termina con `\!`, un’indicazione dinamica assoluta o un’ulteriore indicazione di crescendo o decrescendo. Un’indicazione di *decrescendo* inizia con `\>` e termina nello stesso modo, ovvero con `\!`, un’indicazione dinamica assoluta oppure un altro segno di crescendo o decrescendo. Si possono usare `\cr` e `\decr` al posto di `\<` e `\>`; si possono usare `\endcr` e `\enddecr` al posto di `\!` per terminare un’indicazione di crescendo o decrescendo, rispettivamente. Le *forcelle* vengono create con questa notazione.

```
\relative c'' {
  c2\< c\!
  d2\< d\f
  e2\< e\>
  f2\> f\!
  e2\> e\mp
  d2\> d\>
  c1\!
```

}



Una forcetta che termina con `\!` si estenderà fino al margine destro della nota a cui è assegnato `\!`. Nel caso in cui sia terminata con l'inizio di un altro segno di *crescendo* o *decrescendo*, si estenderà fino al centro della nota a cui è assegnato il successivo `\<` o `\>`. La forcetta successiva partirà dal margine destro della stessa nota invece che dal margine sinistro, come accade quando si termina con `\!`. Una forcetta che termina sul battere si fermerà alla stanghetta precedente.

```
\relative {
  c''1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
}
```



Le forcelle terminate con indicazioni dinamiche assolute invece che da `\!` avranno un aspetto simile. Tuttavia, la lunghezza dell'indicazione dinamica assoluta stessa può cambiare il punto in cui finisce la forcetta precedente.

```
\relative {
  c''1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
}
```



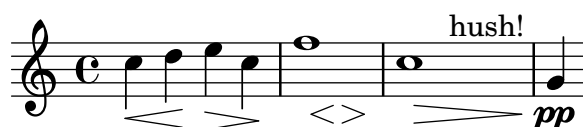
Segni come *crescendo* o *decrescendo* devono spesso iniziare o terminare nel corso di una nota sostenuta. Lo si può fare con `\after`, che può essere usato anche per creare articolazioni o scritte testuali differite:

```
<<
\relative {
  \after 2 \< c'1
  d4\> e f g
  \after 2. \pp c,1
  \after 2. \fermata e
  \after 2. ^"Fine." f
}
\relative {
  \repeat unfold 12 c'4
  c c c c\fermata
  c c c c
}
>>
```



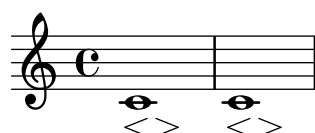
Si può usare ripetutamente `\after` per attaccare più di un'indicazione a una nota. Questo è utile soprattutto quando si aggiungono un crescendo e un decrescendo alla stessa nota:

```
\relative {
  c' '4\< d\! e\> c\!
  \after 4 \< \after 2\> \after 2. \! f1
  \textLength0n
  \after 4 \> \after 2. ~"hush!" c1
  g4\pp
}
```



Se il primo di tale sequenza di segni su una nota coincide con l'attacco della nota, conviene attaccarlo a un accordo vuoto `<>`. In questo modo tutti i segni possono essere inseriti nel loro ordine visivo naturale:

```
{
  <>\< \after 4 \> \after 2 \! c'1
  % più facile da leggere e scrivere di:
  \after 4 \> \after 2 \! c'1\<
}
```



Il comando `\espressivo` può essere usato anche per indicare un crescendo e un decrescendo sulla stessa nota. Tuttavia, si tenga presente che viene implementato come articolazione, non come dinamica.

```
\relative {
  c' '2 b4 a
  g1\espressivo
  \after 2. \espressivo c
}
```



Le indicazioni di crescendo testuali iniziano con `\cresc`, quelle di decrescendo con `\decre` o `\dim`. Le linee di estensione sono aggiunte automaticamente.

```
\relative {
  g'8\cresc a b c b c d e\mf |
```

```
f8\decrese d c e\> d c b |
a1\dim ~ |
a2. r4\! |
}
```



Le indicazioni testuali per i cambi di dinamica possono essere impiegate anche per sostituire le forcelle:

```
\relative c'' {
  \crescTextCresc
  c4\< d e f\! |
  \dimTextDecresc
  g4\> e d c\! |
  \dimTextDecr
  e4\> d c b\! |
  \dimTextDim
  d4\> c b a\! |
  \crescHairpin
  \dimHairpin
  c4\< d\! e\> d\! |
}
```



Per creare nuove indicazioni dinamiche assolute o testi da allineare alle dinamiche, si veda [Nuove indicazioni dinamiche], pagina 140.

Il posizionamento verticale della dinamica è gestito da Sezione “DynamicLineSpanner” in *Guida al Funzionamento Interno*.

Esiste un contesto Dynamics che permette di posizionare le indicazioni dinamiche su un'apposita linea orizzontale. Si usano le pause spaziatrici per indicarne la collocazione temporale (le note in un contesto Dynamics occupano infatti il rispettivo valore musicale, ma senza comparire sul rigo). Il contesto Dynamics può contenere altri elementi utili come indicazioni testuali, estensori del testo e indicazioni di pedalizzazione del pianoforte.

```
<<
\new Staff \relative {
  c'2 d4 e |
  c4 e e,2 |
  g'4 a g a |
  c1 |
}
\new Dynamics {
  s1\< |
  s1\f |
  s2\dim s2-"rit." |
  s1\p |
}
```

>>



Comandi predefiniti

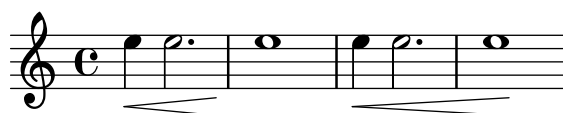
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`,
`\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

Frammenti di codice selezionati

Impostare il comportamento delle forcelle sulle stanghette

Se la nota che termina una forcella si trova sul primo battito di una battuta, la forcella si ferma prima della stanghetta che precede la nota. Si può controllare questo comportamento modificando la proprietà `'to-barline`.

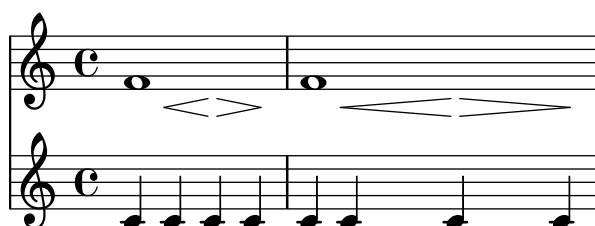
```
\relative c' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



Impostare la lunghezza minima delle forcelle

Se le forcelle sono troppo corte, possono essere allungate modificando la proprietà `minimum-length` dell'oggetto `Hairpin`.

```
<<
{
  \after 4 \< \after 2 \> \after 2. \! f'1
  \override Hairpin.minimum-length = 8
  \after 4 \< \after 2 \> \after 2. \! f'1
}
{
  \repeat unfold 8 c'4
}
>>
```



Allineare le estremità delle forcelle alle direzioni di NoteColumn

Le estremità delle forcelle possono essere allineate alle direzioni LEFT, CENTER o RIGHT dei grob NoteColumn modificando la proprietà `endpoint-alignments`, che è costituita da una coppia di numeri che rappresentano le estremità sinistra e destra della forcella. `endpoint-alignments` devono essere direzioni (-1, 0 o 1). Valori diversi verranno trasformati emettendo un avviso. Non ha effetto quando l'estremità destra di una forcella termina su una pausa; in questo caso termina sempre sul margine sinistro della pausa.

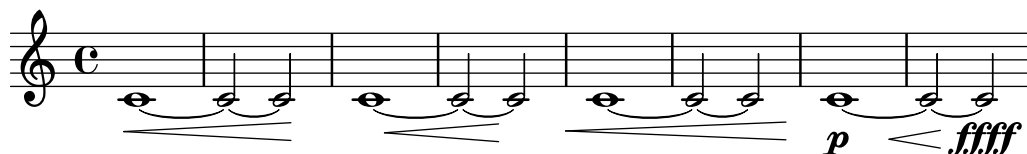
```
{
  c'2\< <c' d'\>\! |
  \override Hairpin.endpoint-alignments = #'(1 . -1)
  c'2\< <c' d'\>\! |
  \override Hairpin.endpoint-alignments = #'(,LEFT . ,CENTER)
  c'2\< <c' d'\>\! |
}
```



Spostare le estremità delle forcelle

Le estremità delle forcelle possono essere spostate in modo relativo alla loro posizione predefinita (`offset`) impostando la proprietà `shorten-pair` dell'oggetto `Hairpin`. Valori positivi spostano le estremità a destra, valori negativi le spostano a sinistra. Diversamente dalla proprietà `minimum-length`, questa proprietà modifica solo l'aspetto della forcella; non cambia la spaziatura orizzontale (inclusa la posizione delle dinamiche confinanti). Questo metodo è quindi utile per ritoccare una forcella entro lo spazio ad essa allocato.

```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```



Forcelle con notazione al niente

Le forcelle di dinamica possono essere rappresentate con una punta tonda (notazione “al niente”) impostando la proprietà `circled-tip` dell'oggetto `Hairpin` su `#t`.

```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
```

```
c4\> c\< c2\!
```

```
}
```



Stampare le forcelle in vari stili

Il segno di dinamica della forcilla può avere diversi stili

```
\relative c' ' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\s fz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\s fz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\s fz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\s fz\> a a a\!
}
```



Dinamiche e segni testuali allineati verticalmente

Tutti gli oggetti `DynamicLineSpanner` (forcelle e testi di dinamica) sono posti a una distanza minima dal rigo determinata da `'staff-padding`. Se si imposta `'staff-padding` su un valore abbastanza grande, le dinamiche saranno allineate.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = 3
}
```

```

\textLength0n
\override TextScript.staff-padding = 1
\music
}

```



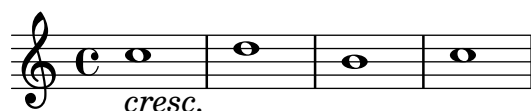
Nascondere la linea di estensione per le dinamiche testuali

I cambi di dinamica in stile testuale (come *cresc.* e *dim.*) appaiono con una linea tratteggiata che mostra la loro estensione. Questa linea può essere soppressa nel modo seguente:

```

\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}

```



Cambiare gli stili del testo e degli estensori per le dinamiche testuali

Il testo usato per i *crescendo* e i *decrescendo* può essere cambiato modificando le proprietà di contesto *crescendoText* e *decrescendoText*.

Lo stile della linea dell'estensore può essere cambiato modificando la proprietà 'style di *DynamicTextSpanner*. Il valore predefinito è 'dashed-line; gli altri valori possibili sono 'line, 'dotted-line e 'none.

```

\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}

```



Vedi anche

Glossario Musicale: Sezione “al niente” in *Glossario Musicale*, Sezione “crescendo” in *Glossario Musicale*, Sezione “decrescendo” in *Glossario Musicale*, Sezione “forcella” in *Glossario Musicale*. Manuale di apprendimento: Sezione “Articolazioni e dinamiche” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Nuove indicazioni dinamiche], pagina 140, Sezione 3.5.9 [Miglioramento dell’output MIDI], pagina 549, Sezione 3.5.4 [Gestione delle dinamiche nel MIDI], pagina 539.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “DynamicText” in *Guida al Funzionamento Interno*, Sezione “Hairpin” in *Guida al Funzionamento Interno*, Sezione “DynamicLineSpanner” in *Guida al Funzionamento Interno*, Sezione “Dynamics” in *Guida al Funzionamento Interno*.

Nuove indicazioni dinamiche

Il modo più semplice per creare indicazioni dinamiche è usare gli oggetti `\markup`.

```
moltoF = \markup { molto \dynamic f }
```

```
\relative {
  <d' e>16_\moltoF <d e>
  <d e>2..
}
```



In modalità markup, si possono creare dinamiche editoriali (racchiuse tra parentesi normali o quadrate). La sintassi della modalità markup è descritta in Sezione 1.8.2 [Formattazione del testo], pagina 260.

```
roundF = \markup {
  \center-align \concat { \bold { \italic ( }
    \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative {
  c'1_\roundF
  c1_\boxF
}
```



È possibile creare semplicemente indicazioni dinamiche centrate verticalmente con la funzione `make-dynamic-script`.

```
sfzp = #(make-dynamic-script "sfzp")
\relative {
  c'4 c c\sfp c
}
```



In generale, `make-dynamic-script` assume come argomento qualsiasi oggetto markup. Il tipo di carattere per la dinamica contiene solo i caratteri `f`, `m`, `p`, `r`, `s` e `z`; dunque, se si desidera creare un'indicazione dinamica che contenga testo semplice e simboli di punteggiatura, occorre usare dei comandi markup che ripristinino la famiglia e la codifica del tipo di carattere per il testo normale, ad esempio `\normal-text`. Il vantaggio nell'uso di `make-dynamic-script` al posto di un normale markup è l'allineamento verticale degli oggetti markup e delle forcelle collegate alla stessa testa di nota.

```
roundF = \markup { \center-align \concat {
  \normal-text { \bold { \italic ( } }
  \dynamic f
  \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
  \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative {
  c'4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g1
  g'1~\mfEspressDynamic
  g1
}
```



Si può usare anche la forma Scheme della modalità markup. La sintassi è spiegata in Sezione “Markup construction in Scheme” in *Estendere*.

```
moltoF = #(make-dynamic-script
  (markup #:normal-text "molto"
    #:dynamic "f"))
\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF
}
```



Per allineare a sinistra il testo di dinamica invece di centrarlo su una nota, si usa un `\tweak`:

```
moltoF = \tweak DynamicText.self-alignment-X #LEFT
```

```

      #(make-dynamic-script
        (markup #:normal-text "molto"
          #:dynamic "f"))
\relative {
  <d' e>16 <d e>
  <d e>2..\moltoF <d e>1
}

```



Le impostazioni dei tipi di carattere in modalità markup sono descritti in [Scelta del tipo di carattere e della dimensione], pagina 261.

Vedi anche

Guida alla notazione: Sezione 1.8.2 [Formattazione del testo], pagina 260, [Scelta del tipo di carattere e della dimensione], pagina 261, Sezione 3.5.9 [Miglioramento dell'output MIDI], pagina 549, Sezione 3.5.4 [Gestione delle dinamiche nel MIDI], pagina 539.

Extending LilyPond: Sezione “Markup construction in Scheme” in *Estendere*.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

1.3.2 Indicazioni espressive curvilinee

Questa sezione spiega come creare varie indicazioni espressive con forma curvilinea: legature di portamento, legature di frase, respiri, portamenti indeterminati discendenti (cadute) o ascendenti.

Legature di portamento

Le *legature di portamento* si inseriscono con delle parentesi:

Nota: Nella musica polifonica, una legatura di portamento deve terminare nella stessa voce in cui è iniziata.

```

\relative {
  f''4( g a) a8 b(
  a4 g2 f4)
  <c e>2( <b d>2)
}

```



Le legature di portamento possono essere posizionate manualmente sopra o sotto il rigo, come è spiegato in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Tracciare due legature di portamento simultanee o sovrapposte richiede una particolare attenzione. Di solito le legature di portamento più esterne indicano in realtà una legatura di frase e le legature di frase possono essere sovrapposte a una normale legatura, vedi [Legature di frase], pagina 145. Quando invece si vogliono usare molteplici legature di portamento normali in una

sola voce, l'inizio e la fine di ogni legatura devono essere preceduti da un `\=` seguito da una chiave identificativa (un simbolo o un numero intero non negativo).

```
\fixed c' {
  <c~ f\=1( g\=2( >2 <c e\=1) a\=2) >
}
```



Le legature di portamento possono essere continue, punteggiate o tratteggiate. Lo stile predefinito è quello continuo:

```
\relative {
  c'4( e g2)
  \slurDashed
  g4( e c2)
  \slurDotted
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```



Le legature di portamento possono essere anche semitratteggiate (half-dashed), ovvero con la prima metà tratteggiata e la seconda continua; oppure semicontinue (half-solid), ovvero con la prima metà continua e la seconda tratteggiata:

```
\relative {
  c'4( e g2)
  \slurHalfDashed
  g4( e c2)
  \slurHalfSolid
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```



Si possono definire modelli di tratteggio personalizzati per le legature di portamento:

```
\relative {
  c'4( e g2)
  \slurDashPattern #0.7 #0.75
  g4( e c2)
  \slurDashPattern #0.5 #2.0
  c4( e g2)
  \slurSolid
  g4( e c2)
}
```

}



Comandi predefiniti

`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurHalfDashed`,
`\slurHalfSolid`, `\slurDashPattern`, `\slurSolid`.

Frammenti di codice selezionati

Uso delle doppie legature di portamento per gli accordi legati

Alcuni compositori scrivono due *legature di portamento* per indicare gli accordi legati. Si può ottenere questo risultato impostando `doubleSlurs`.

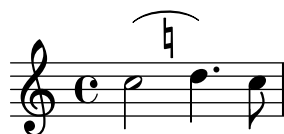
```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Posizionare il testo a margine dentro le legature di portamento

I testi a margine devono avere la proprietà `outside-staff-priority` impostata su `false` per poter apparire dentro le legature di portamento.

```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(~\markup { \halign #-10 \natural } d4.) c8
}
```



Legature di portamento con complesse strutture di tratteggio

Le legature di portamento possono avere schemi di tratteggio complessi definendo la proprietà `dash-definition`. `dash-definition` è una lista di `dash-elements`. Un `dash-element` è una lista di parametri che definiscono il comportamento del tratteggio per un segmento della legatura.

La legatura di portamento è definita come il parametro `t` della curva di bezier che va da 0 sul margine sinistro della legatura fino a 1 su quello destro. `dash-element` è una lista di (`inizio-t fine-t frazione-trattino punto-trattino`). La regione della legatura di portamento che va da `inizio-t` a `fine-t` avrà una frazione `frazione-trattino` di ogni `punto-trattino` nero. `punto-trattino` viene definito in spazi rigo. `frazione-trattino` è impostato su 1 per una legatura di portamento continua.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'(( 0 0.3 0.1 0.75)
```

```

                                (0.3  0.6  1    1    )
                                (0.65 1.0  0.4 0.75))

c4( d e f)
\once \override
  Slur.dash-definition = #'((0    0.25  1    1    )
                             (0.3  0.7   0.4 0.75)
                             (0.75 1.0   1    1    ))

c4( d e f)
}

```



Vedi anche

Glossario Musicale: Sezione “legatura di portamento” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Sul non annidamento di parentesi e legature di valore” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Legature di frase], pagina 145.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Slur” in *Guida al Funzionamento Interno*.

Legature di frase

Le *legature di frase*, che indicano una frase musicale, si scrivono con i comandi `\(` e `\)`:

```

\relative {
  c'4\ ( d( e) f(
  e2) d\ )
}

```



A livello tipografico, una legatura di frase si comporta in modo pressoché identico a una normale legatura di portamento. Sono però trattate come oggetti diversi; ad esempio, `\slurUp` non ha effetto su una legatura di frase. Le legature di frase possono essere posizionate sopra o sotto il rigo, come è spiegato in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Per inserire più legature di frase simultanee o sovrapposte si usa `\=`, come per le normali legature di portamento (vedi [Legature di portamento], pagina 142).

Le legature di frase possono essere continue, puntate o tratteggiate. Lo stile predefinito è quello continuo:

```

\relative {
  c'4\ ( e g2\ )
  \phrasingSlurDashed
  g4\ ( e c2\ )
  \phrasingSlurDotted
  c4\ ( e g2\ )
  \phrasingSlurSolid
  g4\ ( e c2\ )
}

```

}



Le legature di frase possono essere anche semitratteggiate (la prima metà tratteggiata, la seconda continua) o semicontinue (la prima metà continua, la seconda tratteggiata):

```
\relative {
  c'4\ ( e g2\
  \phrasingSlurHalfDashed
  g4\ ( e c2\
  \phrasingSlurHalfSolid
  c4\ ( e g2\
  \phrasingSlurSolid
  g4\ ( e c2\
}
```



Si possono definire modelli di tratteggio personalizzati anche per le legature di frase:

```
\relative {
  c'4\ ( e g2\
  \phrasingSlurDashPattern #0.7 #0.75
  g4\ ( e c2\
  \phrasingSlurDashPattern #0.5 #2.0
  c4\ ( e g2\
  \phrasingSlurSolid
  g4\ ( e c2\
}
```



Le definizioni dei modelli di tratteggio per le legature di frase hanno la stessa struttura di quelle per le legature di portamento. Per maggiori informazioni sui modelli complessi di tratteggio si consultino i frammenti in [Legature di portamento], pagina 142.

Comandi predefiniti

```
\phrasingSlurUp, \phrasingSlurDown, \phrasingSlurNeutral, \phrasingSlurDashed,
\phrasingSlurDotted, \phrasingSlurHalfDashed, \phrasingSlurHalfSolid,
\phrasingSlurDashPattern, \phrasingSlurSolid.
```

Vedi anche

Manuale di apprendimento: Sezione “Sul non annidamento di parentesi e legature di valore” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Legature di portamento], pagina 142.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “PhrasingSlur” in *Guida al Funzionamento Interno*.

Respiri

I respiri si inseriscono col comando `\breathe`:

```
{ c''2. \breathe d''4 }
```



Diversamente da altri segni di espressione, il respiro non è associato alla nota precedente ma è un evento musicale separato. Dunque tutti i segni espressivi attaccati alla nota precedente, tutte le parentesi quadre che indicano le travature manuali e le parentesi che indicano le legature di portamento e di frase devono essere poste prima di `\breathe`.

Un respiro termina una travatura automatica; per evitare questo comportamento, si veda [Travature manuali], pagina 101.

```
\relative { c''8 \breathe d e f g2 }
```



È supportata la `divisio`, indicatore del respiro nella musica antica. Maggiori dettagli in [Divisiones], pagina 467.

Frammenti di codice selezionati

Cambiare il simbolo del segno di respiro

Il glifo del respiro può essere modificato sovrascrivendo la proprietà `text` dell’oggetto di formattazione `BreathingSign` con qualsiasi testo incluso in un blocco markup.

```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}
```



Usare un segno di spunta come simbolo di respiro

La musica vocale e per fiati usa frequentemente il segno di spunta come segno di respiro. Questo indica un respiro che sottrae un po’ di tempo alla nota precedente invece di prendere una piccola pausa, indicata dal segno di respiro rappresentato dalla virgola. Il segno può essere spostato un po’ su per allontanarlo dal rigo.

```
\relative c'' {
  c2
  \breathe
  d2
}
```



```

\override BreathingSign.Y-offset = #2.6
\override BreathingSign.text =
  \markup { \musicglyph "scripts.tickmark" }
c2
\breathe
d2
}

```



Inserire una cesura

I segni di cesura possono essere creati sovrascrivendo la proprietà 'text dell'oggetto BreathingSign. È disponibile anche un segno di cesura curvo.

```

\relative c' ' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
g8 e'4. \breathe g8. e16 c4
}

```



Vedi anche

Glossario Musicale: Sezione “cesura” in *Glossario Musicale*.

Guida alla notazione: [Divisiones], pagina 467.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “BreathingEvent” in *Guida al Funzionamento Interno*, Sezione “BreathingSign” in *Guida al Funzionamento Interno*, Sezione “Breathing-sign-engraver” in *Guida al Funzionamento Interno*.

Portamenti indeterminati discendenti (cadute) e ascendenti

I portamenti indeterminati verso il basso (cadute) e verso l'alto possono essere aggiunti alle note col comando `\bendAfter`. La direzione del portamento è indicata con un più o un meno (su o giù). Il numero indica l'intervallo per cui il portamento si estenderà *oltre* la nota principale.

```

\relative c' ' {
  c2\bendAfter #+4
  c2\bendAfter #-4
  c2\bendAfter #+6.5
  c2\bendAfter #-6.5
  c2\bendAfter #+8
  c2\bendAfter #-8
}

```



Frammenti di codice selezionati

Cambiare la forma dei portamenti indeterminati verso il basso o verso l'alto

La proprietà `shortest-duration-space` può essere modificata per cambiare la forma dei portamenti indeterminati verso il basso o verso l'alto.

```
\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = 4.0
  c2-\bendAfter 5
  c2-\bendAfter -4.75
  c2-\bendAfter 8.5
  c2-\bendAfter -6
}
```



Vedi anche

Glossario Musicale: Sezione “portamento indeterminato verso il basso” in *Glossario Musicale*,
Sezione “portamento indeterminato verso l'alto” in *Glossario Musicale*.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

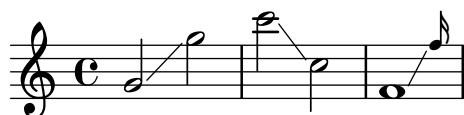
1.3.3 Indicazioni espressive lineari

Questa sezione spiega come creare varie indicazioni espressive che seguono una traiettoria lineare: glissandi, arpeggi e trilli.

Glissando

Un *glissando* si crea attaccando `\glissando` a una nota:

```
\relative {
  g'2\glissando g'
  c2\glissando c,
  \afterGrace f,1\glissando f'16
}
```



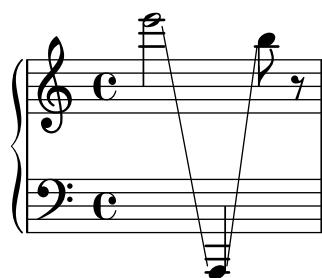
Un glissando può collegare note appartenenti a righi diversi:

```
\new PianoStaff <<
  \new Staff = "right" {
    e'''2\glissando
    \change Staff = "left"
    a,,4\glissando
    \change Staff = "right"
    b''8 r |
```

```

}
\new Staff = "left" {
  \clef bass
  s1
}
>>

```



Un glissando può collegare le note negli accordi. Se è necessario qualcosa di diverso dal normale abbinamento uno a uno delle note, si possono definire le connessioni tra le note impostando `\glissandoMap` su una lista Scheme. Gli elementi sono coppie di numeri interi; ciascuna coppia $(x . y)$ crea una linea di glissando dalla nota x del primo accordo alla nota y del secondo accordo. Le note sono numerate a partire da zero nell'ordine in cui appaiono nel file di input `.ly`. Non tutte le note devono far parte di un glissando.

```

\relative {
  <c' e>2\glissando g'
  <c, e>\glissando <g' b>
  \break
  \set glissandoMap = #'((0 . 1) (1 . 0))
  <c, g'>\glissando <d a'>
  \set glissandoMap = #'((0 . 0) (0 . 1) (0 . 2))
  c\glissando <d f a>
  \set glissandoMap = #'((2 . 2) (0 . 0))
  <f d a'>\glissando <c f c'>
}

```



Si possono adottare diversi stili di glissando. Maggiori dettagli in Sezione 5.4.8 [Stili della linea], pagina 658.

Frammenti di codice selezionati

Glissando contemporaneo

Un glissando contemporaneo senza una nota finale può essere creato usando una nota nascosta e un tempo di cadenza.

```

\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag

```

```

c4 c
\cadenzaOn
c4\glissando
\hideNotes
c,,4
\unHideNotes
\cadenzaOff
\bar "|"
}

```



Aggiungere i segni di tempo per i glissandi lunghi

I battiti saltati nei glissandi molto lunghi vengono talvolta segnalati con delle indicazioni di tempo, che consistono solitamente in dei gambi privi di teste di nota. Questi gambi possono essere usati anche per contenere segni di espressione intermedi.

Se i gambi non si allineano bene al glissando, può essere necessario riposizionarli leggermente.

```

glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando \glissandoSkipOn f4 g a |
  a8\noBeam \glissandoSkipOff a8
  r8 f8\glissando \glissandoSkipOn g4 a8 \glissandoSkipOff a8 |
  r4 f\glissando\< \glissandoSkipOn a4\> \glissandoSkipOff b8\! r |
}

```



Lasciare che i glissandi vadano a capo

Per permettere a un glissando di andare a capo se capita su un'interruzione di riga, si impostano le proprietà `breakable` e `after-line-breaking` su `##t`:

```

glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

```

```

music = {
  \repeat unfold 16 f8 |
  f1\glissando |
  a4 r2. |
  \repeat unfold 16 f8 |
  f1\glissando \once\glissandoSkipOn |
  a2 a4 r4 |
  \repeat unfold 16 f8
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable
               set to \typewriter "#t" }
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  \music
}

\relative c'' {
  <>^\markup { \typewriter Glissando.breakable not set }
  \music
}

\paper {
  line-width = 100\mm
}

```



Estendere i glissandi sulle volte delle ripetizioni

Un glissando che si estende in vari blocchi `\alternative` può essere simulato aggiungendo all'inizio di ogni blocco `\alternative` una nota di abbellimento nascosta da cui inizia un glissando. La nota di abbellimento deve avere la stessa altezza della nota da cui parte il glissando iniziale. In questo frammento si usa una funzione musicale che prende come argomento l'altezza della nota di abbellimento.

Attenzione: nella musica polifonica la nota di abbellimento deve avere una nota di abbellimento corrispondente in tutte le altre voci.

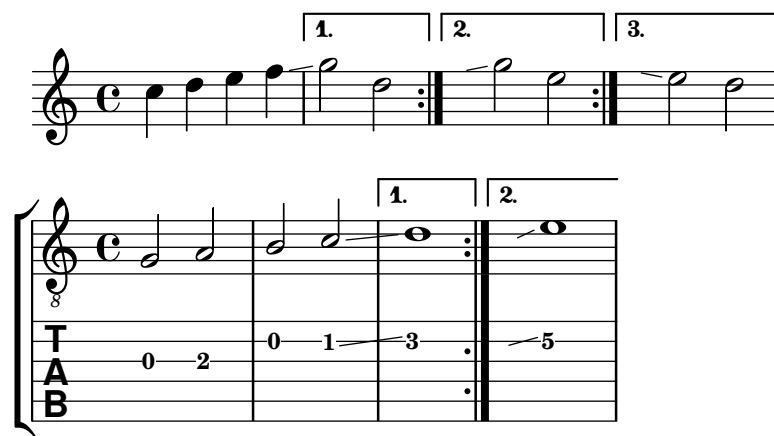
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = 3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music = \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
  \new TabStaff <<
    \new TabVoice { \clef "moderntab" \music }
  >>
}>>
```

}



Vedi anche

Glossario Musicale: Sezione “glissando” in *Glossario Musicale*.

Guida alla notazione: Sezione 5.4.8 [Stili della linea], pagina 658.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Glissando” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Non è supportato il testo lungo la linea del glissando (ad esempio *gliss.*).

Arpeggio

Un *arpeggio* su un accordo (detto anche accordo spezzato) si ottiene aggiungendo `\arpeggio` all'accordo:

```
\relative { <c' e g c>1\arpeggio }
```



Si possono scrivere vari tipi di arpeggio. `\arpeggioNormal` ripristina l'arpeggio normale:

```
\relative {
  <c' e g c>2\arpeggio
```

```
\arpeggioArrowUp
<c e g c>2\arpeggio
```

```
\arpeggioArrowDown
<c e g c>2\arpeggio
```

```
\arpeggioNormal
<c e g c>2\arpeggio
}
```



Questi comandi predefiniti modificano la proprietà 'arpeggio-direction; si può leggere la loro completa definizione nel file `ly/property-init.ly`.

Si possono creare simboli di arpeggio speciali *in forma di parentesi*:

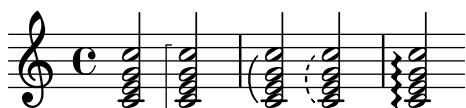
```
\relative {
  <c' e g c>2

  \arpeggioBracket
  <c e g c>2\arpeggio

  \arpeggioParenthesis
  <c e g c>2\arpeggio

  \arpeggioParenthesisDashed
  <c e g c>2\arpeggio

  \arpeggioNormal
  <c e g c>2\arpeggio
}
```



Questi comandi predefiniti sovrascrivono la proprietà 'stencil dell'oggetto Arpeggio e possono anche adattare il suo 'X-extent (ovvero, la dimensione orizzontale necessaria per non entrare in collisione con altri oggetti).

Le proprietà del tratteggio della parentesi dell'arpeggio sono regolate dalla proprietà 'dash-definition, descritta in [Legature di portamento], pagina 142.

Gli arpeggi possono essere scritti in modo esplicito con le legature di valore. Per maggiori dettagli si veda [Legature di valore], pagina 58.

Comandi predefiniti

`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`, `\arpeggioParenthesis`, `\arpeggioParenthesisDashed`.

Frammenti di codice selezionati

Creare degli arpeggi che attraversano il rigo del pianoforte

In un rigo per pianoforte (`PianoStaff`), è possibile far sì che un *arpeggio* attraversi i righi impostando la proprietà `PianoStaff.connectArpeggios`.

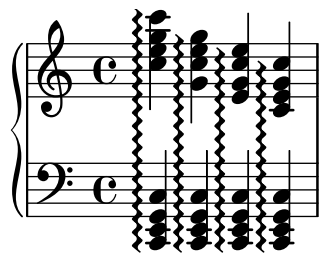
```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
```



```

    <c,, e g c>4\arpeggio
  }
}
>>

```



Creare degli arpeggi che attraversano i righi in altri contesti

Si possono creare *arpeggi* che attraversano i righi in contesti diversi da `GrandStaff`, `PianoStaff` e `StaffGroup` se l'incisore `Span_arpeggio_engraver` è incluso nel contesto `Score`.

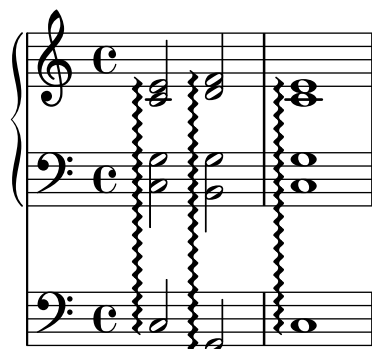
```

<<
  \new PianoStaff <<
    \new Voice \relative c' {
      <c e>2\arpeggio <d f>2\arpeggio
      <c e>1\arpeggio
    }
    \new Voice \relative c {
      \clef bass
      <c g'>2\arpeggio <b g'>2\arpeggio
      <c g'>1\arpeggio
    }
  >>

  \new Staff \relative c {
    \set Score.connectArpeggios = ##t
    \clef bass
    c2\arpeggio g\arpeggio
    c1\arpeggio
  }
>>

\layout {
  \context {
    \Score
    \consists "Span_arpeggio_engraver"
  }
}

```



Creare degli arpeggi che attraversano note appartenenti a voci diverse

Si può disegnare un *arpeggio* che attraversa delle note in voci diverse dello stesso rigo se si aggiunge l'incisore `Span_arpeggio_engraver` nel contesto `Staff`:

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Vedi anche

Glossario Musicale: Sezione “arpeggio” in *Glossario Musicale*.

Guida alla notazione: [Legature di portamento], pagina 142, [Legature di valore], pagina 58.

File installati: `ly/property-init.ly`.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Arpeggio” in *Guida al Funzionamento Interno*, Sezione “Slur” in *Guida al Funzionamento Interno*, Sezione “PianoStaff” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

I comandi predefiniti come `\arpeggioArrowUp` si applicano soltanto al contesto corrente e dunque non hanno effetto sugli arpeggi che attraversano varie voci o righi. In tali casi questi comandi devono essere usati in un blocco `\context` all’interno di `\layout` o in un blocco `\with`, come è spiegato in Sezione 5.1.5 [Modifica delle impostazioni predefinite di un contesto], pagina 615. Altrimenti, invece di usare i comandi predefiniti, potrebbe convenire sovrascrivere direttamente le proprietà interessate per l’oggetto `Arpeggio` nel contesto appropriato, per esempio:

```
\override Staff.Arpeggio.stencil = #ly:chord-bracket::print
```

per far sì che le parentesi dell'arpeggio attraversino le voci al livello del rigo (Staff), oppure

```
\override PianoStaff.Arpeggio.arpeggio-direction = #UP
```

per ottenere arpeggi che attraversano il rigo e con la freccia che punta in sù in un contesto PianoStaff.

Non è possibile mostrare simultaneamente arpeggi connessi e non connessi in un PianoStaff.

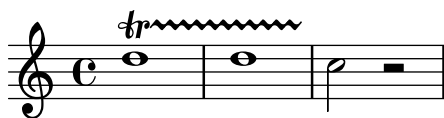
L'arpeggio in forma di parentesi non può essere impostato con facilità negli arpeggi che attraversano i righi; occorre ricorrere a metodi più complessi descritti in [Gambi trasversali ai righi], pagina 350.

Trilli

I trilli senza linea di estensione si ottengono col comando `\trill`; si veda [Articolazioni e abbellimenti], pagina 129.

I trilli con linea di estensione si ottengono con `\startTrillSpan` e `\stopTrillSpan`:

```
\relative {
  d''1\startTrillSpan
  d1
  c2\stopTrillSpan
  r2
}
```



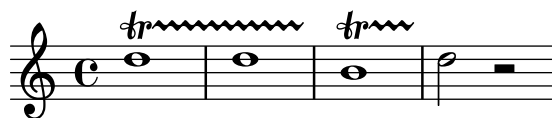
Un estensore del trillo che va a capo ricomincerà esattamente sopra la prima nota della nuova riga.

```
\relative {
  d''1\startTrillSpan
  \break
  d1
  c2\stopTrillSpan
  r2
}
```



È possibile tracciare trilli consecutivi senza dover esplicitare i comandi `\stopTrillSpan`, perché il trillo successivo diventerà automaticamente il limite destro di quello precedente.

```
\relative {
  d''1\startTrillSpan
  d1
  b1\startTrillSpan
  d2\stopTrillSpan
  r2
}
```



I trilli possono essere anche combinati con le note di abbellimento. La sintassi di questo costrutto e il metodo per posizionare in modo preciso gli abbellimenti sono descritti in [Abbellimenti], pagina 122.

```
\relative {
  d''1~\afterGrace
  d1\startTrillSpan { c32[ d]\stopTrillSpan }
  c2 r2
}
```



I trilli che richiedono una nota ausiliaria dall'altezza esplicita si ottengono col comando `\pitchedTrill`. Il primo argomento è la nota principale e il secondo è la nota *trillata*, che appare come una testa di nota senza gambo e racchiusa tra parentesi.

```
\relative {
  \pitchedTrill
  d''2\startTrillSpan fis
  d2
  c2\stopTrillSpan
  r2
}
```



L'alterazione del primo trillo con notina in una misura viene sempre visualizzata, anche per i bequadri.

```
{
  \key d \major
  \pitchedTrill
  d'2\startTrillSpan cis d\stopTrillSpan
  \pitchedTrill
  d2\startTrillSpan c d\stopTrillSpan
  \pitchedTrill
  d2\startTrillSpan e d\stopTrillSpan
}
```



Alterazioni successive (della stessa nota nella stessa misura) devono essere aggiunte manualmente.

```
\relative {
  \pitchedTrill
```

```

eis''4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
}

```



Comandi predefiniti

\startTrillSpan, \stopTrillSpan.

Vedi anche

Glossario Musicale: Sezione “trillo” in *Glossario Musicale*.

Guida alla notazione: [Articolazioni e abbellimenti], pagina 129, [Abbellimenti], pagina 122.

Frammenti: Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TrillSpanner” in *Guida al Funzionamento Interno*, Sezione “TrillPitchHead” in *Guida al Funzionamento Interno*, Sezione “TrillPitchAccidental” in *Guida al Funzionamento Interno*, Sezione “TrillPitchParentheses” in *Guida al Funzionamento Interno*, Sezione “TrillPitchGroup” in *Guida al Funzionamento Interno*, Sezione “Pitched_trill_engraver” in *Guida al Funzionamento Interno*.

1.4 Ripetizioni



La ripetizione è un concetto chiave in musica e può essere resa con varie forme di notazione. LilyPond supporta i seguenti tipi di ripetizioni:

volta	La musica ripetuta non viene scritta per intero ma racchiusa tra barre di ripetizione. Se la ripetizione si trova all'inizio di un brano, la stanghetta di ritornello è posta soltanto alla fine della ripetizione. I finali alternativi (volte) appaiono da sinistra a destra e sono evidenziati da delle parentesi. Questa è la notazione standard per le ripetizioni con finali alternativi.
unfold	La musica ripetuta viene scritta per intero, tante volte quante sono specificate dal <i>numero-ripetizioni</i> . È utile quando si scrive musica ripetitiva.
percent	Si tratta di ripetizioni del singolo tempo (battito) o della battuta. Hanno l'aspetto di una barra obliqua o di segni di percentuale.
tremolo	Si usa per scrivere travature a tremolo.

1.4.1 Ripetizioni lunghe

Questa sezione spiega come inserire ripetizioni lunghe (solitamente di più battute). Tali ripetizioni possono essere in due forme: racchiuse tra segni di ritornello oppure ricopiate interamente (adatte a scrivere musica ripetitiva). Si possono anche controllare manualmente i segni di ripetizione.

Ripetizioni normali

La sintassi per una normale ripetizione è

```
\repeat volta numero-di-ripetizioni espressione-musicale
```

dove *espressione-musicale* è la musica da ripetere.

Un'unica ripetizione senza finale alternativo:

```
\relative {
  \repeat volta 2 { c''4 d e f }
  c2 d
  \repeat volta 2 { d4 e f g }
}
```



Il segno di inizio della ripetizione, per impostazione predefinita, non appare nella prima misura. È tuttavia possibile aggiungerlo inserendo a mano la battuta `\bar ".|:"` prima della prima nota.

```
\relative {
  \repeat volta 2 { \bar ".|:" c''4 d e f }
  c2 d
  \repeat volta 2 { d4 e f g }
}
```



I finali alternativi si ottengono con `\alternative`. Ogni gruppo di alternative deve essere a sua volta racchiuso tra parentesi.

```
\repeat volta numero-di-ripetizioni espressione-musicale
```

```
\alternative {
  { espressione-musicale }
}
```

dove *espressione-musicale* è la musica.

Se il numero di ripetizioni è superiore a quello dei finali alternativi, alle prime ripetizioni viene assegnata la prima alternativa.

Una singola ripetizione con un finale alternativo:

```
\relative {
  \repeat volta 2 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
  c1
}
```



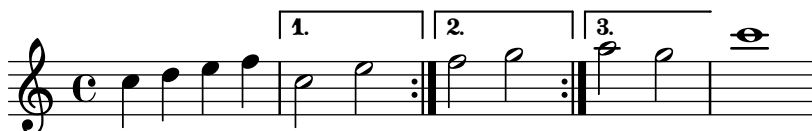
Molteplici ripetizioni con un finale alternativo:

```
\relative {
  \repeat volta 4 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
  }
  c1
}
```



Molteplici ripetizioni con più di un finale alternativo:

```
\relative {
  \repeat volta 3 { c''4 d e f | }
  \alternative {
    { c2 e | }
    { f2 g | }
    { a2 g | }
  }
  c1
}
```



Nota: Se ci sono due o più finali alternativi, non ci deve essere niente tra la parentesi di chiusura di uno e quella di apertura di quello successivo all'interno di un blocco `\alternative`, altrimenti non si otterrà il numero atteso di finali.

Nota: Se si usa `\relative` dentro a un blocco `\repeat` senza istanziare esplicitamente il contesto `Voice`, appare un rigo in più (non desiderato). Vedi Sezione “Appare un rigo in più” in *Uso del Programma*.

Se una ripetizione che non ha finali alternativi inizia in mezzo a una misura, solitamente termina in un punto corrispondente nel mezzo di una misura successiva (così che tra le due estremità ci sia una misura completa). In questo caso i segni di ripetizione non sono delle “vere” e proprie stanghette, dunque né i controlli di battuta né i comandi `\partial` devono essere messi lì:

```
\relative {
  c'4 e g
  \repeat volta 4 {
    e4 |
    c2 e |
    g4 g g
  }
  g4 |
  a2 a |
  g1 |
}
```



Se una ripetizione senza finali alternativi inizia con una misura parziale, si applicano gli stessi principi dell'esempio precedente, a parte il fatto che è richiesto un comando `\partial` all'inizio della misura:

```
\partial 4
\repeat volta 4 {
  e'4 |
  c2 e |
  g4 g g
}
g4 |
a2 a |
g1 |
```



Si possono aggiungere delle legature di valore a un secondo finale:

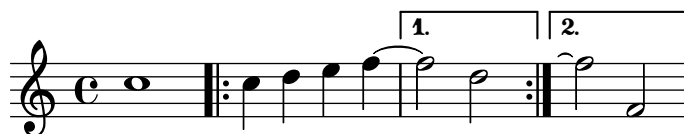
```
\relative {
  c' '1
```



```

\repeat volta 2 { c4 d e f~ }
\alternative {
  { f2 d }
  { f2\repeatTie f, }
}

```



Il comando `\inStaffSegno` può essere usato per generare una stanghetta composita che incorpora il simbolo di segno nella stanghetta di ripetizione appropriata se usato col comando `\repeat volta`. Il tipo corretto di stanghetta di ripetizione, ovvero inizio della ripetizione, fine della ripetizione e doppia ripetizione, viene selezionato automaticamente. Il corrispondente segno “D.S.” deve essere aggiunto manualmente.

Lontano da una ripetizione:

```

\relative {
  e'1
  \inStaffSegno
  f2 g a b
  c1_"D.S." \bar "|"
}

```



All'inizio di una ripetizione:

```

\relative {
  e'1
  \repeat volta 2 {
    \inStaffSegno % inizio ripetizione
    f2 g a b
  }
  c1_"D.S." \bar "|"
}

```

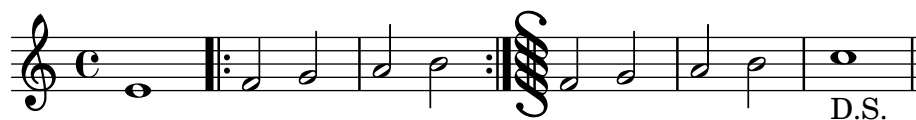


Alla fine di una ripetizione:

```

\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
    \inStaffSegno % fine ripetizione
  }
  f2 g a b
  c1_"D.S." \bar "|"
}

```



Tra due ripetizioni:

```
\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
  }
  \inStaffSegno % doppia ripetizione
  \repeat volta 2 {
    f2 g a b
  }
  c1_"D.S." \bar "|."
}
```



Si possono impostare simboli alternativi delle stanghette modificando nel contesto Score le proprietà `segnoBarType`, `startRepeatSegnoBarType`, `endRepeatSegnoBarType` o `doubleRepeatSegnoBarType` per il tipo di stanghetta richiesto. I tipi di stanghetta alternativi devono essere selezionati dai tipi predefiniti o dai tipi precedentemente definiti col comando `\defineBarLine` (vedi [Stanghette], pagina 105).

```
\defineBarLine ":|.S[" #'(":". "S[" """)
\defineBarLine "]" #'("]" "" """)
\relative {
  e'1
  \repeat volta 2 {
    f2 g a b
    \once \set Score.endRepeatSegnoBarType = ":|.S["
    \inStaffSegno
  }
  f2 g \bar "]" a b
  c1_"D.S." \bar "|."
}
```



Frammenti di codice selezionati

Accorciare le parentesi delle volte

Per impostazione predefinita, le parentesi delle volte si estendono per tutta l'alternativa, ma si possono accorciare impostando `voltaSpannerDuration`. Nell'esempio seguente, la parentesi dura una misura, che ha una durata di $3/4$.

```
\fixed c' {
  \time 3/4
  c4 c c
```

```

\repeat volta 5 {
  d4 d d
  \alternative {
    \volta 1,2,3,4 {
      \once \override Score.VoltaBracket.musical-length =
        \musicLength 2.
      e4 e e
      f4 f f
    }
    \volta 5 {
      g4 g g } } }
}

```



Volta brackets in multiple staves

By adding the `Volta_engraver` to the relevant staff, volte can be put over staves other than the topmost one in a score.

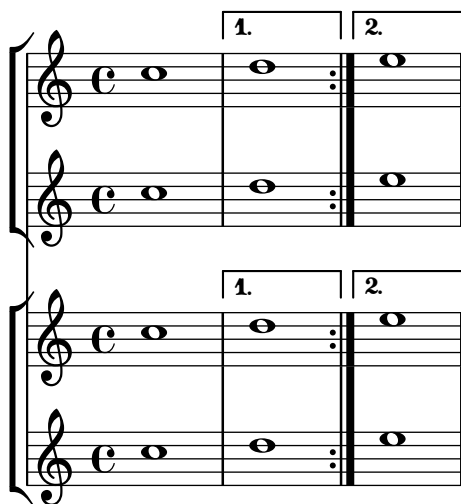
`\repeat` and related commands should be present in all staves.

```

voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
    \alternative {
      \volta 1 { d1 }
      \volta 2 { e1 }
    }
  }
}

<<
\new StaffGroup <<
  \new Staff \voltaMusic
  \new Staff \voltaMusic
>>
\new StaffGroup <<
  \new Staff \with { \consists "Volta_engraver" }
  \voltaMusic
  \new Staff \voltaMusic
>>
>>

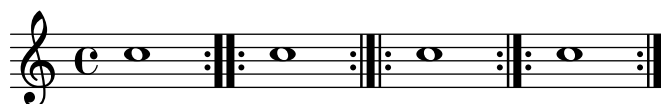
```



Impostare la doppia ripetizione predefinita per le volte

Esistono tre diversi stili di doppie ripetizioni per le volte, che si possono impostare con `doubleRepeatBarType`.

```
\relative c' {
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = "...:"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = "||:|:|"
  \repeat volta 2 { c1 }
  \set Score.doubleRepeatBarType = "||:|:|"
  \repeat volta 2 { c1 }
}
```



Numeri di battuta alternativi

Si possono impostare due metodi alternativi di numerazione della battuta, utili specialmente per le ripetizioni.

```
music = \relative c' {
  \repeat volta 3 {
    c4 d e f |
    \alternative {
      \volta 1 { c4 d e f | c2 d \break }
      \volta 2 { f4 g a b | f4 g a b | f2 a | \break }
      \volta 3 { c4 d e f | c2 d } } }
  c1 \bar "||."
}

{
  \textMark \markup \large "default"
  \music
}

{
```

```

\textMark \markup \large \typewriter "numbers"
\set Score.alternativeNumberingStyle = #'numbers
\music
}

{
  \textMark \markup \large \typewriter "numbers-with-letters"
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \music
}

\layout {
  \context {
    \Score
    \override TextMark.Y-offset = #5
  }
}

```

default

4

7

numbers

2

2

numbers-with-letters

2b



Vedi anche

Glossario Musicale: Sezione “ripetizione” in *Glossario Musicale*, Sezione “volta” in *Glossario Musicale*.

Guida alla notazione: [Stanghette], pagina 105, Sezione 5.1.4 [Modifica dei componenti aggiuntivi di un contesto], pagina 613, [Modifica di legature di valore e di portamento], pagina 666, [Gestione del tempo], pagina 128.

File installati: `ly/engraver-init.ly`.

Frammenti: Sezione “Repeats” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “VoltaBracket” in *Guida al Funzionamento Interno*, Sezione “VoltaRepeatedMusic” in *Guida al Funzionamento Interno*, Sezione “Unfolded-RepeatedMusic” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le legature di portamento che si estendono da un blocco `\repeat` verso un blocco `\alternative` funzioneranno solo nel primo finale alternativo. L’aspetto grafico di una legatura di portamento che continua negli altri finali alternativi può essere simulato con `\repeatTie` se la legatura si estende solo su una nota del blocco dell’alternativa, sebbene questo metodo non funzioni in `TabStaff`. Altri metodi che si possono adattare per indicare legature di portamento che continuano su varie note dei blocchi di alternativa, e che funzionano anche nei contesti `TabStaff`, sono presentati in [Modifica di legature di valore e di portamento], pagina 666.

Inoltre le legature di portamento non possono ricollegarsi dalla fine di un’alternativa all’inizio della ripetizione.

I glissandi che si estendono da un blocco `\repeat` in un blocco `\alternative` funzioneranno soltanto per il primo finale alternativo. L’aspetto grafico di un glissando che continua negli altri finali alternativi può essere indicato creando un glissando che inizia su una nota di abbellimento nascosta. Vedere ad esempio il frammento “Estendere i glissandi attraverso le ripetizioni” nei Frammenti Selezionati in [Glissando], pagina 149.

Se una ripetizione che inizia con una misura incompleta ha un blocco `\alternative` che contiene modifiche alla proprietà `measureLength`, l’uso di `\unfoldRepeats` causerà l’erroneo posizionamento delle stanghette e degli avvisi di controllo di battuta.

Una ripetizione annidata come la seguente

```
\repeat ...
\repeat ...
\alternative
```

è ambigua, perché non è chiaro a quale `\repeat` appartenga il blocco `\alternative`. Questa ambiguità si risolve facendo in modo che `\alternative` appartenga sempre al blocco `\repeat` interno. Per chiarezza, si consiglia di usare le parentesi in queste situazioni.

Indicazioni di ripetizione manuali

Nota: Questi metodi vengono usati solo per mostrare tipi di ripetizioni inusuali, e potrebbero causare un comportamento inaspettato. Nella maggior parte dei casi, le ripetizioni devono essere create col comando standard `\repeat` oppure stampando le stanghette opportune. Maggiori informazioni in [Stanghette], pagina 105.

La proprietà `repeatCommands` permette di controllare la formattazione delle ripetizioni. Il suo valore è una lista Scheme dei comandi di ripetizione.

`start-repeat`

Stampa una stanghetta `.|:.`

```
\relative {
  c''1
  \set Score.repeatCommands = #'(start-repeat)
  d4 e f g
  c1
}
```



Come vuole la pratica comune di incisione, i segni di ripetizione non vengono stampati all'inizio di un brano.

`end-repeat`

Stampa una stanghetta `:|:.`

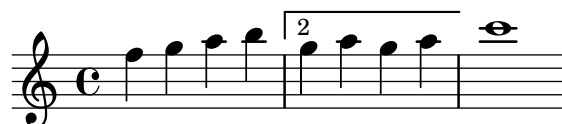
```
\relative {
  c''1
  d4 e f g
  \set Score.repeatCommands = #'(end-repeat)
  c1
}
```



`(volta numero) ... (volta #f)`

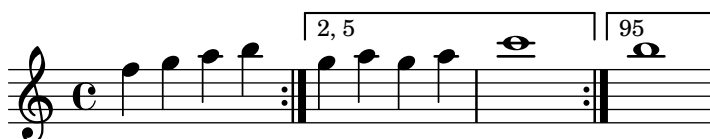
Crea una nuova volta col numero specificato. La parentesi della volta deve essere terminata esplicitamente, altrimenti non sarà stampata.

```
\relative {
  f''4 g a b
  \set Score.repeatCommands = #'((volta "2"))
  g4 a g a
  \set Score.repeatCommands = #'((volta #f))
  c1
}
```



Comandi di ripetizione multipli possono trovarsi nello stesso punto:

```
\relative {
  f' '4 g a b
  \set Score.repeatCommands = #'((volta "2, 5") end-repeat)
  g4 a g a
  c1
  \set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
  b1
  \set Score.repeatCommands = #'((volta #f))
}
```



Si può includere del testo nella parentesi della volta. Il testo può consistere di un numero, di più numeri o di un'indicazione testuale, si veda Sezione 1.8.2 [Formattazione del testo], pagina 260. Il modo più semplice per usare del testo è definirlo prima e poi includerlo nella lista Scheme,

```
voltaAdLib = \markup { \volta-number { 1. 2. 3... } \italic { ad lib. } }
\relative {
  c' '1
  \set Score.repeatCommands = #'((volta ,voltaAdLib) start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f)
                                (volta ,#{ \markup \volta-number "4." #})
                                end-repeat)

  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



Vedi anche

Guida alla notazione: [Stanghette], pagina 105, Sezione 1.8.2 [Formattazione del testo], pagina 260.

Frammenti: Sezione “Repeats” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “VoltaBracket” in *Guida al Funzionamento Interno*, Sezione “VoltaRepeatedMusic” in *Guida al Funzionamento Interno*.

Ripetizioni ricopiate

Col comando `unfold`, le ripetizioni possono servire a semplificare la scrittura di musica ripetitiva. La sintassi è

```
\repeat unfold numero-di-ripetizioni espressione-musicale
```

dove *espressione-musicale* è la musica e *numero-di-ripetizioni* è il numero di volte per cui è ripetuta *espressione-musicale*.

```
\relative {
  \repeat unfold 2 { c' '4 d e f }
```



```
c1
}
```



In alcuni casi, specialmente in un contesto `\relative`, la funzione `\repeat unfold` non equivale a riscrivere l'espressione musicale più volte. Ad esempio

```
\repeat unfold 2 { a'4 b c }
```

non equivale a

```
a'4 b c | a'4 b c
```

Le ripetizioni dispiegate (unfold) possono avere dei finali alternativi.

```
\relative {
  \repeat unfold 2 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
  }
  c1
}
```



Se il numero di ripetizioni è maggiore del numero di finali alternativi, la prima alternativa viene applicata più volte, finché le alternative rimaste non esauriscono il numero totale delle ripetizioni.

```
\relative {
  \repeat unfold 4 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
    { e2 d }
  }
  c1
}
```



Se il numero di finali alternativi è maggiore del numero di ripetizioni, solo le prime alternative vengono applicate. Le alternative rimanenti saranno ignorate e non verranno stampate.

```
\relative {
  \repeat unfold 2 { c''4 d e f }
  \alternative {
    { c2 g' }
    { c,2 b }
    { e2 d }
  }
}
```

```
c1
}
```



È anche possibile annidare molteplici funzioni `unfold` (con o senza finali alternativi).

```
\relative {
  \repeat unfold 2 {
    \repeat unfold 2 { c''4 d e f }
    \alternative {
      { c2 g' }
      { c,2 b }
    }
  }
}
c1
}
```



Gli accordi si ripetono col simbolo di ripetizione dell'accordo `q`. Vedi [Ripetizione di un accordo], pagina 180.

Nota: Se si usa `\relative` dentro a un blocco `\repeat` senza istanziare esplicitamente il contesto `Voice`, appare un rigo in più (non desiderato). Vedi Sezione “Appare un rigo in più” in *Uso del Programma*.

Vedi anche

Guida alla notazione: [Ripetizione di un accordo], pagina 180.

Frammenti: Sezione “Repeats” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “UnfoldedRepeatedMusic” in *Guida al Funzionamento Interno*.

1.4.2 Ripetizioni brevi

Questa sezione tratta il modo in cui inserire brevi ripetizioni. Le ripetizioni brevi possono avere due forme: segni di tratto obliquo o percentuale per rappresentare le ripetizioni di una singola nota, di una singola misura o di due misure; tremolo negli altri casi.

Ripetizioni con percentuale

Brevi sezioni ripetute vengono stampate la prima volta e le ripetizioni vengono sostituite da un apposito segno.

La sintassi è

```
\repeat percent numero espressione-musicale
```

dove *espressione-musicale* è l'espressione musicale da ripetere.

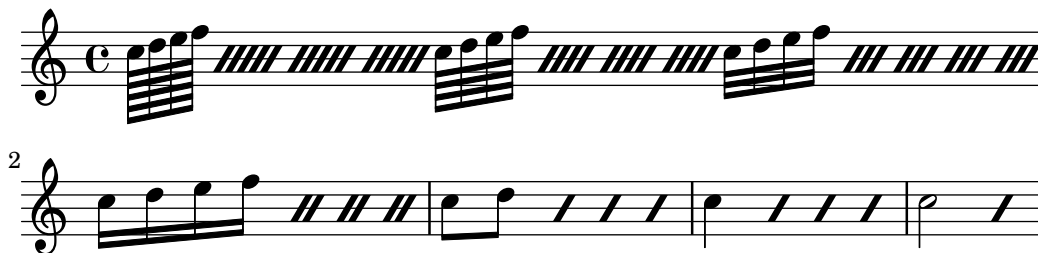
Fraseggi più brevi di una misura vengono sostituiti dal tratto obliquo.

```
\relative c'' {
```

```

\repeat percent 4 { c128 d e f }
\repeat percent 4 { c64 d e f }
\repeat percent 5 { c32 d e f }
\repeat percent 4 { c16 d e f }
\repeat percent 4 { c8 d }
\repeat percent 4 { c4 }
\repeat percent 2 { c2 }
}

```



Fraseggi di una o due misure vengono sostituiti da simboli simili alla percentuale.

```

\relative c' {
  \repeat percent 2 { c4 d e f }
  \repeat percent 2 { c2 d }
  \repeat percent 2 { c1 }
}

```



```

\relative {
  \repeat percent 3 { c''4 d e f | c2 g' }
}

```

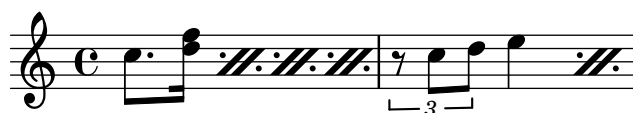


Fraseggi più brevi di una misura ma con durate miste adottano un simbolo di doppia percentuale.

```

\relative {
  \repeat percent 4 { c''8. <d f>16 }
  \repeat percent 2 { \tuplet 3/2 { r8 c d } e4 }
}

```



Frammenti di codice selezionati

Contatore della ripetizione con segno percentuale

Le ripetizioni di misura che hanno più di due ripetizioni possono avere un contatore se si cambia la proprietà opportuna, come mostra questo esempio:

```

\relative c' {

```

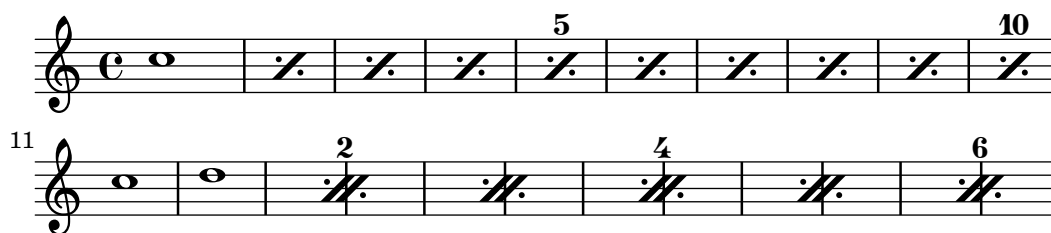
```
\set countPercentRepeats = ##t
\repeat percent 4 { c1 }
}
```



Visibilità del conto della ripetizione con segno percentuale

I contatori della ripetizione con segno percentuale possono essere mostrati a intervalli regolari impostando la proprietà di contesto `repeatCountVisibility`.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Ripetizioni con segni di percentuale isolati

Si possono stampare anche segni di percentuale isolati.

```
makePercent =
#(define-music-function (note) (ly:music?)
  "Make a percent repeat the same length as NOTE."
  (make-music 'PercentEvent
    'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```



Vedi anche

Glossario Musicale: Sezione “percent repeat” in *Glossario Musicale*, Sezione “simile” in *Glossario Musicale*.

Frammenti: Sezione “Repeats” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “RepeatSlash” in *Guida al Funzionamento Interno*, Sezione “RepeatSlashEvent” in *Guida al Funzionamento Interno*, Sezione “DoubleRepeatSlash” in *Guida al Funzionamento Interno*, Sezione “PercentRepeat” in *Guida al Funzionamento Interno*, Sezione “PercentRepeatCounter” in *Guida al Funzionamento Interno*, Sezione “PercentRepeatedMusic” in *Guida al Funzionamento Interno*, Sezione “Percent-repeat-engraver” in *Guida al Funzionamento Interno*, Sezione “DoublePercentEvent”

in *Guida al Funzionamento Interno*, Sezione “DoublePercentRepeat” in *Guida al Funzionamento Interno*, Sezione “DoublePercentRepeatCounter” in *Guida al Funzionamento Interno*, Sezione “Double_percent_repeat_engraver” in *Guida al Funzionamento Interno*, Sezione “Slash_repeat_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le ripetizioni con percentuale non contengono nient’altro che il segno di percentuale; in particolare, i cambi di tempo non saranno ripetuti.

```
\repeat percent 3 { \time 5/4 c2. 2 \time 4/4 2 2 }
```



Qualsiasi cambio di tempo o comando `\partial` devono trovarsi in passaggi paralleli *esterni* a qualsiasi ripetizione con percentuale, per esempio su una traccia di tempo separata.

```
<<
\repeat percent 3 { c2. 2 2 2 }
\repeat unfold 3 { \time 5/4 s4*5 \time 4/4 s1 }
>>
```

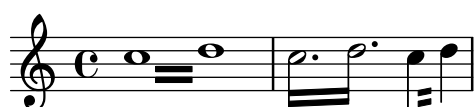


Ripetizioni con tremolo

I tremoli possono avere due forme: alternanza tra due note, o due accordi, e rapida ripetizione di una singola nota o accordo. I tremoli costituiti da un’alternanza si indicano con delle travature che collegano le note o gli accordi che si alternano, mentre i tremoli che consistono in una rapida ripetizione di una nota singola si indicano aggiungendo delle travature o dei tratti di suddivisione obliqui alla singola nota.

Per inserire i segni del tremolo tra le note, si usa `\repeat` con lo stile tremolo:

```
\relative c' {
  \repeat tremolo 8 { c16 d }
  \repeat tremolo 6 { c16 d }
  \repeat tremolo 2 { c16 d }
}
```



La sintassi di `\repeat tremolo` prevede specificamente che all’interno delle parentesi siano indicate due note, e che il numero di ripetizioni corrisponda a un valore espresso in durate di note normali o puntate. Dunque `\repeat tremolo 7` è valido e produce una nota doppiamente puntata, mentre `\repeat tremolo 9` non è valido.

La durata del tremolo equivale alla durata dell'espressione musicale tra parentesi moltiplicata per il numero di ripetizioni: `\repeat tremolo 8 { c'16 d'16 }` corrisponde a un tremolo di una semibreve, rappresentata come due semibrevi unite dalle travature del tremolo.

Ci sono due modi di inserire dei segni di tremolo su una singola nota. Anche in questo caso si usa la sintassi `\repeat tremolo`, ma la nota non deve essere racchiusa tra parentesi:

```
\repeat tremolo 4 c'16
```



Si può ottenere lo stesso output aggiungendo `:N` dopo la nota, dove N indica la durata della suddivisione (deve essere almeno 8). Se N è 8, viene aggiunta una travatura al gambo della nota. Se N è omesso, viene usato l'ultimo valore:

```
\relative {
  c'12:8 c:32
  c: c:
}
```

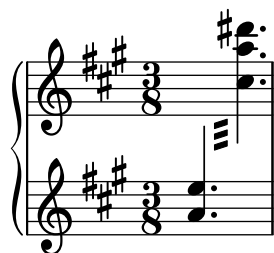


Frammenti di codice selezionati

Tremoli attraverso i righi

Dato che `\repeat tremolo` si aspetta esattamente due argomenti musicali per i tremoli di accordi, la nota o l'accordo che cambiano rigo in un tremolo che attraversa i righi devono essere posti tra parentesi graffe insieme al comando `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'1 {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'1 {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
}>>
```



Vedi anche

Frammenti: Sezione “Repeats” in *Frammenti di codice*.

1.5 Note simultanee

La polifonia in musica si riferisce alla coesistenza simultanea di più di una voce in un brano musicale. In LilyPond la polifonia si riferisce alla coesistenza di più voci sullo stesso rigo.

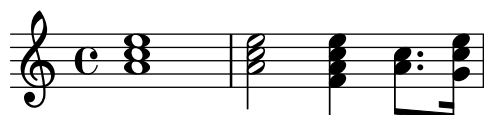
1.5.1 Una voce

In questa sezione vengono spiegate le note simultanee che fanno parte di un'unica voce.

Note in un accordo

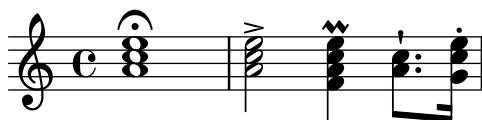
Un accordo si forma racchiudendo una serie di altezze tra < e > e può essere seguito da una durata, come accade per le semplici note.

```
\relative {
  <a' c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
}
```



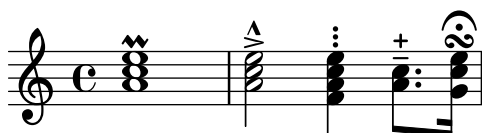
Proprio come per le note, si possono specificare le articolazioni da riferire all'accordo.

```
\relative {
  <a' c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^! <g c e>16-.
}
```



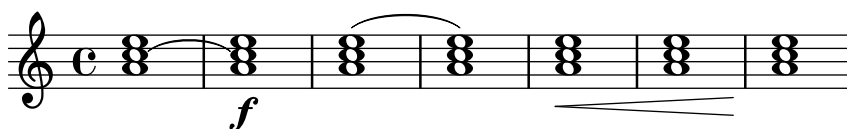
Si possono specificare abbellimenti e articolazioni per ogni nota che fa parte dell'accordo.

```
\relative {
  <a' c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4
  <a++ c-->8. <g\fermata c e\turn>16
}
```



Tuttavia, alcuni elementi della notazione, come le dinamiche e le forcelle, devono essere attaccate all'accordo invece che alle sue singole note, altrimenti non appariranno. Altri elementi della notazione, come le ditekgiature e le legature di portamento, saranno posizionate in modo nettamente diverso se attaccate alle note di un accordo invece che a un accordo intero o a singole note.

```
\relative {
  <a'\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>)
  <a c e>\< <a c e> <a c e>\!
}
```



Un accordo si comporta semplicemente come un contenitore di note, articolazioni e altri elementi. Di conseguenza, un accordo privo di note non ha una durata. Qualsiasi articolazione attaccata a un accordo vuoto si troverà nel momento musicale della nota o accordo seguenti e si combinerà con questi (possibilità più complesse di combinazione sono spiegate in [Espressioni simultanee], pagina 182):

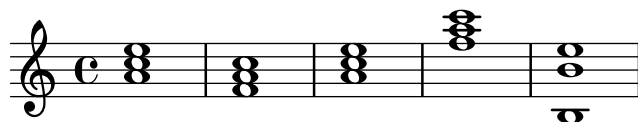
```
\relative {
  \grace { g'8( a b }
  <> ) \p \< -. -\markup \italic "sempre staccato"
  \repeat unfold 4 { c4 e } c1\f
}
```



Si può usare la modalità relativa per indicare l'altezza degli accordi. La prima nota di ogni accordo è sempre relativa alla prima nota dell'accordo che lo precede oppure, se non c'è un

accordo precedente, è relativa all'altezza dell'ultima nota che precede l'accordo. Le altezze di tutte le altre note dell'accordo sono relative alla nota che le precede *all'interno dell'accordo*.

```
\relative {
  <a' c e>1 <f a c> <a c e> <f' a c> <b, e b,>
}
```



Maggiori informazioni sugli accordi si trovano in Sezione 2.7 [Notazione per accordi], pagina 431.

Vedi anche

Glossario Musicale: Sezione “accordo” in *Glossario Musicale*.

Manuale d'apprendimento: Sezione “Combinare le note negli accordi” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 2.7 [Notazione per accordi], pagina 431, [Articolazioni e abbellimenti], pagina 129, [Ottava relativa], pagina 2, Sezione 1.5.2 [Più voci], pagina 184.

Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

Problemi noti e avvertimenti

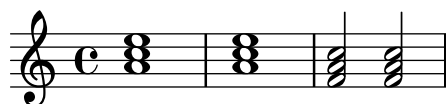
Gli accordi che contengono più di due altezze in uno spazio del rigo, come ad esempio ‘<e f! fis!>’, presentano le teste di tali note sovrapposte. A seconda della situazione, si può migliorare l'aspetto con

- l'uso temporaneo di Sezione 1.5.2 [Più voci], pagina 184, ‘<< f! \\
<e fis!> >>’,
- la trascrizione enarmonica di una o più altezze, ‘<e f ges>’, oppure
- i [Cluster], pagina 183.

Ripetizione di un accordo

Per inserire la musica più rapidamente, si può usare una scorciatoia che ripete l'accordo precedente. Il simbolo di ripetizione dell'accordo è q:

```
\relative {
  <a' c e>1 q <f a c>2 q
}
```



Come nel caso dei normali accordi, il simbolo di ripetizione dell'accordo si può usare con le durate, le articolazioni, i testi a margine, le legature di portamento, le travature, etc. dato che solo le altezze dell'accordo precedente vengono duplicate.

```
\relative {
  <a' c e>1 \p^"text" q2 \< ( q8) [-! q8.] \! q16-1-2-3 q8 \prall
}
```



Il simbolo di ripetizione dell'accordo ricorda sempre l'ultimo accordo inserito, quindi è possibile inserire l'accordo più recente anche se nel frattempo sono state inserite altre note (senza accordi) o pause.

```
\relative {
  <a' c e>1 c'4 q2 r8 q8 |
  q2 c, |
}
```



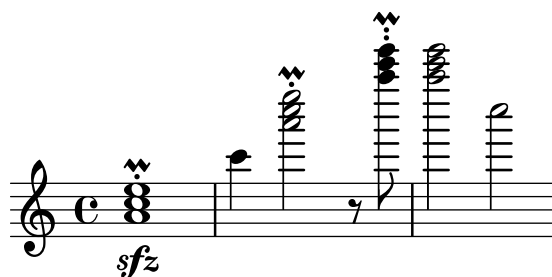
Tuttavia questo simbolo non conserva le dinamiche, le articolazioni o gli abbellimenti dell'accordo precedente.

```
\relative {
  <a'-. c\prall e>1\sffz c'4 q2 r8 q8 |
  q2 c, |
}
```



Per far sì che alcuni elementi siano conservati, si può invocare esplicitamente la funzione `\chordRepeats` con un'ulteriore argomento che indica una lista di *tipi di evento* da mantenere, a meno che eventi di quel tipo non siano già presenti nell'accordo q stesso.

```
\relative {
  \chordRepeats #'(articulation-event)
  { <a'-. c\prall e>1\sffz c'4 q2 r8 q8-. } |
  q2 c, |
}
```

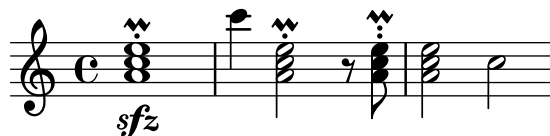


In questo esempio l'uso di `\chordRepeats` all'interno di un blocco `\relative` produce risultati indesiderati: gli eventi di un accordo, una volta espansi, non si distinguono più per essere stati inseriti come accordi normali, quindi `\relative` assegna un'ottava basata sul contesto corrente.

Dato che `\relative` annidati non si influenzano l'un l'altro, si può usare un altro `\relative` dentro `\chordRepeats` per stabilire le relazioni di ottava prima di espandere gli accordi ripetuti. In questo caso l'intero contenuto del `\relative` più interno non influenza quello esterno; ecco perché in questo esempio la nota finale è stata specificata con un'ottava diversa.

```
\new Voice
\relative c'' {
  \chordRepeats #'(articulation-event)
  \relative
```

```
{ <a'- . c\prall e>1\sفز c'4 q2 r8 q8- . } |
q2 c |
}
```



Le interazioni con `\relative` si verificano solo con chiamate esplicite di `\chordRepeats`: l'espansione implicita all'inizio della creazione della partitura viene fatta in un momento in cui tutti i `\relative` sono stati già elaborati.

Vedi anche

Guida alla notazione: Sezione 2.7 [Notazione per accordi], pagina 431, [Articolazioni e abbellimenti], pagina 129.

File installati: `ly/chord-repetition-init.ly`.

Espressioni simultanee

Una o più espressioni musicali racchiuse tra due coppie di parentesi uncinate sono considerate simultanee. Se la prima espressione inizia con una nota singola o se l'intera espressione simultanea appare esplicitamente all'interno di una voce, sarà posta in un solo rigo; altrimenti gli elementi dell'espressione simultanea saranno messi in righe separate.

Gli esempi seguenti mostrano espressioni simultanee su un rigo:

```
\new Voice { % voce singola esplicita
  << \relative { a'4 b g2 }
    \relative { d'4 g c,2 } >>
}
```



```
\relative {
  % prima nota singola
  a' << \relative { a'4 b g2 }
    \relative { d'4 g c, } >>
}
```



Questo può essere utile se le sezioni simultanee hanno durate identiche, ma i tentativi di collegare note con durate diverse allo stesso gambo causerà degli errori. Le note, le articolazioni e le modifiche delle proprietà in una *singola* voce ('Voice') sono raccolte e create secondo l'ordine della musica:

```
\relative {
  <a' c>4- . <>- . << c a >> << { c- . <c a> } { a s- . } >>
}
```



Per poter inserire gambi o travature multiple e variare le durate o altre proprietà di note riferibili allo stesso momento musicale, occorre usare più voci.

L'esempio seguente mostra come le espressioni simultanee possano generare implicitamente righi multipli:

```
% nessuna singola nota precede l'espressione simultanea
<< \relative { a'4 b g2 }
    \relative { d'4 g2 c,4 } >>
```



In questo caso le durate diverse non causano problemi perché sono interpretate in voci diverse.

Problemi noti e avvertimenti

Se le note appartenenti a due o più voci, senza che sia stato specificato uno spostamento, hanno i gambi nella stessa direzione, il messaggio

attenzione: questa voce ha bisogno di un'impostazione `\voiceXx` o `\shiftXx`

apparirà durante la compilazione del file. Si può evitare con:

```
\override NoteColumn.ignore-collision = ##t
```

Tuttavia, questo comando non si limita a eliminare l'avvertimento, ma impedisce qualsiasi risoluzione delle collisioni, e potrebbe comportare altri effetti indesiderati (vedi anche i *Problemi noti* in [Risoluzione delle collisioni], pagina 188).

Cluster

Un cluster prescrive l'esecuzione simultanea di tutti i suoni compresi in un determinato intervallo. Può essere rappresentato come un involucro che contiene le note che ne fanno parte. Si inserisce applicando la funzione `\makeClusters` a una sequenza di accordi, ad esempio:

```
\relative \makeClusters { <g' b>2 <c g'> }
```



Si possono inserire insieme sullo stesso rigo le normali note e i cluster, anche contemporaneamente. In tal caso non viene fatto alcun tentativo di evitare automaticamente collisioni tra le note normali e i cluster.

Vedi anche

Glossario Musicale: Sezione “cluster” in *Glossario Musicale*.

Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “ClusterSpanner” in *Guida al Funzionamento Interno*, Sezione “ClusterSpannerBeacon” in *Guida al Funzionamento Interno*, Sezione “Cluster-spanner_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

I cluster hanno un buon aspetto solo se durano almeno per due accordi; altrimenti appaiono troppo stretti.

I cluster non hanno un gambo e non possono indicare delle durate da soli, ma la lunghezza del cluster è determinata dalle durate degli accordi che lo definiscono. Più cluster distinti devono essere separati da una pausa.

I cluster non generano output MIDI.

1.5.2 Più voci

Questa sezione presenta le note simultanee in più voci o più righi.

Polifonia su un solo rigo

Istanziare esplicitamente le voci

La struttura di base necessaria per ottenere più voci indipendenti in un solo rigo è illustrata nell'esempio seguente:

```
\new Staff <<
  \new Voice = "prima"
    \relative { \voiceOne r8 r16 g'' e8. f16 g8[ c,] f e16 d }
  \new Voice= "seconda"
    \relative { \voiceTwo d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>
```



Le voci sono istanziate esplicitamente e vengono contrassegnate da dei nomi. I comandi `\voiceOne ... \voiceFour` impostano le voci in modo che la prima e terza voce abbiano i gambi in su, la seconda e la quarta voce i gambi in giù, le teste di nota della terza e quarta voce siano spostate orizzontalmente e le pause in ciascuna voce siano spostate automaticamente per evitare collisioni. Il comando `\oneVoice` ripristina tutte le impostazioni della voce alle direzioni neutrali predefinite.

Passaggi polifonici temporanei

Un passaggio polifonico temporaneo si può creare col seguente costrutto:

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

In questo esempio la prima espressione all'interno di un passaggio polifonico temporaneo è posta nel contesto `Voice` che era in uso immediatamente prima del passaggio polifonico e quello stesso contesto `Voice` continua dopo la sezione temporanea. Le altre espressioni comprese nelle parentesi uncinate vengono assegnate a voci temporanee distinte. Questo permette di assegnare il testo a una voce che continua prima, durante e dopo una sezione polifonica:

```
\relative <<
  \new Voice = "melody" {
    a'4
    <<
    {
      \voiceOne
```

```

      g f
    }
    \new Voice {
      \voiceTwo
      d2
    }
  >>
  \oneVoice
  e4
}
\new Lyrics \lyricsto "melody" {
This is my song.
}
>>

```



I comandi `\voiceOne` e `\voiceTwo` sono necessari per definire le impostazioni di ogni voce.

Il costrutto con la doppia barra inversa (backslash)

Il costrutto `<< {...} \ \ {...} >>`, in cui due (o più) espressioni sono separate da due barre inverse (backslash), si comporta diversamente dal costrutto simile privo delle due barre: *tutte* le espressioni in questo costrutto vengono assegnate a nuovi contesti Voice. Questi nuovi contesti Voice vengono creati implicitamente e ad essi vengono assegnati dei nomi prestabiliti "1", "2", etc.

Il primo esempio potrebbe essere riscritto nel modo seguente:

```

<<
  \relative { r8 r16 g'' e8. f16 g8[ c,] f e16 d }
  \
  \relative { d''16 c d8~ 16 b c8~ 16 b c8~ 16 b8. }
>>

```



Questa sintassi si usa quando non importa che le voci temporanee siano create e poi eliminate. A queste voci create implicitamente vengono assegnate le impostazioni equivalenti all'uso dei comandi `\voiceOne ... \voiceFour`, nell'ordine in cui appaiono nell'input.

Nell'esempio seguente, la voce intermedia ha i gambi in su, dunque viene inserita in terza posizione in modo che diventi la terza voce, che ha i gambi in su. Si usano le pause spaziatrici per evitare il raddoppio delle pause ordinarie.

```

<<
  \relative { r8 g'' g g f16 ees f8 d }
  \
  \relative { ees'8 r ees r d r d r }
  \
  \relative { d''8 s c s bes s a s }
>>

```

>>



In tutti i brani, a eccezione di quelli più semplici, è consigliabile creare contesti Voice espliciti come è spiegato in Sezione “Contesti e incisori” in *Manuale di Apprendimento* e Sezione “Definire esplicitamente le voci” in *Manuale di Apprendimento*.

Ordine delle voci

Quando si inseriscono più voci nel file di input, conviene usare il seguente ordine:

Voce 1: la più alta
 Voce 2: la più bassa
 Voce 3: la seconda più alta
 Voce 4: la seconda più bassa
 Voce 5: la terza più alta
 Voce 6: la terza più bassa
 etc.

Sebbene possa sembrare controintuitivo, ciò semplifica il processo automatico di formattazione. Si noti che le voci con numero dispari hanno i gambi in su, quelle con numero pari hanno i gambi in giù:

```
\new Staff <<
  \time 2/4
  { f''2 } % 1: la più alta
  \\
  { c'2 } % 2: la più bassa
  \\
  { d''2 } % 3: seconda più alta
  \\
  { e'2 } % 4: seconda più bassa
  \\
  { b'2 } % 5: terza più alta
  \\
  { g'2 } % 6: terza più bassa
>>
```



Quando si desidera inserire le voci in un ordine diverso, conviene usare il comando `\voices`:

```
\new Staff \voices 1,3,5,6,4,2 <<
  \time 2/4
  { f''2 } % 1: la più alta
  \\
  { d''2 } % 3: seconda più alta
  \\
  { b'2 } % 5: terza più alta
```

```

\\
{ g'2 } % 6: terza più bassa
\\
{ e'2 } % 4: seconda più bassa
\\
{ c'2 } % 2: la più bassa
>>

```



Nota: Il testo e gli estensori (come le legature di portamento e di valore, le forcelle, etc.) non possono ‘attraversare’ le voci.

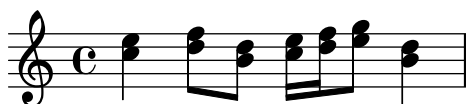
Durate identiche

Nel caso speciale in cui si desideri inserire sezioni musicali parallele con il medesimo ritmo, si possono combinare in un unico contesto Voice, formando dunque degli accordi. Per farlo, vanno racchiuse in un semplice costrutto musicale simultaneo all’interno di una voce esplicita:

```

\new Voice <<
  \relative { e''4 f8 d e16 f g8 d4 }
  \relative { c''4 d8 b c16 d e8 b4 }
>>

```



Questo metodo produce strane travature e avvertimenti se le sezioni musicali non hanno lo stesso ritmo.

Comandi predefiniti

\voiceOne, \voiceTwo, \voiceThree, \voiceFour, \oneVoice.

Vedi anche

Manuale d’apprendimento: Sezione “Le voci contengono la musica” in *Manuale di Apprendimento*, Sezione “Definire esplicitamente le voci” in *Manuale di Apprendimento*.

Guida alla notazione: [Righi delle percussioni], pagina 414, [Pause invisibili], pagina 65, [Gambi], pagina 244.

Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

Stili di voce

Si possono assegnare colori e forme diverse a ciascuna voce per facilitarne l’identificazione:

```

<<
  \relative { \voiceOneStyle d''4 c2 b4 }
  \\
  \relative { \voiceTwoStyle e'2 e }
  \\
  \relative { \voiceThreeStyle b2. c4 }

```



```

\\
\relative { \voiceFourStyle g'2 g }
>>

```



Il comando `\voiceNeutralStyle` permette di ripristinare l'aspetto predefinito.

Comandi predefiniti

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`,
`\voiceNeutralStyle`.

Vedi anche

Manuale d'apprendimento: Sezione “Sento le Voci” in *Manuale di Apprendimento*, Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

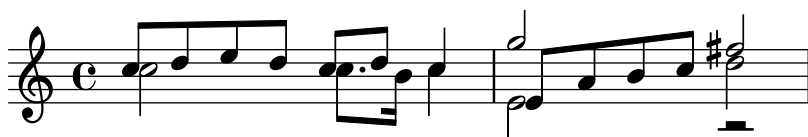
Risoluzione delle collisioni

Le teste di note che si trovano in voci diverse ma hanno stessa altezza, stessa testa e direzione del gambo opposta vengono unite automaticamente; invece, le note che hanno la stessa testa o la stessa direzione del gambo non vengono unite. Le pause opposte a un gambo in una voce diversa vengono spostate verticalmente. L'esempio seguente mostra tre diverse circostanze, sul primo e terzo movimento della prima battuta e sul primo movimento della seconda battuta, in cui l'unione automatica delle teste di nota non funziona.

```

<<
\relative {
  c'8 d e d c d c4
  g'2 fis
} \\
\relative {
  c'2 c8. b16 c4
  e,2 r
} \\
\relative {
  \oneVoice
  s1
  e'8 a b c d2
}
>>

```



Note con teste diverse possono essere unite come è mostrato sotto. In questo esempio le teste delle note nel primo battito della prima battuta sono unite:

```

<<
\relative {
  \mergeDifferentlyHeadedOn

```

```

        c''8 d e d c d c4
        g'2 fis
    } \\\
    \relative {
        c''2 c8. b16 c4
        e,2 r
    } \\\
    \relative {
        \oneVoice
        s1
        e'8 a b c d2
    }
>>

```



Le minime e le semiminime, invece, non sono unite, perché sarebbe difficile distinguerle.

Anche le teste di note con diversi punti, come nel terzo movimento della prima battuta, possono essere unite:

```
<<
\relative {
  \mergeDifferentlyHeadedOn
  \mergeDifferentlyDottedOn
  c' '8 d e d c d c4
  g'2 fis
} \\\
\relative {
  c' '2 c8. b16 c4
  e,2 r
} \\\
\relative {
  \oneVoice
  s1
  e'8 a b c d2
}
>>
```



La minima e la croma all’inizio della seconda misura sono unite per errore, perché l’unione automatica non riesce a completare correttamente l’unione quando tre o più note sono allineate sulla stessa colonna di note: in questo caso la testa di nota unita non è corretta. Per far sì che l’unione selezioni la testa di nota corretta, occorre applicare il comando `\shift` alla nota che non deve essere unita. In questo esempio si usa `\shiftOn` per spostare il Sol superiore (*g*) fuori dalla colonna e di conseguenza `\mergeDifferentlyHeadedOn` funziona correttamente.

```
<<
  \relative {
```

```

\mergeDifferentlyHeadedOn
\mergeDifferentlyDottedOn
c''8 d e d c d c4
\shiftOn
g'2 fis
} \
\relative {
  c''2 c8. b16 c4
  e,2 r
} \
\relative {
  \oneVoice
  s1
  e'8 a b c d2
}
>>

```



Il comando `\shiftOn` permette (senza forzare) lo spostamento delle note in una voce. Quando si applica `\shiftOn` a una voce, una nota o accordo in quella voce vengono spostati solo se il suo gambo altrimenti entrerebbe in collisione col gambo di un'altra voce, e solo se i gambi che collidono puntano nella stessa direzione. Il comando `\shiftOff` impedisce che si verifichi questo tipo di spostamento.

Per impostazione predefinita, le voci più esterne (solitamente la prima e la seconda voce) hanno specificato `\shiftOff`, mentre le voci più interne (terza e successive) hanno specificato `\shiftOn`. Quando si applica uno spostamento, le voci con i gambi in su (voci dispari) vengono spostate a destra, e le voci con i gambi in giù (voci pari) vengono spostate a sinistra.

Ecco un esempio che aiuta a visualizzare come un'espressione simultanea abbreviata viene espansa internamente.

Nota: Attenzione: con tre o più voci, l'ordine verticale delle voci nel file di input non deve essere lo stesso dell'ordine verticale delle voci del rigo!

```

\new Staff \relative {
  %% inserimento abbreviato
  <<
    { f''2 } % 1: più alta
    \
    { g,2 } % 2: più bassa
    \
    { d'2 } % 3: più alta centrale
    \
    { b2 } % 4: più bassa centrale
  >>
  %% espansione interna dell'input precedente
  <<
    \new Voice = "1" { \voiceOne \shiftOff f'2 }

```

```

\new Voice = "2" { \voiceTwo \shiftOff g,2 }
\new Voice = "3" { \voiceThree \shiftOn d'2 } % sposta a destra
\new Voice = "4" { \voiceFour \shiftOn b2 } % sposta a sinistra
>>
}

```



Due ulteriori comandi, `\shiftOnn` e `\shiftOnnn`, mettono a disposizione altri livelli di spostamento che possono essere specificati in modo temporaneo per risolvere delle collisioni in situazioni complesse – vedi Sezione “Esempio musicale” in *Manuale di Apprendimento*.

Le note vengono unite solo se presentano opposta direzione dei gambi (come accade, ad esempio, nella prima o seconda voce o quando i gambi sono impostati esplicitamente in direzioni opposte).

Comandi predefiniti

`\mergeDifferentlyDottedOn`, `\mergeDifferentlyDottedOff`, `\mergeDifferentlyHeadedOn`, `\mergeDifferentlyHeadedOff`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`.

Frammenti di codice selezionati

Voci ulteriori per evitare le collisioni

In alcuni casi di musica polifonica complessa sono necessarie delle voci ulteriori per evitare le collisioni tra note. Se servono più di quattro voci parallele, si possono aggiungere altre voci definendo una variabile con la funzione Scheme function `context-spec-music`.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)
```

```

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
    }
  >>
}

```

```

    a4 s2
  }
  \new Voice {
    \voiceFive
    s2
    g4 g2
    f4 f2
  }
  >>
}
```



Spostare le note puntate in polifonia

Quando una nota puntata della voce più alta viene spostata per evitare una collisione con una nota di un'altra voce, il comportamento predefinito è spostare la nota più alta a destra. Tale comportamento può essere modificato tramite la proprietà `prefer-dotted-right` di `NoteCollision`.

```

\new Staff \relative c' <<
{
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##f
  f2. f4
  \override Staff.NoteCollision.prefer-dotted-right = ##t
  f2. f4
}
\\
{ e4 e e e e e e e e e e }
>>
```



Forzare lo spostamento orizzontale delle note

Quando il motore tipografico non riesce a risolvere una situazione, si può usare la sintassi che sovrascrive le decisioni tipografiche. L'unità di misura usata è lo spazio del rigo.

```

\relative c' <<
{
  <d g>2 <d g>
}
\\
{
  <b f'>2
  \once \override NoteColumn.force-hshift = 1.7
  <b f'>2
}
>>
```



Vedi anche

Glossario Musicale: Sezione “polifonia” in *Glossario Musicale*.

Manuale d’apprendimento: Sezione “Note simultanee” in *Manuale di Apprendimento*, Sezione “Le voci contengono la musica” in *Manuale di Apprendimento*, Sezione “Esempio musicale” in *Manuale di Apprendimento*.

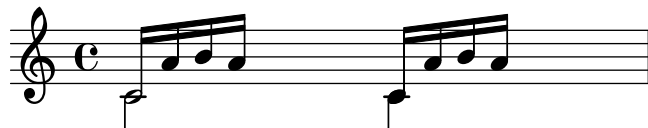
Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “NoteColumn” in *Guida al Funzionamento Interno*, Sezione “NoteCollision” in *Guida al Funzionamento Interno*, Sezione “RestCollision” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Se si usa `\override NoteColumn.ignore-collision = ##t`, le note con teste diverse che si trovano in voci diverse saranno unite in modo non corretto.

```
\mergeDifferentlyHeadedOn
<< \relative { c'16 a' b a } \ \relative { c'2 } >>
\override NoteColumn.ignore-collision = ##t
<< \relative { c'16 a' b a } \ \relative { c'2 } >>
```



Accorpare le pause

In caso di voci multiple, di solito si accorpano le pause che capitano in entrambe. Per farlo si usa l’incisore `Merge_rests_engraver`.

```
voiceA = \relative { d''4 r d2 | R1 | }
voiceB = \relative { fis'4 r g2 | R1 | }
\score {
  <<
    \new Staff \with {
      instrumentName = "non accorpato"
    }
    <<
      \new Voice { \voiceOne \voiceA }
      \new Voice { \voiceTwo \voiceB }
    >>
  \new Staff \with {
    instrumentName = "accorpato"
    \consists Merge_rests_engraver
  }
  <<
    \new Voice { \voiceOne \voiceA }
    \new Voice { \voiceTwo \voiceB }
  >>
  >>
```

}



Impostando la proprietà di contesto `suspendRestMerging` su `##t` è possibile disattivare temporaneamente l'accorpamento delle pause.

Combinazione automatica delle parti

La combinazione automatica delle parti si usa per combinare in un unico rigo due parti musicali separate. Ciò è utile soprattutto quando si scrivono partiture orchestrali. Viene stampata una sola voce se le due parti musicali sono identiche, ma nei punti in cui sono diverse viene aggiunta una seconda voce. Le direzioni dei gambi sono impostate in su e in giù in base alla voce di appartenenza, mentre le sezioni solistiche e *a due* sono a loro volta identificate e contrassegnate.

La sintassi per la combinazione automatica delle parti è:

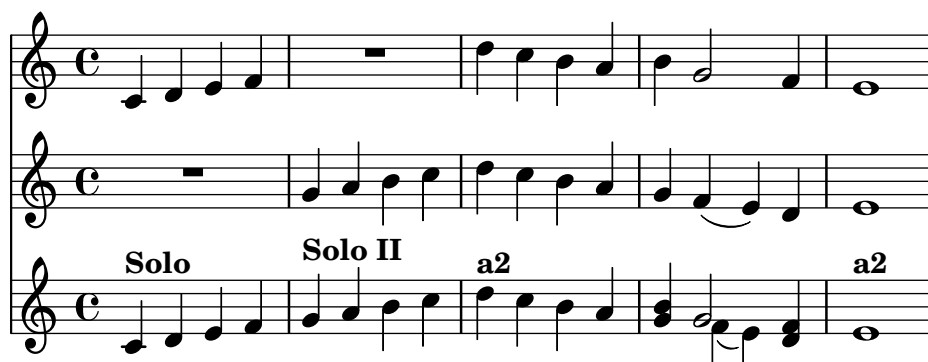
```
\partCombine espressione-musicale1 espressione-musicale2
```

L'esempio seguente illustra il funzionamento di base: le parti sono poste su un unico rigo in modo polifonico e le direzioni dei gambi sono regolate di conseguenza. Si usano le stesse variabili per le parti indipendenti e il rigo combinato.

```
instrumentOne = \relative {
  c'4 d e f |
  R1 |
  d'4 c b a |
  b4 g2 f4 |
  e1 |
}

instrumentTwo = \relative {
  R1 |
  g'4 a b c |
  d4 c b a |
  g4 f( e) d |
  e1 |
}

<<
\new Staff \instrumentOne
\new Staff \instrumentTwo
\new Staff \partCombine \instrumentOne \instrumentTwo
>>
```



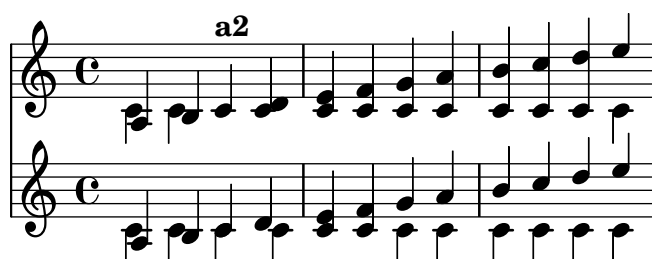
Entrambe le parti hanno note identiche nella terza misura, dunque viene stampata una sola nota. Le direzioni dei gambi e delle legature di portamento e di valore sono impostate automaticamente, a seconda che l'esecuzione delle parti sia solistica o all'unisono. Quando si rende necessario, in caso di polifonia, la prima parte (nel contesto one) ha i gambi in "su", mentre la seconda (nel contesto two) ha sempre i gambi in "giù". In caso di parti solistiche, la prima e seconda parte sono contrassegnate rispettivamente con "Solo" e "Solo II". Le parti (*a due*) all'unisono sono contrassegnate con la scritta "a2".

Il comportamento predefinito del combinatore delle parti è quello di unire due note della stessa altezza in una nota *a due*, combinare in un accordo note della stessa durata e separate da un intervallo inferiore alla nona e separare in voci distinte le note con un intervallo superiore alla nona (o quando le voci collidono). Tale comportamento può essere modificato con un argomento opzionale di una coppia di numeri dopo il comando `\partCombine`: il primo indica l'intervallo in cui le note iniziano a essere combinate (il valore predefinito è zero) e il secondo dove le note vengono divise in note distinte. Impostando il secondo argomento su zero, il combinatore delle parti divide le note che hanno un intervallo di una seconda o più; impostandolo su uno, divide le note di una terza o più, e così via.

```
instrumentOne = \relative {
  a4 b c d |
  e f g a |
  b c d e |
}
```

```
instrumentTwo = \relative {
  c'4 c c c |
  c c c c |
  c c c c |
}
```

```
<<
  \new Staff \partCombine \instrumentOne \instrumentTwo
  \new Staff \partCombine #'(2 . 3) \instrumentOne \instrumentTwo
>>
```



Entrambi gli argomenti di `\partCombine` sono interpretati come contesti Voice separati, dunque se la musica viene inserita in modo relativo *entrambe* le parti devono contenere una funzione `\relative`, ovvero:

```
\partCombine
  \relative ... espressione-musicale1
  \relative ... espressione-musicale2
```

Un blocco `\relative` che racchiude un `\partCombine` non ha effetto sulle altezze di *espressione-musicale1* e *espressione-musicale2*.

Nelle partiture professionali, spesso le voci sono tenute separate per lunghi passaggi anche se alcune note sono le stesse in entrambe le voci e potrebbero essere stampate come unisono. Combinare le note in un accordo o mostrare una voce come solista, dunque, non è la scelta ideale, perché la funzione `\partCombine` considera ogni nota individualmente. In questo caso si può sovrascrivere la funzione `\partCombine` con uno dei comandi elencati sotto. Tutti i comandi devono essere preceduti da `\once` per applicarli soltanto alla nota successiva dell'espressione musicale.

- `\partCombineApart` mantiene le note su due voci distinte, anche se potrebbero essere combinate in un accordo o in un unisono.
- `\partCombineChords` unisce le note in un accordo.
- `\partCombineUnisono` unisce entrambe le voci come “unisono”.
- `\partCombineSoloI` stampa soltanto la prima voce e la contrassegna come un “Solo”.
- `\partCombineSoloII` stampa soltanto la seconda voce e la contrassegna come un “Solo”.
- `\partCombineAutomatic` termina le funzioni dei comandi precedenti e ripristina il funzionamento predefinito di `\partCombine`.

```
instrumentOne = \relative c' {
  \partCombineApart c2^"separato" e |
  \partCombineAutomatic e2^"automatico" e |
  \partCombineChords e'2^"accordo" e |
  \partCombineAutomatic c2^"automatico" c |
  \partCombineApart c2^"separato" \once \partCombineChords e^"accordo una volta sola"
  c2 c |
}
instrumentTwo = \relative {
  c'2 c |
  e2 e |
  a,2 c |
  c2 c' |
  c2 c |
  c2 c |
}

<<
  \new Staff { \instrumentOne }
  \new Staff { \instrumentTwo }
  \new Staff { \partCombine \instrumentOne \instrumentTwo }
>>
```

Uso di \partCombine col testo vocale

Il comando `\partCombine` non è progettato per funzionare col testo vocale; al punto che se una delle voci è nominata in modo esplicito per poterle assegnare del testo, l'unione delle parti smette di funzionare. Tuttavia, questo risultato si può ottenere usando un contesto `NullVoice`. Vedi [Polifonia con testo in comune], pagina 308.

Frammenti di codice selezionati

Combinare due parti sullo stesso rigo

Lo strumento di unione delle parti (il comando `\partCombine`) permette di combinare varie parti sullo stesso rigo. Indicazioni testuali come “solo” e “a2” sono aggiunte automaticamente; per toglierle basta impostare la proprietà `printPartCombineTexts` su `f`. Per le partiture vocali (inni), non c'è bisogno di aggiungere i testi “solo/a2”, quindi dovrebbero essere disattivati. Tuttavia potrebbe convenire non usarlo se c'è una qualche parte solista, perché non verrebbe indicata. In tali casi è preferibile usare la notazione polifonica normale.

Questo frammento illustra i tre modi con cui due parti possono essere stampate su uno stesso rigo: normale polifonia, `\partCombine` senza testo e `\partCombine` con testo.

```
musicUp = \relative c' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    \new Staff \with {
      instrumentName = "standard polyphony"
    } << \musicUp \\\musicDown >>

    \new Staff \with {
      instrumentName =
```

```

\markup { \typewriter "\\partCombine" without text}
printPartCombineTexts = ##f
} \partCombine \musicUp \musicDown

\new Staff \with {
  instrumentName =
  \markup { \typewriter "\\partCombine" with text}
} \partCombine \musicUp \musicDown
>>

\layout {
  indent = 6.0\cm
  \context {
    \Score
    % Setting this to a large value avoids a bar line at the
    % beginning that would connect the three staves otherwise.
    \override SystemStartBar.collapse-height = 30
  }
}

```

standard polyphony

\partCombine without text

\partCombine with text



Changing \partCombine texts

When using the automatic part combining feature, the printed text for the solo and unison sections may be changed.

```

\new Staff <<
  \set Staff.soloText = "girl"
  \set Staff.soloIIIText = "boy"
  \set Staff.aDueText = "together"
  \partCombine
  \relative c'' {
    g4 g r r
    a2 g
  }
  \relative c'' {
    r4 r a( b)
    a2 g
  }
}
>>

```



Vedi anche

Glossario Musicale: Sezione “a due” in *Glossario Musicale*, Sezione “parte” in *Glossario Musicale*.

Guida alla notazione: Sezione 1.6.3 [Scrittura delle parti], pagina 219.

Frammenti: Sezione “Simultaneous notes” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “PartCombineMusic” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Tutte le funzioni `\partCombine...` possono accettare soltanto due voci.

Le funzioni `\partCombine...` non possono essere inserite all’interno di un blocco `\tuplet` o `\relative`.

Se `printPartCombineTexts` è attivo e le due voci eseguono le stesse note “in modo discontinuo” nella stessa misura, potrebbe apparire il testo a2 più di una volta in quella misura.

`\partCombine` sa soltanto quando una nota inizia in una voce (Voice); non può, ad esempio, ricordare se una nota in una voce è già iniziata quando combina le note già iniziate nell’altra voce. Questo può portare a vari problemi inattesi, tra cui la stampa non corretta dei segni “Solo” e “Unisono”.

`\partCombine` tiene tutti gli estensori (legature di portamento e di valore, forcelle, etc.) nella stessa voce, quindi se uno di questi estensori inizia o termina in una voce diversa potrebbe essere stampato incorrettamente o non essere stampato affatto.

Se la funzione `\partCombine` non riesce a combinare le due espressioni musicali (ovvero quando le due voci hanno durate diverse), assegnerà alle voci, internamente, nomi personalizzati: rispettivamente `one` e `two`. Ciò significa che se c’è un “passaggio” a un contesto Voice nominato diversamente, gli eventi in quel contesto verranno ignorati.

Consultare i *Problemi noti e avvertimenti* in [Intavolature predefinite], pagina 360, se si usa `\partCombine` con l’intavolatura, e la *Nota* in [Travature automatiche], pagina 88, se si usa la disposizione automatica delle travature.

Scrivere la musica in parallelo

La musica che contiene parti diverse può essere messa in parallelo nel codice di input. La funzione `\parallelMusic` accetta una lista contenente i nomi di un insieme di variabili da creare e un’espressione musicale. Il contenuto delle misure alternate nell’espressione diventa il valore delle rispettive variabili, in modo che possano essere usate successivamente per stampare la musica.

Nota: L’uso dei controlli di battuta | è obbligatorio e le misure devono avere la stessa durata.

```
\parallelMusic voiceA,voiceB,voiceC {
  % Battuta 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~ 4 r16 e'8.~ 4 |
  c'2 c'2 |
```

```

% Battuta 2
r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
r16 d'8.~ 4 r16 d'8.~ 4 |
c'2 c'2 |

}
\new StaffGroup <<
  \new Staff << \voiceA \\\ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>

```



L'uso del modo relativo è permesso. Attenzione: il comando `\relative` non deve essere messo dentro `\parallelMusic`. Le note sono relative alla nota precedente della voce, non a quella precedente nell'input. In altre parole, le note relative di voiceA ignorano le note in voiceB.

```

\parallelMusic voiceA,voiceB,voiceC {
  % Battuta 1
  r8 g16 c e g, c e r8 g,16 c e g, c e |
  r16 e8.~ 4 r16 e8.~ 4 |
  c2 c |

  % Battuta 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~ 4 r16 d8.~ 4 |
  c2 c |

}
\new StaffGroup <<
  \new Staff << \relative c' \voiceA \\\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>

```



Questo è molto utile nella musica per pianoforte. Questo esempio combina sezioni di quattro battute consecutive con quattro variabili:

```

global = {
  \key g \major
  \time 2/4

```

```

}

\parallelMusic voiceA,voiceB,voiceC,voiceD {
  % Battuta 1
  a8    b    c    d    |
  d4          e    |
  c16 d e fis d e fis g |
  a4          a    |

  % Battuta 2
  e8    fis g    a    |
  fis4          g    |
  e16 fis g a fis g a b |
  a4          a    |

  % Bar 3 ...
}

\score {
  \new PianoStaff <<
    \new Staff {
      \global
      <<
        \relative c'' \voiceA
        \\
        \relative c' \voiceB
      >>
    }
    \new Staff {
      \global \clef bass
      <<
        \relative c \voiceC
        \\
        \relative c \voiceD
      >>
    }
  >>
}

```



Vedi anche

Manuale d'apprendimento: Sezione "Organizzare i brani con le variabili" in *Manuale di Apprendimento*.

Frammenti: Sezione "Simultaneous notes" in *Frammenti di codice*.

1.6 Notazione del rigo

The image shows a musical score for three instruments: Trumpet Bb, Tambourine, and Piano. The score is in 2/4 time and features two systems. The first system has two measures. The second system has three measures, with a '3' above the first measure. The Trumpet Bb part is in Bb major, the Tambourine in 2/4, and the Piano in Bb major. The Piano part includes dynamics 'p' and 'p grazioso'.

Questa sezione spiega come modificare l'aspetto del rigo, come stampare partiture multirigo e come aggiungere indicazioni di tempo e citazioni in corpo più piccolo nel rigo.

1.6.1 Aspetto del rigo

Questa sezione presenta i diversi metodi per creare e raggruppare i righi.

Istanziare nuovi righi

Il *rigo musicale* si crea con i comandi `\new` o `\context`. Ulteriori dettagli in Sezione 5.1.2 [Creazione e citazione di un contesto], pagina 607.

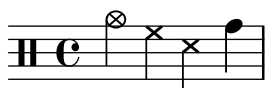
Il contesto di base del rigo è `Staff`:

```
\new Staff \relative { c'4 d e f }
```



Il contesto `DrumStaff` crea un rigo di cinque linee impostato per una tipica batteria. Ogni strumento viene mostrato con un simbolo diverso. Gli strumenti si inseriscono nella modalità percussioni, che si attiva col comando `\drummode`: ogni strumento viene indicato con un nome. Ulteriori dettagli in [Righi delle percussioni], pagina 414.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



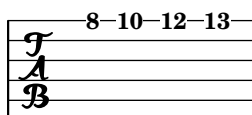
`RhythmicStaff` crea un rigo con una sola linea che mostra soltanto i valori ritmici dell'input. Le durate reali vengono mantenute. Ulteriori dettagli in [Mostrare i ritmi della melodia], pagina 86.

```
\new RhythmicStaff { c4 d e f }
```



`TabStaff` crea un'intavolatura (o tablatura) con sei corde nell'accordatura standard per chitarra. Ulteriori dettagli in [Intavolature predefinite], pagina 360.

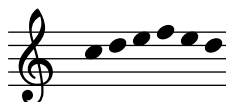
```
\new TabStaff \relative { c''4 d e f }
```



Ci sono due contesti del rigo specifici per la notazione di musica antica, `MensuralStaff` e `VaticanaStaff`, descritti in [Contesti predefiniti], pagina 456.

Il contesto `GregorianTranscriptionStaff` crea un rigo per il canto gregoriano moderno. Non mostra le stanghette delle battute.

```
\new GregorianTranscriptionStaff \relative { c''4 d e f e d }
```



Si possono creare nuovi contesti per un singolo rigo, come è spiegato dettagliatamente in Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

Vedi anche

Glossario musicale: Sezione “rigo” in *Glossario Musicale*,

Guida alla notazione: Sezione 5.1.2 [Creazione e citazione di un contesto], pagina 607, [Righi delle percussioni], pagina 414, [Mostrare i ritmi della melodia], pagina 86, [Intavolature predefinite], pagina 360, [Contesti predefiniti], pagina 456, [Simbolo del rigo], pagina 210, [Contesti del canto gregoriano], pagina 465, [Contesti mensurali], pagina 458, Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

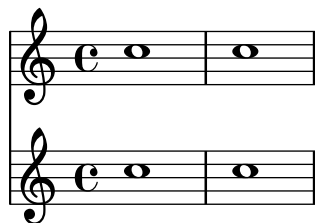
Guida al funzionamento interno: Sezione “Staff” in *Guida al Funzionamento Interno*, Sezione “DrumStaff” in *Guida al Funzionamento Interno*, Sezione “GregorianTranscriptionStaff” in *Guida al Funzionamento Interno*, Sezione “RhythmicStaff” in *Guida al Funzionamento Interno*, Sezione “TabStaff” in *Guida al Funzionamento Interno*, Sezione “MensuralStaff” in *Guida al Funzionamento Interno*, Sezione “VaticanaStaff” in *Guida al Funzionamento Interno*, Sezione “StaffSymbol” in *Guida al Funzionamento Interno*.

Raggruppare i righi

Esistono vari contesti per raggruppare insieme singoli righi in modo da formare sistemi multirigo. Ogni contesto di raggruppamento imposta il comportamento delle stanghette e lo stile del segno che delimita l'inizio del sistema.

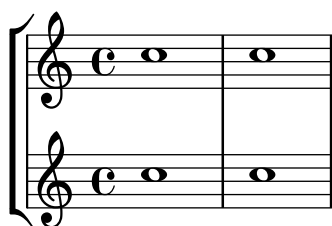
Se non si specifica alcun contesto, vengono usate le proprietà predefinite: il gruppo inizia con una linea verticale e le stanghette non sono collegate.

```
<<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



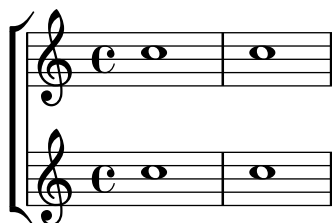
Nel contesto `StaffGroup`, il gruppo inizia con una parentesi quadra e le stanghette attraversano tutti i righi.

```
\new StaffGroup <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



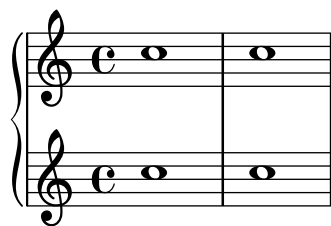
Nel contesto `ChoirStaff`, il gruppo inizia con una parentesi quadra, ma le stanghette non sono collegate.

```
\new ChoirStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



Nel contesto `GrandStaff`, il gruppo inizia con una parentesi graffa e le stanghette sono collegate da rigo a rigo.

```
\new GrandStaff <<
  \new Staff \relative { c''1 c }
  \new Staff \relative { c''1 c }
>>
```



Il contesto `PianoStaff` è identico a `GrandStaff`, con l'unica differenza che permette di mostrare il nome dello strumento direttamente. Ulteriori dettagli in [Nomi degli strumenti], pagina 219.

```
\new PianoStaff \with { instrumentName = "Piano" }
<<
  \new Staff \relative { c'1 c }
  \new Staff \relative { \clef bass c1 c }
>>
```



Ogni contesto per il gruppo di righe imposta la proprietà `systemStartDelimiter` su uno dei seguenti valori: `SystemStartBar`, `SystemStartBrace` o `SystemStartBracket`. È presente anche un quarto delimitatore, `SystemStartSquare`, ma deve essere indicato esplicitamente.

Si possono definire nuovi contesti di gruppi di rigo. I dettagli sono spiegati in Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

Frammenti di codice selezionati

Usare una parentesi quadra all'inizio di un gruppo di righe

Si può usare il segno `SystemStartSquare` (uno dei segni che delimitano l'inizio del sistema) impostandolo esplicitamente in un contesto `StaffGroup` o `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
    \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```

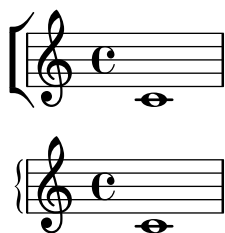


Mostrare la parentesi anche se c'è un solo rigo nel sistema

Se c'è un solo rigo in uno dei tipi di rigo `ChoirStaff` o `StaffGroup`, la parentesi e la stanghetta iniziale non appaiono. Si può modificare questo comportamento predefinito sovrascrivendo `collapse-height` e impostando un valore inferiore al numero di linee del rigo.

Nei contesti `PianoStaff` e `GrandStaff`, dove i sistemi iniziano con una parentesi graffa invece di una parentesi quadra, occorre impostare un'altra proprietà, come si vede nel secondo sistema dell'esempio.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = 4
    \override Score.SystemStartBar.collapse-height = 4
    \new Staff {
      c'1
    }
  >>
}
```



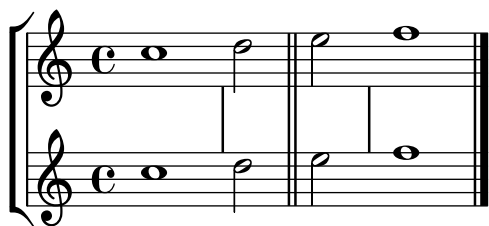
Formattazione mensurale (stanghette tra i righi)

La formattazione mensurale, in cui le stanghette non appaiono sui righi ma nello spazio tra i righi, si può ottenere usando `StaffGroup` al posto di `ChoirStaff`. La stanghetta sui righi viene nascosta con `\hide`.

```
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music
>>
```



Vedi anche

Glossario musicale: Sezione “graffa” in *Glossario Musicale*, Sezione “parentesi quadra” in *Glossario Musicale*, Sezione “accollatura” in *Glossario Musicale*.

Guida alla notazione: [Nomi degli strumenti], pagina 219, Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Staff” in *Guida al Funzionamento Interno*, Sezione “StaffGroup” in *Guida al Funzionamento Interno*, Sezione “ChoirStaff” in *Guida al Funzionamento Interno*, Sezione “GrandStaff” in *Guida al Funzionamento Interno*, Sezione “PianoStaff” in *Guida al Funzionamento Interno*, Sezione “SystemStartBar” in *Guida al Funzionamento Interno*, Sezione “SystemStartBrace” in *Guida al Funzionamento Interno*, Sezione “SystemStartBracket” in *Guida al Funzionamento Interno*, Sezione “SystemStartSquare” in *Guida al Funzionamento Interno*.

Gruppi di righi annidati

I contesti dei gruppi di righi possono essere annidati fino a qualsiasi livello. In questo caso, ogni contesto inferiore crea una nuova parentesi accanto alla parentesi del gruppo superiore.

```
\new StaffGroup <<
  \new Staff \relative { c'2 c | c2 c }
  \new StaffGroup <<
    \new Staff \relative { g'2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
  <<
    \new Staff \relative { e'2 e | e2 e }
    \new Staff \relative { c'2 c | c2 c }
  >>
>>
>>
```



Si possono definire nuovi gruppi di righi annidati. Ulteriori dettagli in Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

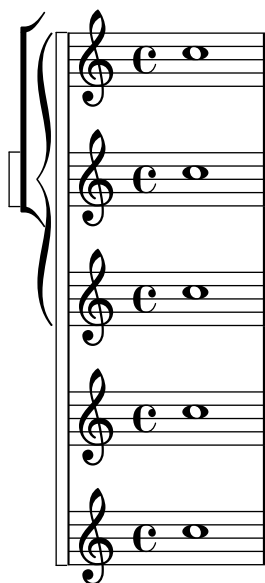
Frammenti di codice selezionati

Annidare i righi

Si può usare la proprietà `systemStartDelimiterHierarchy` per creare gruppi di righi annidati più complessi. Il comando `\set StaffGroup.systemStartDelimiterHierarchy` prende come argomento una lista alfabetica dell'insieme di righi prodotti. Prima di ogni rigo si può assegnare un delimitatore di inizio del sistema. Deve essere racchiuso tra parentesi e collega tutti i righi compresi tra le parentesi. Gli elementi nella lista possono essere omessi, ma la prima parentesi quadra collega sempre tutti i righi. Le possibilità sono `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` e `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = 4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare
          (SystemStartBrace
            (SystemStartBracket a
              (SystemStartSquare b))
            c)
          d)

  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Vedi anche

Guida alla notazione: [Raggruppare i righi], pagina 203, [Nomi degli strumenti], pagina 219, Sezione 5.1.6 [Definizione di nuovi contesti], pagina 620.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “StaffGroup” in *Guida al Funzionamento Interno*, Sezione “ChoirStaff” in *Guida al Funzionamento Interno*, Sezione “SystemStartBar” in *Guida al Funzionamento Interno*, Sezione “SystemStartBrace” in *Guida al Funzionamento Interno*, Sezione “SystemStartBracket” in *Guida al Funzionamento Interno*, Sezione “SystemStartSquare” in *Guida al Funzionamento Interno*.

Separare i sistemi

Se il numero di sistemi per pagina cambia di pagina in pagina, è consuetudine separare i sistemi con un segno separatore. Per impostazione predefinita questo segno è disattivo, ma può essere attivato con un’opzione in `\paper`.

```
\book {
  \score {
    \new StaffGroup <<
      \new Staff {
        \relative {
          c''4 c c c
          \break
          c4 c c c
        }
      }
      \new Staff {
        \relative {
          c''4 c c c
          \break
          c4 c c c
        }
      }
    >>
  }
}
```

```

\paper {
  system-separator-markup = \slashSeparator
  % i seguenti comandi servono soltanto alla formattazione di questa documentazione
  paper-width = 100\mm
  paper-height = 100\mm
  tagline = ##f
}
}

```



Vedi anche

Guida alla notazione: Sezione 4.1 [Formattazione della pagina], pagina 552.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

1.6.2 Modificare singoli righi

Questa sezione spiega come modificare gli attributi specifici di un rigo, per esempio il numero di linee o la dimensione del rigo. Vengono descritti anche i metodi per iniziare e finire un rigo e per impostare le sezioni ossia.

Simbolo del rigo

I comandi `\stopStaff` e `\startStaff` servono a fermare o (ri)avviare le linee del rigo, per impedire che appaiano in un punto della partitura.

```

\relative {
  \stopStaff f'4 d \startStaff g, e
  f'4 d \stopStaff g, e
  f'4 d \startStaff g, e
}

```



Comandi predefiniti

`\startStaff`, `\stopStaff`.

Le linee di un rigo appartengono all'oggetto `StaffSymbol` (che comprende i tagli aggiuntivi) e si possono modificare tramite le proprietà di `StaffSymbol`; però queste modifiche devono essere fatte prima che il rigo sia (ri)avviato.

Si può cambiare il numero di linee del rigo:

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-count = #2
  \startStaff g, e |

  f'4 d \stopStaff
  \revert Staff.StaffSymbol.line-count
  \startStaff g, e |
}
```



Si può cambiare anche la posizione di ogni linea del rigo. Un elenco di numeri definisce la posizione di ogni linea. I valori consueti sono 0 per la linea centrale e (-4 -2 0 2 4) per le altre. La linea del rigo appare solo se è presente il suo valore, quindi questo comando permette di variare anche il numero delle linee, oltre alla loro posizione.

```
\relative {
  f''4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(1 3 5 -1 -3)
  \startStaff g, e |
  f'4 d \stopStaff
  \override Staff.StaffSymbol.line-positions = #'(8 6.5 -6 -8 -0.5)
  \startStaff g, e |
}
```



Per conservare le tipiche direzioni dei gambi (nella metà inferiore del rigo i gambi puntano in su, mentre in quella superiore sono rivolti in giù), occorre allineare la linea centrale (o lo spazio) del rigo personalizzato alla posizione della linea centrale normale (0). Potrà essere necessario regolare la posizione della chiave e del Do centrale per adattarsi alle nuove linee. Si veda [Chiave], pagina 17.

Si può modificare lo spessore della linea del rigo. Per impostazione predefinita, questa modifica ha effetto anche sui tagli addizionali e sui gambi.

```
\new Staff \with {
  \override StaffSymbol.thickness = #3
} \relative {
  f''4 d g, e
}
```



È anche possibile impostare lo spessore dei tagli addizionali in modo indipendente dalle linee del rigo.

```
\new Staff \with {
```



```

\override StaffSymbol.thickness = #2
\override StaffSymbol.ledger-line-thickness = #'(0.5 . 0.4)
} \relative {
  f'''4 a, a,, f
}

```



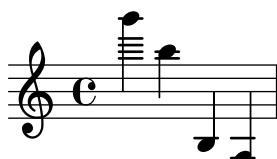
Il primo valore viene moltiplicato per lo spessore della linea del rigo, il secondo per la spaziatura del rigo; la somma dei due valori definisce il nuovo valore dello spessore del taglio addizionale.

Si possono modificare le posizioni verticali dei tagli addizionali:

```

\new Staff \with {
  \override StaffSymbol.ledger-positions = #'(-3 -2 -1 2 5 6)
} \relative {
  f'''4 a, a,, f
}

```



Si possono far apparire ulteriori tagli addizionali sopra o sotto le teste delle note, a seconda della posizione corrente relativa alle altre teste, anch'esse con i propri tagli addizionali.

```

\new Staff \with {
  \override StaffSymbol.ledger-extra = #4
} \relative {
  f'''4 a, d, f,
}

```



Si possono far apparire i tagli addizionali anche dentro il rigo quando servono delle linee personalizzate. L'esempio mostra la posizione predefinita dei tagli addizionali quando la proprietà `ledger-position` è impostata e quando non lo è. Nell'esempio il comando `\stopStaff` serve ad annullare il comando `\override` per l'oggetto `StaffSymbol`.

```

\relative d' {
  \override Staff.StaffSymbol.line-positions = #'(-8 0 2 4)
  d4 e f g
  \stopStaff
  \startStaff
  \override Staff.StaffSymbol.ledger-positions = #'(-8 -6 (-4 -2) 0)
  d4 e f g
}

```



Si può cambiare la distanza tra le linee del rigo. Tale modifica ha effetto anche sulla spaziatura della linea.

```
\new Staff \with {
  \override StaffSymbol.staff-space = #1.5
} \relative {
  f' '4 d, g, e,
}
```



Frammenti di codice selezionati

Rendere alcune linee del rigo più spesse delle altre

In ambito didattico può essere utile rendere più spesso una linea del rigo (per esempio, la linea centrale, o per sottolineare la linea della chiave di Sol). Per farlo si possono aggiungere altre linee e posizzarle molto vicino alla linea che deve essere evidenziata, usando la proprietà `line-positions` dell'oggetto `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Vedi anche

Glossario musicale: Sezione “linea” in *Glossario Musicale*, Sezione “taglio addizionale” in *Glossario Musicale*, Sezione “rigo (o pentagramma)” in *Glossario Musicale*.

Guida alla notazione: [Chiave], pagina 17.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “StaffSymbol” in *Guida al Funzionamento Interno*, Sezione “staff-symbol-interface” in *Guida al Funzionamento Interno*.

Righi ossia

I righi ossia si possono creare aggiungendo un nuovo rigo simultaneo nel punto giusto:

```
\new Staff \relative {
  c' '4 b d c
  <<
  { c4 b d c }
```

```

\new Staff { e4 d f e }
>>
c4 b c2
}

```



Tuttavia, questo esempio non produce quel che normalmente si desidera. Per creare righe ossia che siano sopra il rigo originale, non abbiano indicazione di tempo né chiave e abbiano un tipo di carattere più piccolo, sono necessarie delle modifiche manuali. Il Manuale d'apprendimento descrive una tecnica specifica per ottenere questo risultato, a partire da Sezione “Annidare le espressioni musicali” in *Manuale di Apprendimento*.

L'esempio seguente usa la proprietà `alignAboveContext` per allineare il rigo ossia. Questo metodo conviene quando sono necessari solo pochi righe ossia.

```

\new Staff = "main" \relative {
  c' '4 b d c
  <<
  { c4 b d c }

  \new Staff \with {
    \remove Time_signature_engraver
    alignAboveContext = "main"
    \magnifyStaff #2/3
    firstClef = ##f
  }
  { e4 d f e }
  >>
  c4 b c2
}

```



Se si hanno molti righe ossia isolati, è meglio creare un contesto Staff vuoto con un *identificativo del contesto* specifico; i righe ossia possono essere creati *chiamando* questo contesto e usando `\startStaff` e `\stopStaff` nei punti richiesti. I vantaggi di questo metodo sono più evidenti se il brano è più lungo del seguente esempio.

```

<<
\new Staff = "ossia" \with {
  \remove Time_signature_engraver
  \hide Clef
  \magnifyStaff #2/3
}
{ \stopStaff s1*6 }

```

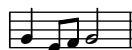
```

\new Staff \relative {
  c'4 b c2
  <<
    { e4 f e2 }
    \context Staff = "ossia" {
      \startStaff e4 g8 f e2 \stopStaff
    }
  >>
  g4 a g2 \break
  c4 b c2
  <<
    { g4 a g2 }
    \context Staff = "ossia" {
      \startStaff g4 e8 f g2 \stopStaff
    }
  >>
  e4 d c2
}
>>

```



4



Altrimenti si può usare il comando `\RemoveAllEmptyStaves` per creare i righi ossia. Questo metodo conviene quando i righi ossia si trovano subito dopo un'interruzione di linea. Ulteriori informazioni su `\RemoveAllEmptyStaves` si trovano in [Nascondere i righi], pagina 217.

```

<<
\new Staff = "ossia" \with {
  \remove Time_signature_engraver
  \hide Clef
  \magnifyStaff #2/3
  \RemoveAllEmptyStaves
} \relative {
  R1*3
  c' '4 e8 d c2
}
\new Staff \relative {
  c'4 b c2
  e4 f e2
  g4 a g2 \break
  c4 b c2

```

```

      g4 a g2
      e4 d c2
    }
  >>

```



Frammenti di codice selezionati

Allineare verticalmente gli ossia e il testo vocale

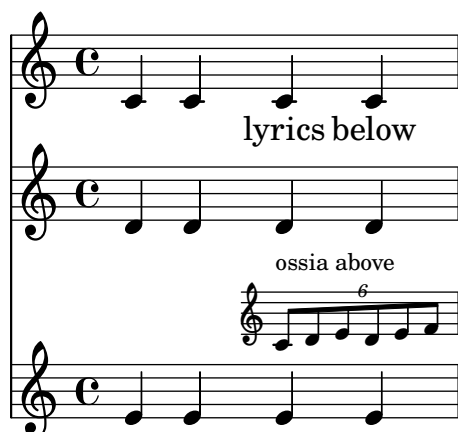
Questo frammento mostra come usare le proprietà di contesto `alignBelowContext` e `alignAboveContext` per controllare il posizionamento del testo vocale e degli ossia.

```

\relative c' <<
  \new Staff = "1" { c4 c c c }
  \new Staff = "2" { d4 d d d }
  \new Staff = "3" { e4 e e e }

  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = "1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = "3"
        fontSize = -2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
        \override VerticalAxisGroup.staff-staff-spacing =
          #'((minimum-distance . 0)
             (basic-distance . 0)
             (padding . 1))
      } {
        \tuplet 6/4 {
          \override TextScript.padding = 2
          c8[~"ossia above" d e d e f]
        }
      }
    }
  >>
}
>>

```



Vedi anche

Glossario musicale: Sezione “ossia” in *Glossario Musicale*, Sezione “rigo (pentagramma)” in *Glossario Musicale*, Sezione “rigo temporaneo” in *Glossario Musicale*.

Manuale d’apprendimento: Sezione “Annidare le espressioni musicali” in *Manuale di Apprendimento*, Sezione “Dimensione degli oggetti” in *Manuale di Apprendimento*, Sezione “Lunghezza e spessore degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: [Nascondere i righi], pagina 217.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “StaffSymbol” in *Guida al Funzionamento Interno*.

Nascondere i righi

Le linee del rigo si possono nascondere togliendo l’incisore `Staff_symbol_engraver` dal contesto `Staff`. Altrimenti si può usare `\stopStaff`.

```
\new Staff \with {
  \remove Staff_symbol_engraver
}
\relative { a''8 f e16 d c b a2 }
```



I righi vuoti si possono nascondere (per la cosiddetta ‘partitura alla francese’) applicando il comando `\RemoveEmptyStaves` a un contesto, che può essere fatto globalmente (in un blocco `\layout`) oppure soltanto per rigi specifici (in un blocco `\with`). Questo comando toglie tutti i rigi vuoti di una partitura eccetto quelli nel primo sistema. Per nascondere anche quelli del primo sistema usare il comando `\RemoveAllEmptyStaves`. I contesti supportati sono `Staff`, `RhythmicStaff` e `VaticanaStaff`.

Nota: Un rigo viene considerato vuoto quando contiene soltanto pause multiple, pause, salti, pause spaziatrici o una combinazione di questi elementi.

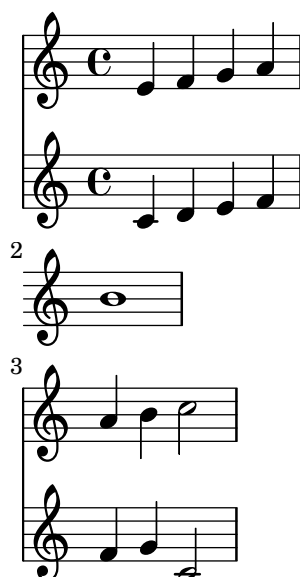
```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
```

```

    }
  }

  \relative <<
    \new Staff {
      e'4 f g a \break
      b1 \break
      a4 b c2
    }
    \new Staff {
      c,4 d e f \break
      R1 \break
      f4 g c,2
    }
  >>

```



`\RemoveAllEmptyStaves` si può usare anche per creare sezioni ossia per un rigo. I dettagli si trovano in [Righi ossia], pagina 213.

Comandi predefiniti

`\RemoveEmptyStaves`, `\RemoveAllEmptyStaves`.

Vedi anche

Glossario musicale: Sezione “rigo temporaneo” in *Glossario Musicale*.

Manuale d’apprendimento: Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.1.5 [Modifica delle impostazioni predefinite di un contesto], pagina 615, [Simbolo del rigo], pagina 210, [Righi ossia], pagina 213, [Note nascoste], pagina 240, [Pause invisibili], pagina 65, Sezione 5.4.7 [Visibilità degli oggetti], pagina 652.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “ChordNames” in *Guida al Funzionamento Interno*, Sezione “FiguredBass” in *Guida al Funzionamento Interno*, Sezione “Lyrics” in *Guida al Funzionamento Interno*, Sezione “Staff” in *Guida al Funzionamento Interno*, Sezione “Vertical-AxisGroup” in *Guida al Funzionamento Interno*, Sezione “Staff_symbol_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Se si toglie l'incisore `Staff_symbol_engraver` vengono nascoste anche le stanghette. Se si forza la visibilità delle stanghette, potrebbero verificarsi degli errori di formattazione. In questo caso, conviene usare i seguenti comandi invece di togliere l'incisore:

```
\omit StaffSymbol
\override NoteHead.no-ledgers = ##t
```

Per i problemi noti e gli avvertimenti relativi a `\Staff \RemoveEmptyStaves` si veda Sezione 5.1.5 [Modifica delle impostazioni predefinite di un contesto], pagina 615.

1.6.3 Scrittura delle parti

Questa sezione spiega come inserire in una partitura le indicazioni di tempo e i nomi degli strumenti. Mostra anche come citare altre voci e come formattare le citazioni in corpo più piccolo.

Nomi degli strumenti

I nomi degli strumenti possono essere fatti apparire, alla sinistra dei righi, nei contesti `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` e `ChoirStaff`. Il valore di `instrumentName` viene usato per il primo rigo e quello di `shortInstrumentName` per tutti i righi successivi.

```
\new Staff \with {
  instrumentName = "Violin "
  shortInstrumentName = "Vln. "
} \relative {
  c'4.. g'16 c4.. g'16 \break | c1 |
}
```



Si può usare `\markup` per creare nomi più complessi:

```
\new Staff \with {
  instrumentName = \markup {
    \column { "Clarinetti"
      \line { "in B" \smaller \flat }
    }
  }
} \relative {
  c'4 c,16 d e f g2
}
```



Se due o più contesti del rigo sono raggruppati insieme, i nomi degli strumenti, sia quello normale che quello abbreviato, vengono centrati automaticamente. Per allineare al centro i nomi degli strumenti che vanno a capo, occorre usare `\center-column`:

```
<<
```



```

\new Staff \with {
  instrumentName = "Flute"
} \relative {
  f''2 g4 f
}
\new Staff \with {
  instrumentName = \markup {
    \center-column { "Clarinet"
      \line { "in B" \smaller \flat }
    }
  }
} \relative { c''4 b c2 }
>>

```



Tuttavia, se i nomi degli strumenti sono lunghi, potranno essere centrati solo aumentando i valori di indent e short-indent. Ulteriori dettagli su queste impostazioni si trovano in [Variabili \paper per spostamenti e indentazioni], pagina 560.

```

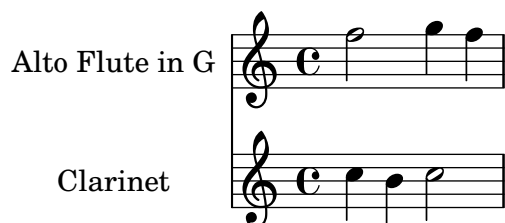
<<
\new Staff \with {
  instrumentName = "Alto Flute in G"
  shortInstrumentName = "Flt."
} \relative {
  f''2 g4 f \break
  g4 f g2
}
\new Staff \with {
  instrumentName = "Clarinet"
  shortInstrumentName = "Clar."
} \relative {
  c''4 b c2 \break
  c2 b4 c
}
>>

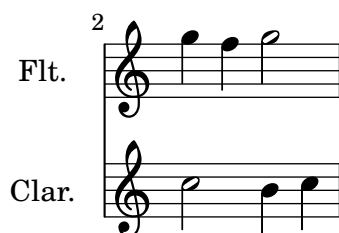
```

```

\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}

```





Per impostare i nomi degli strumenti in altri contesti (come ChordNames o FiguredBass), si deve aggiungere l'incisore `Instrument_name_engraver` a quel contesto. Ulteriori dettagli in Sezione 5.1.4 [Modifica dei componenti aggiuntivi di un contesto], pagina 613.

`shortInstrumentName` può essere cambiato all'interno di un brano, insieme a altre impostazioni necessarie al nuovo strumento. Tuttavia, di `instrumentName` apparirà solo la prima definizione e le modifiche successive saranno ignorate:

```

prepPiccolo = <>^\markup \italic { muta in Piccolo }

prepFlute = <>^\markup \italic { muta in Flauto }

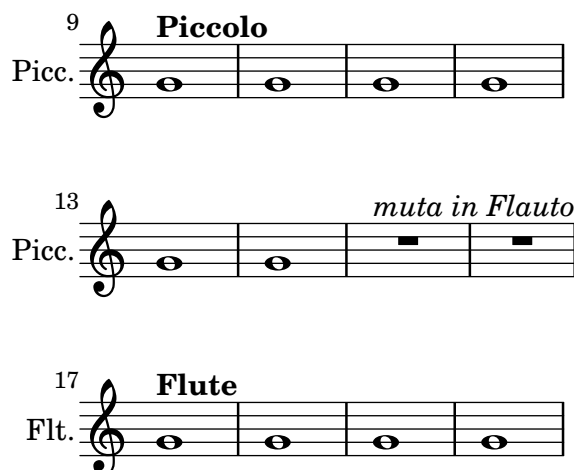
setPiccolo = {
  <>^\markup \bold { Piccolo }
  \transposition c''
}

setFlute = {
  <>^\markup \bold { Flute }
  \transposition c'
}

\new Staff \with {
  instrumentName = "Flute"
  shortInstrumentName = "Flt."
}
\relative {
  g'1 g g g \break
  g1 g \prepPiccolo R R \break
  \set Staff.instrumentName = "Piccolo"
  \set Staff.shortInstrumentName = "Picc."
  \setPiccolo
  g1 g g g \break
  g1 g \prepFlute R R \break
  \set Staff.instrumentName = "Flute"
  \set Staff.shortInstrumentName = "Flt."
  \setFlute
  g1 g g g
}

```





Vedi anche

Guida alla notazione: [Variabili `\paper` per spostamenti e indentazioni], pagina 560, Sezione 5.1.4 [Modifica dei componenti aggiuntivi di un contesto], pagina 613.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “InstrumentName” in *Guida al Funzionamento Interno*, Sezione “PianoStaff” in *Guida al Funzionamento Interno*, Sezione “Staff” in *Guida al Funzionamento Interno*.

Citare altre voci

È molto comune che una voce usi le stesse note di un'altra voce. Per esempio, il primo e il secondo violino che suonano la stessa frase durante un particolare passaggio del brano. Per evitare di reinserire la musica di nuovo per la seconda voce, si può far sì che una voce *citi* l'altra.

Il comando `\addQuote`, usato nell'ambito di più alto livello, definisce un flusso musicale da cui poter citare i frammenti.

Il comando `\quoteDuring` serve a indicare il punto in cui inizia la citazione. È seguito da due argomenti: il nome della voce citata, come è definito da `\addQuote`, e un'espressione musicale per la durata della citazione.

```
fluteNotes = \relative {
  a'4 gis g gis | b4~"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c'4 cis c b \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



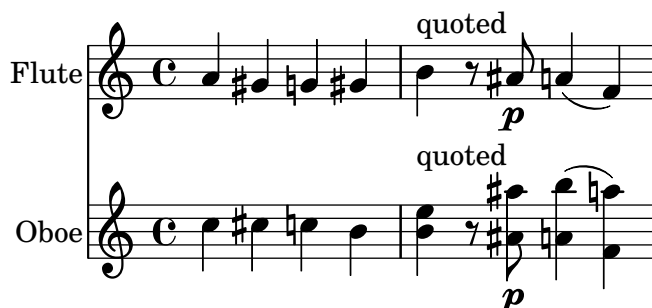
Se l'espressione musicale usata in `\quoteDuring` contiene note invece di pause spaziatrici o multiple, la citazione apparirà in forma polifonica e potrebbe causare risultati indesiderati.

```
fluteNotes = \relative {
  a'4 gis g gis | b4~"quoted" r8 ais\p a4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "flute" { e4 r8 ais b4 a }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Se un comando `\unfoldRepeats` in un'espressione musicale deve essere stampato quando si usa `\quoteDuring`, allora anch'esso deve contenere il suo comando `\unfoldRepeats`;

```
fluteNotes = \relative {
  \repeat volta 2 { a'4 gis g gis }
}

oboeNotesDW = \relative {
  \repeat volta 2 \quoteDuring "incorrect" { s1 }
}

oboeNotesW = \relative {
  \repeat volta 2 \quoteDuring "correct" { s1 }
}
```

```

\addQuote "incorrect" { \fluteNotes }

\addQuote "correct" { \unfoldRepeats \fluteNotes }

\score {
  \unfoldRepeats
  <<
    \new Staff \with { instrumentName = "Flute" }
    \fluteNotes
    \new Staff \with { instrumentName = "Oboe (incorrect)" }
    \oboeNotesDW
    \new Staff \with { instrumentName = "Oboe (correct)" }
    \oboeNotesW
  >>
}

```

The image shows a musical score with three staves. The top staff is labeled 'Flute' and contains a melody in C major. The middle staff is labeled 'Oboe (incorrect)' and is empty. The bottom staff is labeled 'Oboe (correct)' and contains the same melody as the Flute staff, transposed to match the Flute's pitch.

Il comando `\quoteDuring` usa le impostazioni `\transposition` sia della parte citata sia di quella che cita per produrre delle note per la parte che cita che abbiano la stessa altezza di quelle nella parte citata.

```

clarinetNotes = \relative c' {
  \transposition bes
  \key d \major
  b4 ais a ais | cis4~"quoted" r8 bis\p b4( f)
}

oboeNotes = \relative {
  c''4 cis c b \quoteDuring "clarinet" { s1 }
}

\addQuote "clarinet" { \clarinetNotes }

\score {
  <<
    \new Staff \with { instrumentName = "Clarinet" } \clarinetNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}

```



La musica citata include tutte le articolazioni, dinamiche, annotazioni, etc. presenti nel frammento citato. È possibile scegliere quali di questi oggetti far apparire usando la proprietà di contesto `quotedEventTypes`.

```
fluteNotes = \relative {
  a'2 g2 |
  b4\<^"quoted" r8 ais a4\f( c->)
}

oboeNotes = \relative {
  c'2. b4 |
  \quoteDuring "flute" { s1 }
}

\addQuote "flute" { \fluteNotes }

\score {
  <<
    \set Score.quotedEventTypes = #'(note-event articulation-event
                                   crescendo-event rest-event
                                   slur-event dynamic-event)
    \new Staff \with { instrumentName = "Flute" } \fluteNotes
    \new Staff \with { instrumentName = "Oboe" } \oboeNotes
  >>
}
```



Le citazioni possono anche essere contrassegnate; si veda [Uso delle etichette], pagina 527.

Vedi anche

Guida alla notazione: [Trasporto strumentale], pagina 27, [Uso delle etichette], pagina 527.

File installati: `scm/define-event-classes.scm`.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Music classes” in *Guida al Funzionamento Interno*, Sezione “QuoteMusic” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Solo il contenuto della prima voce in un comando `\addQuote` sarà preso in considerazione per la citazione; quindi se l'espressione musicale contiene comandi `\new` o `\context Voice`, il loro contenuto non verrà citato. La citazione degli abbellimenti non è supportata e potrebbe causare il crash di LilyPond; la citazione di terzine annidate potrebbe produrre una notazione mediocre.

Formattazione delle notine

Il modo più semplice per formattare le notine è creare esplicitamente un contesto `CueVoice` all'interno della parte.

```
\relative {
  R1
  <<
    { e'2\rest r4. e8 }
    \new CueVoice {
      \stemUp d'8^"flute" c d e fis2
    }
  >>
  d,,4 r a r
}
```



Si può usare il comando `\cueClef` all'interno di un contesto `CueVoice` esplicito se è richiesto un cambiamento di chiave; in questo modo la chiave apparirà nella dimensione giusta per le notine. Si può poi usare il comando `\cueClefUnset` per tornare alla chiave originale, di nuovo nella dimensione giusta.

```
\relative {
  \clef "bass"
  R1
  <<
    { e'2\rest r4. \cueClefUnset e,8 }
    \new CueVoice {
      \cueClef "treble" \stemUp d''8^"flute" c d e fis2
    }
  >>
  d,,4 r a r
}
```



I comandi `\cueClef` e `\cueClefUnset` si possono usare anche senza un esplicito contesto `CueVoice`.

```
\relative {
  \clef "bass"
  R1
  \cueClef "treble"
  d''8^"flute" c d e fis2
}
```

```
\cueClefUnset
d,,4 r a r
}
```



Per posizionamenti complessi delle notine, per esempio includere la trasposizione o inserire delle notine da varie sorgenti musicali, si possono usare i comandi `\cueDuring` o `\cueDuringWithClef`. Questi sono delle varianti più specializzate di `\quoteDuring`, introdotto in [Citare altre voci], pagina 222, nella sezione precedente.

La sintassi è:

```
\cueDuring nomecitazione #direzione musica
e
\cueDuringWithClef nomecitazione #direzione #chiave musica
```

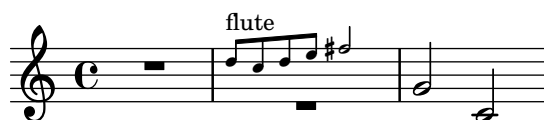
La musica delle misure che corrispondono a *nomecitazione* viene aggiunta in un contesto `CueVoice` e si colloca in simultanea con *musica*, creando quindi una situazione polifonica. La *direzione* prende l'argomento UP o DOWN, e corrisponde alla prima e alla seconda voce rispettivamente, determinando come le notine appaiono in relazione all'altra voce.

```
fluteNotes = \relative {
  r2. c''4 | d8 c d e fis2 | g2 d |
}
```

```
oboeNotes = \relative c'' {
  R1
  <>^\markup \tiny { flute }
  \cueDuring "flute" #UP { R1 }
  g2 c,
}
```

```
\addQuote "flute" { \fluteNotes }
```

```
\new Staff {
  \oboeNotes
}
```



È possibile controllare quali aspetti della musica vengono citati con `\cueDuring` impostando la proprietà `quotedCueEventTypes`. Il suo valore predefinito è ' (note-event rest-event tie-event beam-event tuplet-span-event), che significa che vengono citati solo note, pause, legature di valore, travature e gruppi irregolari, ma non le articolazioni, le indicazioni dinamiche, il testo a margine, etc.

Nota: Quando una voce inizia con `\cueDuring`, come nell'esempio seguente, il contesto `Voice` deve essere dichiarato esplicitamente, altrimenti l'intera espressione musicale appartiene al contesto `CueVoice`.

```
oboeNotes = \relative {
```



```

r2 r8 d''16(\f f e g f a)
g8 g16 g g2.
}
\addQuote "oboe" { \oboeNotes }

\new Voice \relative c'' {
  \set Score.quotedCueEventTypes = #'(note-event rest-event tie-event
                                     beam-event tuplet-span-event
                                     dynamic-event slur-event)

  \cueDuring "oboe" #UP { R1 }
  g2 c,
}

```



Si può usare il comando `\markup` per mostrare il nome dello strumento citato. Se le citazioni in corpo più piccolo richiedono un cambio di chiave, si può fare manualmente, ma anche il ripristino della chiave originale dovrà essere fatto manualmente al termine delle citazioni.

```

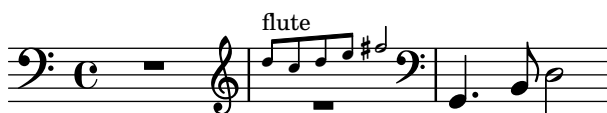
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

bassoonNotes = \relative c {
  \clef bass
  R1
  \clef treble
  <>^\markup \tiny { flute }
  \cueDuring "flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

```



Altrimenti si può usare la funzione `\cueDuringWithClef`. Questo comando prende un ulteriore argomento per specificare il cambio di chiave da usare per le citazioni in corpo più piccolo ma mostrerà automaticamente la chiave originale appena le citazioni sono finite.

```

fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

```

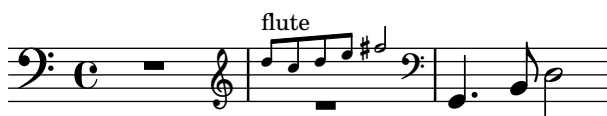
```

bassoonNotes = \relative c {
  \clef bass
  R1
  <>^\markup { \tiny "flute" }
  \cueDuringWithClef "flute" #UP "treble" { R1 }
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

```



Come `\quoteDuring`, `\cueDuring` prende in considerazione la trasposizione degli strumenti. Le citazioni in corpo più piccolo vengono mostrate nelle altezze necessarie allo strumento che riprende la citazione per riprodurre gli stessi suoni dello strumento citato.

Per trasporre le citazioni in corpo più piccolo in modo diverso, si usa `\transposedCueDuring`. Questo comando prende un ulteriore argomento per specificare (in modalità assoluta) l'altezza da usare nella partitura per rappresentare il Do centrale in intonazione reale. È utile nel caso di citazioni da uno strumento che ha un registro completamente diverso.

```

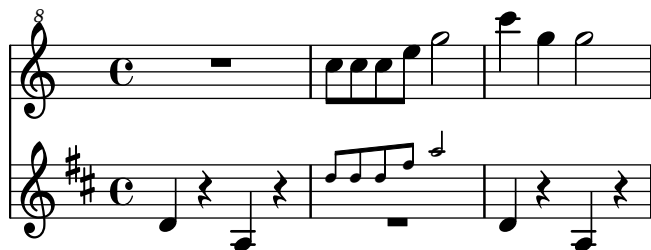
piccoloNotes = \relative {
  \clef "treble^8"
  R1
  c''^8 c c e g2
  c4 g g2
}

bassClarinetNotes = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring "piccolo" #UP d { R1 }
  d4 r a r
}

\addQuote "piccolo" { \piccoloNotes }

<<
  \new Staff \piccoloNotes
  \new Staff \bassClarinetNotes
>>

```



Il comando `\killCues` toglie le citazioni in corpo più piccolo da un'espressione musicale, in modo che la stessa espressione musicale possa essere usata per produrre sia la parte strumentale con le citazioni in corpo più piccolo sia l'intera partitura. Il comando `\killCues` toglie soltanto le note e gli eventi citati da `\cueDuring`. Altre annotazioni relative alle citazioni in corpo più piccolo, come i cambi di chiave e il nome che identifica lo strumento sorgente, possono essere contrassegnate per includerle in modo selettivo nella partitura; si veda [Uso delle etichette], pagina 527.

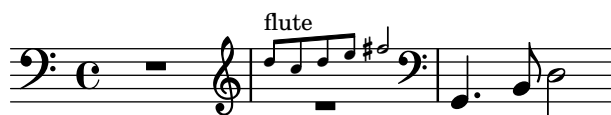
```
fluteNotes = \relative {
  r2. c''4 d8 c d e fis2 g2 d2
}

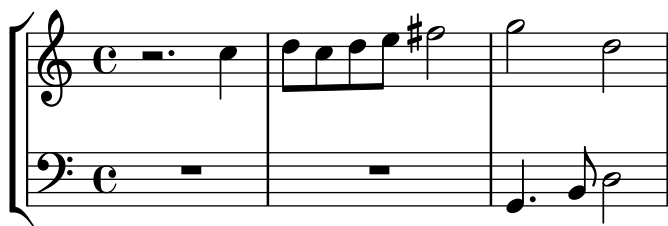
bassoonNotes = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    <>^\markup { \tiny "flute" }
  }
  \cueDuring "flute" #UP { R1 }
  \tag #'part \clef bass
  g4. b8 d2
}

\addQuote "flute" { \fluteNotes }

\new Staff {
  \bassoonNotes
}

\new StaffGroup <<
  \new Staff {
    \fluteNotes
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoonNotes } }
  }
>>
```





Vedi anche

Guida alla notazione: [Citare altre voci], pagina 222, [Trasporto strumentale], pagina 27, [Nomi degli strumenti], pagina 219, [Chiave], pagina 17, [Suggerimenti musicali], pagina 325, [Uso delle etichette], pagina 527.

Frammenti: Sezione “Staff notation” in *Frammenti di codice*.

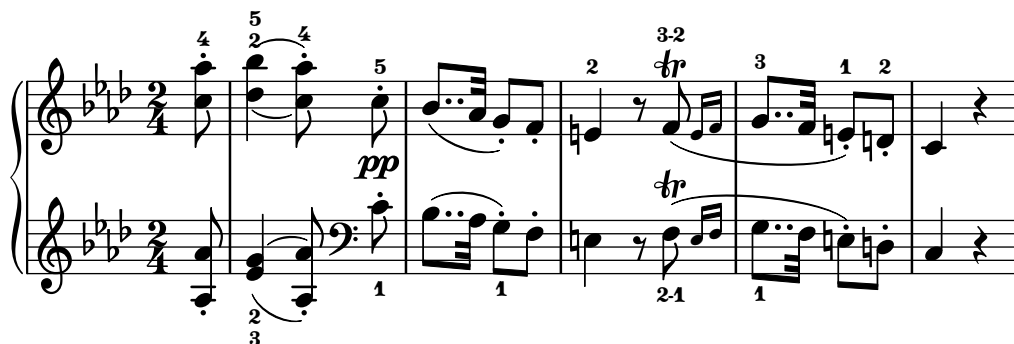
Guida al funzionamento interno: Sezione “CueVoice” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Quando si usa `\cueDuring`, si possono verificare delle collisioni tra le pause nel contesto Voice e quelle in CueVoice. Quando si usa `\cueDuringWithClef` o `\transposedCueDuring`, l'argomento ulteriore richiesto da entrambi deve venire dopo la citazione e la direzione.

Compressione delle misure vuote

1.7 Note editoriali



Questa sezione tratta dei vari modi con cui cambiare l'aspetto delle note e aggiungere un'analisi o un accento didattico.

1.7.1 Interne al rigo

Questa sezione spiega come aggiungere enfasi agli elementi interni al rigo.

Scelta della dimensione del tipo di carattere

Nota:

Per le dimensioni del testo, leggere [Scelta del tipo di carattere e della dimensione], pagina 261.

Per la dimensione del rigo, leggere Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Per le citazioni in corpo piccolo, leggere [Formattazione delle note], pagina 226.

Per i righi ossia, leggere [Righi ossia], pagina 213.

Per modificare la dimensione di un elemento della notazione senza cambiare anche la dimensione del rigo, si può specificare un fattore di ingrandimento col comando `\magnifyMusic`:

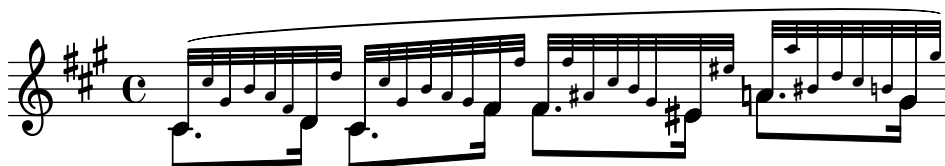
```
\new Staff <<
  \new Voice \relative {
    \voiceOne
    <e' e'>4 <f f'>8. <g g'>16 <f f'>8 <e e'>4 r8
  }
  \new Voice \relative {
    \voiceTwo
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment = #(* 1.2 0.63)
      r32 c'' a c a c a c r c a c a c a c
      r c a c a c a c a c a c a c a c
    }
  }
>>
```



L'`\override` in questo esempio serve a eludere un difetto del programma, spiegato in “Known issues and warnings” alla fine di questa sezione.

Se la testa di una nota di dimensione normale è accorpata con una più piccola, potrebbe essere necessario ripristinare la dimensione della nota più piccola (con `\once \normalsize`) in modo che i gambi e le alterazioni siano allineati correttamente:

```
\new Staff <<
  \key fis \minor
  \mergeDifferentlyDottedOn
  \new Voice \relative {
    \voiceOne
    \magnifyMusic 0.63 {
      \override Score.SpacingSpanner.spacing-increment = #(* 1.2 0.63)
      \once \normalsize cis'32( cis' gis b a fis \once \normalsize d d'
      \once \normalsize cis, cis' gis b a gis \once \normalsize fis fis'
      \once \normalsize fis, fis' ais, cis b gis \once \normalsize eis eis'
      \once \normalsize a, a' bis, d cis b \once \normalsize gis gis')
    }
  }
  \new Voice \relative {
    \voiceTwo
    cis'8. d16 cis8. fis16 fis8. eis16 a8. gis16
  }
>>
```



Il comando `\magnifyMusic` non è adatto per le citazioni in corpo piccolo, gli abbellimenti o i righi ossia, per i quali esistono metodi di inserimento più appropriati. È invece utile quando la dimensione della notazione cambia in una singola parte strumentale su un rigo e quando gli abbellimenti non sono la scelta appropriata, come nei passaggi di tipo cadenza o in casi simili agli esempi precedenti. Impostando il valore di `\magnifyMusic` su 0.63 si duplicano le dimensioni del contesto `CueVoice`.

Nota: Il comando `\magnifyMusic` *non* deve essere usato quando si ridimensiona anche il rigo. Maggiori informazioni in Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Ridimensionare oggetti della formattazione individualmente

Un singolo oggetto della formattazione può essere ridimensionato coi comandi `\tweak` o `\override` per regolare la sua proprietà `font-size`:

```
\relative {
  % ridimensiona una testa di nota
  <' \tweak font-size -4 b e>-5
  % ridimensiona una diteggiatura
  bes-\tweak font-size 0 -3
  % ridimensiona un'alterazione
  \once \override Accidental.font-size = -4 bes!-~
  % ridimensiona un'articolazione
  \once \override Script.font-size = 4 bes!-~
}
```



Il valore predefinito di `font-size` per ogni oggetto della formattazione è elencato nella Guida al funzionamento interno. La proprietà `font-size` può essere impostata solo per quegli oggetti che supportano l'interfaccia di formattazione `font-interface`. Se `font-size` non è specificato nella lista 'Standard settings' dell'oggetto, il suo valore è 0. Si veda Sezione "All layout objects" in *Guida al Funzionamento Interno*.

Capire la proprietà `fontSize`

La proprietà di contesto `fontSize` regola la dimensione relativa di tutti gli elementi della notazione basati su un glifo in un contesto:

```
\relative {
  \time 3/4
  d' '4---5 c8( b a g) |
  \set fontSize = -6
  e'4-- c!8-4( b a g) |
  \set fontSize = 0
  fis4---3 e8( d) fis4 |
  g2.
}
```



Il valore di `fontSize` è un numero che indica la dimensione relativa alla dimensione standard dell'altezza del rigo corrente. Il valore predefinito di `fontSize` è 0; aggiungendo 6 a qualsiasi valore di `fontSize` si raddoppia la dimensione dei glifi e togliendo 6 si dimezza. Ogni punto aumenta la dimensione di circa il 12%.

Dato che le unità logaritmiche della proprietà `font-size` non sono del tutto intuitive, viene fornita per comodità la funzione Scheme `magnification->font-size`. Per esempio, per ridurre la notazione musicale al 75% della dimensione predefinita si usa:

```
\set fontSize = #(magnification->font-size 0.75)
```

La funzione Scheme `magstep` fa l'opposto: converte un valore di `font-size` in un fattore di ingrandimento.

La proprietà `fontSize` avrà effetto soltanto sugli elementi della notazione che sono disegnati con glifi, come le teste di nota, le alterazioni, i segni, etc. Non modificherà la dimensione del rigo stesso né ridimensionerà proporzionalmente gambi, travature o la spaziatura orizzontale. Per ridimensionare gambi, travature e spaziatura orizzontale insieme alla dimensione degli elementi della notazione (senza cambiare la dimensione del rigo), si usa il comando `\magnifyMusic` presentato prima. Per ridimensionare tutto, compreso il rigo, leggere Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Ogni volta che la *proprietà di contesto* `fontSize` è impostata, il suo valore viene aggiunto al valore della *proprietà del grob* `font-size` per i singoli oggetti di formattazione, prima che siano stampati i glifi. Ciò può creare confusione quando si impostano individualmente le proprietà `font-size` mentre è impostato anche `fontSize`:

```
% il valore predefinito di font-size per NoteHead è 0
% il valore predefinito di font-size per Fingering è -5
c''4-3
```

```
\set fontSize = -3
% la dimensione effettiva per NoteHead è ora -3
% la dimensione effettiva per Fingering è ora -8
c''4-3
```

```
\override Fingering.font-size = 0
% la dimensione effettiva per Fingering è ora -3
c''4-3
```



Sono anche disponibili le seguenti scorciatoie:

Comando	Equivalente a	Dimensione relativa
<code>\teeny</code>	<code>\set fontSize = -3</code>	71%
<code>\tiny</code>	<code>\set fontSize = -2</code>	79%
<code>\small</code>	<code>\set fontSize = -1</code>	89%
<code>\normalsize</code>	<code>\set fontSize = 0</code>	100%
<code>\large</code>	<code>\set fontSize = 1</code>	112%
<code>\huge</code>	<code>\set fontSize = 2</code>	126%

```
\relative c'' {
  \teeny
  c4.-> d8---3
  \tiny
  c4.-> d8---3
}
```

```

\small
c4.-> d8---3
\normalsize
c4.-> d8---3
\large
c4.-> d8---3
\huge
c4.-> d8---3
}

```



La modifica della dimensione del tipo di carattere si ottiene ridimensionando la dimensione, tra quelle predefinite, più vicina a quella desiderata. La dimensione standard (per `font-size = 0`) dipende dall'altezza standard del rigo: per un rigo di 20pt, viene scelto un tipo di carattere di 11pt.

Comandi predefiniti

`\magnifyMusic`, `\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`.

Vedi anche

Guida alla notazione: [Scelta del tipo di carattere e della dimensione], pagina 261, Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565, [Formattazione delle notine], pagina 226, [Righi ossia], pagina 213.

File installati: `ly/music-functions-init.ly`, `ly/property-init.ly`.

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “font-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Attualmente ci sono due bug che impediscono una corretta spaziatura orizzontale quando si usa `\magnifyMusic`. C'è un solo modo per eludere questi bug e non funziona in tutte le circostanze. Nell'esempio seguente, sostituire la variabile `mag` con un valore a piacere. Si può provare anche a togliere uno o entrambi i comandi `\newSpacingSection` e/o i comandi `\override` e `\revert`:

```

\magnifyMusic mag {
  \newSpacingSection
  \override Score.SpacingSpanner.spacing-increment = #(* 1.2 mag)
  [music]
  \newSpacingSection
  \revert Score.SpacingSpanner.spacing-increment
}

```

Indicazioni di diteggiatura

Le indicazioni di diteggiatura si inseriscono con ‘`nota-numero`’:

```

\relative { c''4-1 d-2 f-4 e-3 }

```



Si può usare il testo incluso dentro `\markup` o tra virgolette per indicare un cambio di dito.

```
\relative {
  c' '4-1 d-2 f\finger \markup \tied-lyric "4~3" c\finger "2 - 3"
}
```



Si può aggiungere la diteggiatura del pollice per indicare che una nota deve essere suonata col pollice (ad esempio, nella musica per violoncello).

```
\relative { <a'_\thumb a'-3>2 <b_\thumb b'-3> }
```



È possibile indicare la diteggiatura di ogni singola nota di un accordo specificandola dopo ciascuna altezza.

```
\relative {
  <c'-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
}
```



Le indicazioni di diteggiatura possono essere poste sopra o sotto il rigo, come è spiegato in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

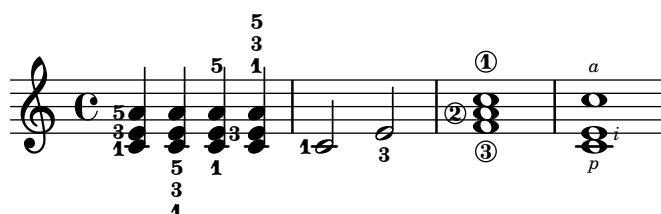
Frammenti di codice selezionati

Controllare il posizionamento delle diteggiature di un accordo

Il posizionamento dei numeri della diteggiatura può essere regolato in modo preciso. Perché l'orientamento funzioni, occorre usare il costrutto per gli accordi `<>` anche per le note singole. Si può impostare in modo simile l'orientamento dei numeri di corda e delle diteggiature della mano destra.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
```

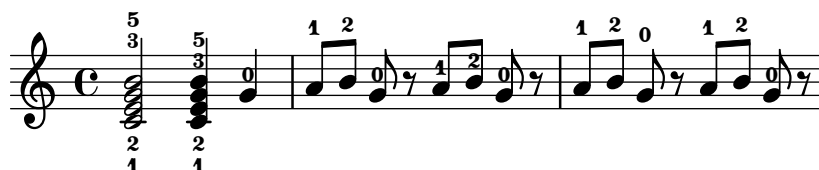
```
<e-3>2
\set stringNumberOrientations = #'(up left down)
<f\3 a\2 c\1>1
\set strokeFingerOrientations = #'(down right up)
<c\rightHandFinger 1 e\rightHandFinger 2 c\rightHandFinger 4 >
```



Far sì che la diteggiatura appaia dentro il rigo

Per impostazione predefinita, le diteggiature orientate verticalmente sono poste fuori dal rigo; questo comportamento tuttavia può essere disabilitato. Occorre fare attenzione alle situazioni in cui le diteggiature e i gambi sono rivolti nella stessa direzione: normalmente le diteggiature evitano soltanto i gambi con travature. Questa impostazione predefinita può essere cambiata in modo da evitare tutti i gambi oppure nessuno. L'esempio seguente mostra queste due opzioni, così come tornare al comportamento predefinito.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Vedi anche

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “FingeringEvent” in *Guida al Funzionamento Interno*, Sezione “fingering-event” in *Guida al Funzionamento Interno*, Sezione “Fingering_engraver” in *Guida al Funzionamento Interno*, Sezione “New_fingering_engraver” in *Guida al Funzionamento Interno*, Sezione “Fingering” in *Guida al Funzionamento Interno*.

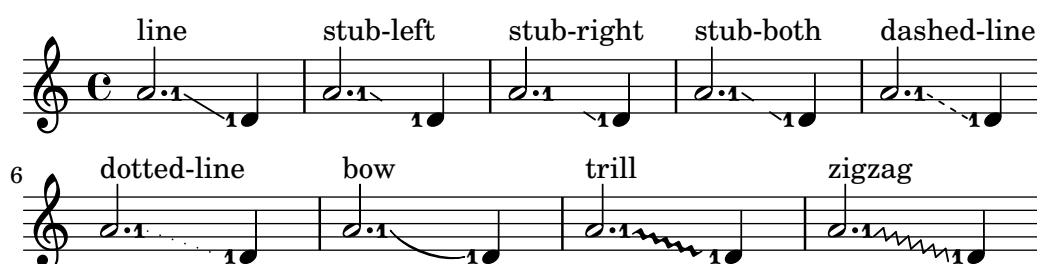
Dita che scivolano

Negli strumenti a corda un dito che scivola viene indicato solitamente da una linea che connette lo stesso dito da usare per note suonate in posizioni diverse sulla stessa corda. Questa linea inizia

inserendo `\glide` prima di una diteggiatura (Fingering) e termina quando si riusa lo stesso dito. La linea può apparire in vari stili.

```
mus = {
  \set fingeringOrientations = #'(right)
  <a'\glide-1>2.
  \set fingeringOrientations = #'(left)
  <d'-1>4
}

{
  <>^"line"
  \mus
  <>^"stub-left"
  \override FingerGlideSpanner.style = #'stub-left
  \mus
  <>^"stub-right"
  \override FingerGlideSpanner.style = #'stub-right
  \mus
  <>^"stub-both"
  \override FingerGlideSpanner.style = #'stub-both
  \mus
  <>^"dashed-line"
  \override FingerGlideSpanner.style = #'dashed-line
  \mus
  \break
  <>^"dotted-line"
  \override FingerGlideSpanner.style = #'dotted-line
  \mus
  <>^"bow"
  \override FingerGlideSpanner.style = #'bow
  \mus
  <>^"trill"
  \override FingerGlideSpanner.style = #'trill
  \mus
  <>^"zigzag"
  \override FingerGlideSpanner.style = #'zigzag
  \mus
}
```



Se style viene impostato su 'bow (arco), la direzione dell'arco può essere regolata attraverso i modificatori della direzione.

```
{
  \override FingerGlideSpanner.style = #'bow
  \set fingeringOrientations = #'(down)
```

```

<b\glide-1>4 <d'-1>
\set fingeringOrientations = #'(up)
<e''\glide-2> <c''-2>

\set fingeringOrientations = #'(down)
<b^\glide-1>4 <d'-1>
\set fingeringOrientations = #'(up)
<e''^\glide-2> <c''-2>

\set fingeringOrientations = #'(down)
<b_\glide-1>4 <d'-1>
\set fingeringOrientations = #'(up)
<e''_\glide-2> <c''-2>
}

```

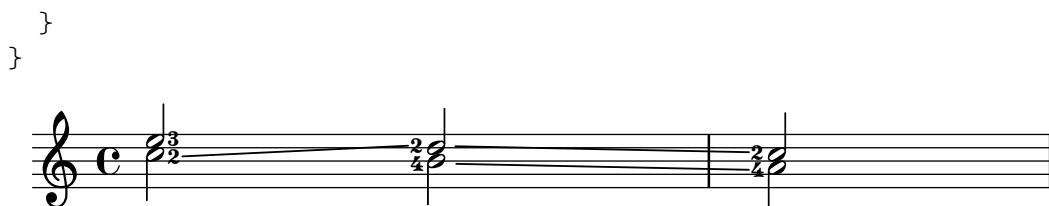


Se si sposta l'incisore `Finger_glide_engraver` nel contesto `Staff`, i grob `Fingering` che appartengono a contesti `Voice` diversi possono essere connessi.

```

\score {
  \new Staff <<
    \new Voice {
      \voiceOne
      \set fingeringOrientations = #'(right)
      <e''-3>2
      \set fingeringOrientations = #'(left)
      <d''-\tweak bound-details.left.padding #2.5 \glide-2>
      <c''-2>
      \bar "||"
    }
    \new Voice {
      \voiceTwo
      \set fingeringOrientations = #'(right)
      <c''\glide-2>
      \set fingeringOrientations = #'(left)
      <b''-\tweak bound-details.left.padding #2.5 \glide-4>
      <a''-4>
    }
  >>
  \layout {
    ragged-right = ##f
    \context {
      \Voice
      \remove "Finger_glide_engraver"
    }
    \context {
      \Staff
      \consists "Finger_glide_engraver"
    }
  }
}

```



Vedi anche

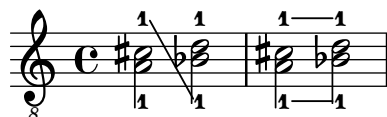
Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Guida al funzionamento interno: Sezione “FingerGlideEvent” in *Guida al Funzionamento Interno*, Sezione “finger-glide-event” in *Guida al Funzionamento Interno*, Sezione “Finger_glide_engraver” in *Guida al Funzionamento Interno*, Sezione “finger-glide-interface” in *Guida al Funzionamento Interno*, Sezione “FingerGlideSpanner” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Molteplici legature con lo stesso dito non sono supportate e producono risultati imprevisti. Un espediente consiste nell’usare dita diverse e `\tweak text`.

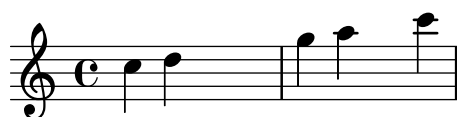
```
{
  \clef "G_8"
  <a\glide-1 cis'\glide-1>2 <bes-1 d'-1>
  <a\glide-1 cis'\glide-\tweak text "1"-2>2
  <bes-1 d'-\tweak text "1"-2>
}
```



Note nascoste

Le note nascoste (o invisibili o trasparenti) possono essere utili nella preparazione di esercizi di teoria e composizione.

```
\relative {
  c' '4 d
  \hideNotes
  e4 f
  \unHideNotes
  g a
  \hideNotes
  b
  \unHideNotes
  c
}
```



Questo comando rende invisibili le teste, i gambi e le code delle note, e le pause. Le travature sono invisibili se iniziano su una nota nascosta. Mentre gli oggetti attaccati a note invisibili sono comunque visibili.

```
\relative c' ' {
```

```
e8(\p f g a)--
\hideNotes
e8(\p f g a)--
}
```



Comandi predefiniti

`\hideNotes`, `\unHideNotes`.

Vedi anche

Manuale d'apprendimento: Sezione "Visibilità e colore degli oggetti" in *Manuale di Apprendimento*.

Guida alla notazione: [Pause invisibili], pagina 65, Sezione 5.4.7 [Visibilità degli oggetti], pagina 652, [Nascondere i righi], pagina 217.

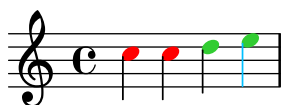
Frammenti: Sezione "Editorial annotations" in *Frammenti di codice*.

Guida al funzionamento interno: Sezione "Note-spacing-engraver" in *Guida al Funzionamento Interno*, Sezione "NoteSpacing" in *Guida al Funzionamento Interno*.

Colorare gli oggetti

Si possono assegnare dei colori a ciascun oggetto. I nomi dei colori validi sono elencati nell'Sezione A.7 [Elenco dei colori], pagina 690.

```
\override NoteHead.color = #red
c''4 c''
\override NoteHead.color = #(x11-color 'LimeGreen)
d''
\override Stem.color = "deepskyblue"
e''
```



Oltre a un insieme limitato di colori semplici disponibili come variabili predefinite (vedi "Colori normali" in Sezione A.7 [Elenco dei colori], pagina 690), si può inserire qualsiasi colore come una stringa. Questa stringa può essere un nome colore definito in stile CSS (<https://www.w3.org/Style/CSS/>) oppure un codice colore esadecimale preceduto dal carattere # (*dentro le virgolette doppie*):

```
\override NoteHead.color = "lightsalmon"
\override Flag.color = "#E30074"
\override Beam.color = "#5e45ad"
\override Rest.color = "#3058"
g'8 \huge r4 a'16 f'
```



Il codice colore può avere un canale alfa per la semitrasparenza usando un codice a otto caratteri `"#RRGGBBAA"` o la sua forma abbreviata `"#RGBA"`.

In un modo diverso, si può accedere all'intera gamma di colori definita per X11 (https://en.wikipedia.org/wiki/X11_color_names) con la funzione Scheme `x11-color`. La funzione prende un argomento, che può essere un simbolo, come `'DarkSeaGreen4`, o una stringa, come `"DarkSeaGreen4"`. La prima forma è più veloce da scrivere e un po' più efficiente. Tuttavia, la seconda forma permette di specificare colori X11 anche attraverso più parole: in questo esempio, `"dark sea green 4"`.

La funzione `x11-color`, se non riesce a comprendere il parametro, restituisce il colore nero.

```
\new Staff \with {
  instrumentName = \markup {
    \with-color #(x11-color 'red) "Clarinet"
  }
}
\relative c'' {
  \override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
  gis8 a
  \override Beam.color = #(x11-color "medium turquoise")
  gis a
  \override Accidental.color = #(x11-color 'DarkRed)
  gis a
  \override NoteHead.color = #(x11-color "LimeGreen")
  gis a
  % this is deliberate nonsense; note that the stems remain black
  \override Stem.color = #(x11-color 'Boggle)
  b2 cis
}
```



I colori RGB esatti si specificano con la funzione Scheme `rgb-color`. Questa funzione prende tre argomenti, usati rispettivamente per i canali *red* (rosso), *green* (verde) e *blue* (blu), e un numero *alfa* opzionale per la semitrasparenza. (Tutti i valori devono essere numeri compresi tra 0 e 1.) La semitrasparenza è supportata nell'output SVG, mentre per l'output PS e PDF è supportata solo se si usa una versione di Ghostscript uguale o superiore a 9.53. Nel frammento seguente la chiave del rigo appare semitrasparente se elaborato secondo le condizioni menzionate sopra.

```
\new Staff \with {
  instrumentName = \markup {
    \with-color #(x11-color 'red) "Clarinetto"
  }
  \override Clef.color = #(rgb-color 0 0 0 0.5)
}
\relative c'' {
  \override Staff.StaffSymbol.color = #(x11-color 'SlateBlue2)
  \override Stem.color = #(rgb-color 0 0 0)
  gis8 a
  \override Stem.color = #(rgb-color 1 1 1)
  gis8 a
  \override Stem.color = #(rgb-color 0 0 0.5)
  gis4 a
}
```



Vedi anche

Guida alla notazione: Sezione A.7 [Elenco dei colori], pagina 690, Sezione 5.3.4 [Il comando `\tweak`], pagina 632.

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Problemi noti e avvertimenti

Un colore X11 non ha necessariamente la stessa identica tonalità di un normale colore dal nome simile.

Non tutti i colori X11 sono distinguibili in un browser web. Per esempio, un browser potrebbe non mostrare alcuna differenza tra LimeGreen e ForestGreen. Per il web si consiglia di usare i colori CSS, come è spiegato in Sezione A.7 [Elenco dei colori], pagina 690.

Le note in un accordo non possono essere colorate separatamente con un `\override`; al suo posto si usa `\tweak` o l'equivalente `\single\override`, vedi Sezione 5.3.4 [Il comando `\tweak`], pagina 632.

Parentesi

Gli oggetti possono essere messi tra parentesi se si usa il comando `\parenthesize` prima dell'evento musicale. Se precede un accordo, viene messa tra parentesi ogni nota dell'accordo. Si possono mettere tra parentesi anche singole note di un accordo.

```
\relative {
  c' '2 \parenthesize d
  c2 \parenthesize <c e g>
  c2 <c \parenthesize e g>
}
```



Si possono mettere tra parentesi anche oggetti diversi dalle note. Per le articolazioni è necessario usare un trattino prima del comando `\parenthesize`.

```
\relative {
  c' '2-\parenthesize -. d
  c2 \parenthesize r
}
```



Quando `\parenthesize` si applica a un accordo, tutte le sue note vengono racchiuse tra parentesi.



Per racchiudere tra parentesi solo alcune note di un accordo, usare un costrutto per la musica parallela `<< ... >>`.



Una forma alternativa del comando `\parenthesize` prevede un percorso al grob: `\parenthesize NomeContesto.NomeGrob` o semplicemente `\parenthesize NomeGrob` (l'ultimo sottintende il contesto al livello più basso, normalmente `Voice`). Bisogna inserirlo prima del momento musicale, come un `\once \override`. Questa forma permette di racchiudere tra parentesi grob che sono causati solo indirettamente dagli eventi.

```
\new Staff \relative <<
{
  \parenthesize NoteHead
  c'1
}
\new CueVoice {
  s2
  \voiceOne
  \once \override Staff.Parentheses.font-size = 3
  \parenthesize Staff.CueClef
  \cueClef treble
  e'8 f a g
}
>>
```



Vedi anche

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Parenthesis engraver” in *Guida al Funzionamento Interno*, Sezione “Parentheses” in *Guida al Funzionamento Interno*, Sezione “parentheses-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Attualmente, la proprietà `font-size` del grob `Parentheses` deve essere modificata manualmente per ottenere parentesi dalle dimensioni corrette per gli accordi e alcuni altri oggetti.

Gambi

Per ogni nota viene creato automaticamente un oggetto `Stem` (gambo). Vale anche per le semi-brevi e le pause, anche se i loro gambi sono resi invisibili.

I gambi si possono posizionare sopra o sotto, vedi Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Comandi predefiniti

`\stemUp`, `\stemDown`, `\stemNeutral`.

Frammenti di codice selezionati

Direzione predefinita dei gambi sulla linea centrale del rigo

La direzione predefinita dei gambi sulla linea centrale del rigo si imposta con la proprietà `neutral-direction` dell'oggetto `Stem`.

```
\relative c' ' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```



Cambiare automaticamente la direzione del gambo della nota centrale in base alla melodia

LilyPond può modificare la direzione del gambo della nota centrale di un rigo in modo che segua la melodia: occorre aggiungere l'incisore `Melody_engraver` al contesto `Voice`.

La proprietà di contesto `suspendMelodyDecisions` può essere usata per disattivare questo comportamento localmente.

```
\relative c' ' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a b g f b g |
  \unset suspendMelodyDecisions
  c b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```



Vedi anche

Guida alla notazione: Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Stem_engraver” in *Guida al Funzionamento Interno*, Sezione “Stem” in *Guida al Funzionamento Interno*, Sezione “stem-interface” in *Guida al Funzionamento Interno*.

1.7.2 Esterne al rigo

Questa sezione spiega come dare risalto agli elementi nel rigo attraverso delle note esterne al rigo.

Nomi delle note

È possibile mostrare i nomi delle note come testo usando il contesto `NoteNames`. Se usato in contemporanea con un rigo normale, si può sincronizzare ogni nota col suo nome, posto sopra o sotto il rigo.

```
\language "italiano"
melody = \relative do'' {
  fad2 si,8 dod re mi fad4. re8 fad2
}

<<
  \new NoteNames { \melody }
  \new Staff { \key si \minor \melody }
  \new NoteNames {
    \set printNotesLanguage = "deutsch"
    \set printAccidentalNames = ##f
    \melody
  }
>>
```



I nomi delle note appaiono nella stessa lingua usata per inserirle, ma la proprietà `printNotesLanguage` permette di selezionare qualsiasi altra lingua disponibile (vedi [Nomi delle note in altre lingue], pagina 8). Le alterazioni possono essere mostrate oppure no attraverso la proprietà `printAccidentalNames`.

Impostando questa proprietà su un simbolo e `printOctaveNames` su `##t`, si possono ottenere nomi delle note che assomigliano molto alla sintassi di LilyPond. Se si desidera un risultato più generico, si possono avere anche i nomi “scientifici” delle ottave.

```
melody = \relative c'' {
  fis2 b,8 cis d e fis4. d8 fis2
}

<<
  \new NoteNames {
    \set printOctaveNames = ##t
    \set printAccidentalNames = #'lily
    \melody
  }
  \new Staff { \key b \minor \melody }
  \new NoteNames {
    \set printOctaveNames = #'scientific
    \melody
  }
>>
```

>>



La proprietà `noteNameSeparator` definisce come debbano apparire gli accordi. Si possono definire altre funzioni di formattazione come `noteNameFunction`; tale funzione si aspetta come argomenti un'altezza e un contesto, anche se uno di questi può essere ignorato.

```
somechords = \relative c' {
  <b d fis>2 <b cis e g> <b d fis> q
}

<<
\new NoteNames {
  \set noteNameSeparator = "+"
  \somechords
}
\new Staff { \key b \minor \somechords }
\new NoteNames {
  \set noteNameFunction =
    #(lambda (pitch ctx)
      (accidental->text-markup
        (ly:pitch-alteration pitch)))
  \somechords
}
>>
```



Vedi anche

Guida alla notazione: [Nomi delle note in altre lingue], pagina 8.

Guida al funzionamento interno: Sezione “NoteName” in *Guida al Funzionamento Interno*, Sezione “NoteNames” in *Guida al Funzionamento Interno*, Sezione “Note_name_engraver” in *Guida al Funzionamento Interno*.

Nuvoletta di aiuto

Si possono contrassegnare e nominare gli elementi della notazione tramite una nuvoletta quadrata. La sua funzione principale è spiegare la notazione.

```
\relative c'' {
  \new Voice \with { \consists Balloon_engraver }
  {
    \balloonGrobText #'Stem #'(3 . 4) \markup { "Sono un gambo" }
    a8
    \balloonGrobText #'Rest #'(-4 . -4) \markup { "Sono una pausa" }
    r
  }
}
```

```

    <c, g'\balloonText #'(-2 . -2) \markup { "Sono la testa di una nota" } c>2.
  }
}

```



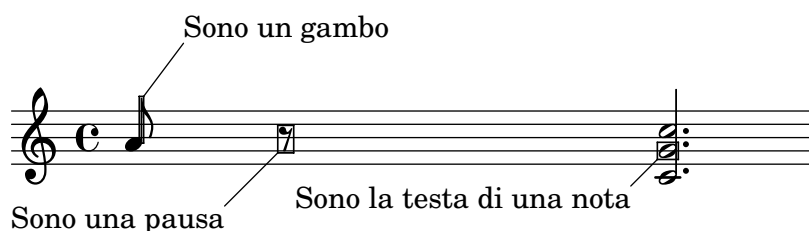
Ci sono due funzioni musicali, `balloonGrobText` e `balloonText`; la prima si usa nella forma `\once \override` per attaccare del testo a un qualsiasi oggetto grafico (grob), mentre la seconda viene usata come `\tweak`, solitamente all'interno degli accordi, per attaccare del testo a una singola nota.

Il testo nella nuvoletta influenza la spaziatura delle note, ma è possibile modificare questo comportamento:

```

\relative c'' {
  \new Voice \with { \consists Balloon_engraver }
  {
    \balloonGrobText #'Stem #'(3 . 4) \markup { "Sono un gambo" }
    a8
    \balloonGrobText #'Rest #'(-4 . -4) \markup { "Sono una pausa" }
    r
    \balloonLengthOn
    <c, g'\balloonText #'(-2 . -2) \markup { "Sono la testa di una nota" } c>2.
  }
}

```



Comandi predefiniti

`\balloonLengthOn`, `\balloonLengthOff`.

Vedi anche

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Balloon_engraver” in *Guida al Funzionamento Interno*, Sezione “BalloonText” in *Guida al Funzionamento Interno*, Sezione “balloon-interface” in *Guida al Funzionamento Interno*.

Linee della griglia

Si possono disegnare delle linee verticali tra i righi sincronizzate con le note.

Si deve usare l'incisore `Grid_point_engraver` per creare le estremità delle linee, mentre l'incisore `Grid_line_span_engraver` serve a disegnare le linee. Per impostazione predefinita, le linee della griglia sono allineate orizzontalmente sotto e sul lato sinistro delle teste di nota. Le

linee si estendono a partire dalle linee centrali di ciascun rigo. `gridInterval` deve specificare la durata che separa le linee.

```
\layout {
  \context {
    \Staff
    \consists Grid_point_engraver
    gridInterval = #1/4
  }
  \context {
    \Score
    \consists Grid_line_span_engraver
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative {
      \stemUp
      c' '4. d8 e8 f g4
    }
    \new Staff \relative {
      \clef bass
      \stemDown
      c4 g' f e
    }
  >>
}
```



Frammenti di codice selezionati

Modificare l'aspetto delle linee della griglia

L'aspetto delle linee della griglia può essere modificato sovrascrivendo alcune delle loro proprietà.

```
\new ChoirStaff <<
  \new Staff {
    \relative c'' {
      \stemUp
      c' '4. d8 e8 f g4
    }
  }
  \new Staff {
    \relative c {
      % this moves them up one staff space from the default position
      \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
      \stemDown
    }
  }
```

```

\clef bass
\once \override Score.GridLine.thickness = 5.0
c4
\once \override Score.GridLine.thickness = 1.0
g'4
\once \override Score.GridLine.thickness = 3.0
f4
\once \override Score.GridLine.thickness = 5.0
e4
}
}
>>

\layout {
  \context {
    \Staff
    % set up grids
    \consists "Grid_point_engraver"
    % set the grid interval to one quarter note
    gridInterval = #1/4
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
    % this moves them to the right half a staff space
    \override NoteColumn.X-offset = -0.5
  }
}

```



Vedi anche

Frammenti: Sezione “Editorial annotations” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Grid_line_span_engraver” in *Guida al Funzionamento Interno*, Sezione “Grid_point_engraver” in *Guida al Funzionamento Interno*, Sezione “GridLine” in *Guida al Funzionamento Interno*, Sezione “GridPoint” in *Guida al Funzionamento Interno*, Sezione “grid-line-interface” in *Guida al Funzionamento Interno*, Sezione “grid-point-interface” in *Guida al Funzionamento Interno*.

Parentesi analitiche

Nell’analisi musicale si usano le parentesi per indicare la struttura dei brani musicali. Sono supportate delle semplici parentesi orizzontali.

```

\layout {
  \context {

```

```

\Voice
\consists Horizontal_bracket_engraver
}
}
\relative {
  c''2\startGroup
  d\stopGroup
}

```



Le parentesi analitiche si possono annidare.

```

\layout {
  \context {
    \Voice
    \consists Horizontal_bracket_engraver
  }
}
\relative {
  c''4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}

```



Frammenti di codice selezionati

Parentesi analitiche sopra il rigo

Delle semplici parentesi analitiche orizzontali vengono aggiunte, per impostazione predefinita, sotto il rigo. L'esempio seguente mostra un modo per posizionarle sopra il rigo.

```

\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}

```

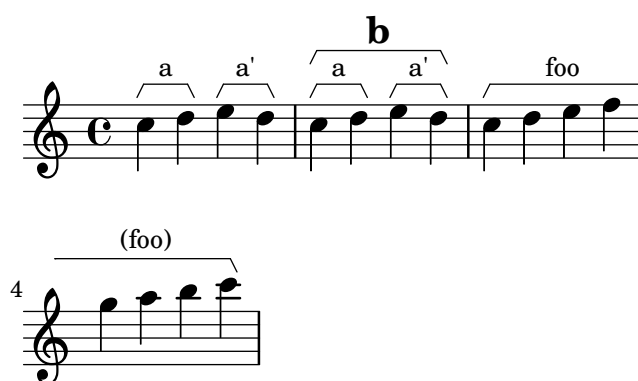


Parentesi analitiche con etichette

Si può aggiungere del testo alle parentesi analitiche tramite la proprietà `text` del grob `HorizontalBracketText`. L'aggiunta di vari frammenti di testo alle parentesi che iniziano nello stesso momento musicale richiede l'uso del comando `\tweak`. Dopo un'interruzione di linea il testo viene messo tra parentesi.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
  c''\startGroup d''\stopGroup
  \once\override HorizontalBracketText.text = "a'"
  e''\startGroup d''\stopGroup |
  c''-\tweak outside-staff-priority #801
    \tweak HorizontalBracketText.text
      \markup \bold \huge "b" \startGroup
    -\tweak HorizontalBracketText.text "a" \startGroup
  d''\stopGroup
  e''-\tweak HorizontalBracketText.text "a'" \startGroup
  d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
  d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Vedi anche

Guida al funzionamento interno: Sezione “Horizontal_bracket_engraver” in *Guida al Funzionamento Interno*, Sezione “HorizontalBracket” in *Guida al Funzionamento Interno*, Sezione “horizontal-bracket-interface” in *Guida al Funzionamento Interno*, Sezione “HorizontalBracketText” in *Guida al Funzionamento Interno*, Sezione “horizontal-bracket-text-interface” in *Guida al Funzionamento Interno*, Sezione “Measure_spanner_engraver” in *Guida al Funzionamento Interno*, Sezione “MeasureSpanner” in *Guida al Funzionamento Interno*, Sezione “measure-spanner-interface” in *Guida al Funzionamento Interno*, Sezione “Staff” in *Guida al Funzionamento Interno*.

1.8 Testo

Moderato cantabile molto espressivo

Questa sezione spiega come includere del testo (con vari tipi di formattazione) nelle partiture musicali.

Alcuni elementi testuali che non sono trattati qui sono discussi in altre sezioni specifiche: Sezione 2.1 [Musica vocale], pagina 280, Sezione 3.2 [Titoli e intestazioni], pagina 499.

1.8.1 Inserimento del testo

Questa sezione presenta vari modi di aggiungere del testo a una partitura.

Nota: Per scrivere caratteri accentati e speciali (come quelli di altre lingue), basta inserire semplicemente i caratteri nel file LilyPond, purché il file sia salvato in formato UTF-8. Ulteriori informazioni in [Codifica del testo], pagina 531.

Scritte

Si possono aggiungere a una partitura delle semplici indicazioni con del “testo tra virgolette”, come mostrato nell’esempio seguente. Tali indicazioni possono essere posizionate sopra o sotto il rigo, usando la sintassi descritta in Sezione 5.4.2 [Direzione e posizionamento], pagina 643.

```
\relative { a'8^"pizz." g f e a4-"scherz." f }
```

In realtà questa sintassi è una scorciatoia; si può specificare una formattazione del testo più complessa usando in modo esplicito un blocco `\markup`, come è spiegato in Sezione 1.8.2 [Formattazione del testo], pagina 260.

```
\relative {
  a'8^\markup { \italic pizz. } g f e
  a4_\markup { \tiny scherz. \bold molto } f }
```



Le indicazioni testuali, di norma, non influenzano la spaziatura delle note. Ma è possibile far sì che la loro larghezza venga presa in considerazione: nell'esempio seguente la prima stringa di testo non influenza la spaziatura, mentre la seconda sì.

```
\relative {
  a'8^"pizz." g f e
  \textLength0n
  a4_"scherzando" f
}
```



Oltre alle scritte, si possono attaccare alle note anche le articolazioni. Ulteriori informazioni in [Articolazioni e abbellimenti], pagina 129.

Per maggiori informazioni sull'ordinamento relativo delle scritte e delle articolazioni si veda Sezione “Posizionamento degli oggetti” in *Manuale di Apprendimento*.

Comandi predefiniti

`\textLength0n`, `\textLength0ff`.

Vedi anche

Manuale d'apprendimento: Sezione “Posizionamento degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Articolazioni e abbellimenti], pagina 129.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Per verificare che le scritte e il testo vocale siano entro i margini occorrono ulteriori calcoli. Nei casi in cui è richiesta un'esecuzione leggermente più veloce, usare

```
\override Score.PaperColumn.keep-inside-line = ##f
```

Estensori del testo

Alcune indicazioni esecutive, per esempio *rallentando* o *accelerando*, appaiono in forma testuale e vengono estese lungo molteplici note con delle linee punteggiate. Tali oggetti, chiamati “estensori” (spanner), si creano collegando due note con la seguente sintassi:

```
\relative {
```

```
\override TextSpanner.bound-details.left.text = "rit."
b'1\startTextSpan
e,\stopTextSpan
}
```



La stringa testuale da stampare viene impostata attraverso le proprietà dell'oggetto. Per impostazione predefinita, appare in corsivo, ma si può ottenere una formattazione diversa tramite i blocchi `\markup`, come è spiegato in Sezione 1.8.2 [Formattazione del testo], pagina 260.

```
\relative {
\override TextSpanner.bound-details.left.text =
\markup { \upright "rit." }
b'1\startTextSpan c
e,\stopTextSpan
}
```



Lo stile della linea, così come la stringa testuale, può essere definito come una proprietà dell'oggetto. Questa sintassi è descritta in Sezione 5.4.8 [Stili della linea], pagina 658.

Comandi predefiniti

`\textSpannerUp`, `\textSpannerDown`, `\textSpannerNeutral`.

Problemi noti e avvertimenti

LilyPond è capace di gestire un solo estensore del testo per ogni voce.

Frammenti di codice selezionati

Dynamics spanner with custom text

Postfix functions for custom crescendo text spanners. The spanners should start on the first note of the measure. One has to use `-\mycresc`, otherwise the spanner start will rather be assigned to the next note.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
  (make-music 'CrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))

mydecrec =
#(define-music-function (mymarkup) (markup?)
  (make-music 'DecrescendoEvent
    'span-direction START
    'span-type 'text
    'span-text mymarkup))
```

```
\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4 c4\! c4 c4
}
```



Vedi anche

Guida alla notazione: Sezione 5.4.8 [Stili della linea], pagina 658, [Dinamiche], pagina 132, Sezione 1.8.2 [Formattazione del testo], pagina 260.

Frammenti: Sezione “Text” in *Frammenti di codice*, Sezione “Expressive marks” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextSpanner” in *Guida al Funzionamento Interno*.

Indicazioni testuali

Si possono aggiungere vari elementi testuali a una partitura tramite la sintassi descritta in [Segni di chiamata], pagina 119:

```
\relative {
  \mark "Verse"
  c'2 g'
  \bar "||"
  \mark "Chorus"
  g2 c,
  \bar "|."
}
```



Questa sintassi permette di porre del testo sopra una stanghetta; una formattazione del testo più complessa è possibile grazie al blocco `\markup`, come è spiegato in Sezione 1.8.2 [Formattazione del testo], pagina 260:

```
\relative {
  <c' e>1
  \mark \markup { \italic { colla parte } }
  <d f>2 <e g>
  <c f aes>1
}
```



Questa sintassi permette anche di stampare segni speciali, come coda, segno o corona, se si specifica il nome appropriato del simbolo, come è spiegato in [Notazione musicale nel blocco markup], pagina 270:

```
\relative {
  <bes' f>2 <aes d>
  \mark \markup { \musicglyph "scripts.ufermata" }
  <e g>1
}
```



Tali oggetti vengono posizionati soltanto sopra il rigo superiore della partitura; a seconda che siano inseriti alla fine o a metà di una battuta, possono trovarsi sopra la stanghetta o tra le note. Se inserito prima di un'interruzione di linea, l'indicazione apparirà all'inizio della linea successiva.

```
\relative c'' {
  \mark "Allegro"
  c1 c
  \mark "assai" \break
  c c
}
```



Comandi predefiniti

`\markLengthOn`, `\markLengthOff`.

Frammenti di codice selezionati

Stampare le indicazioni su ogni rigo

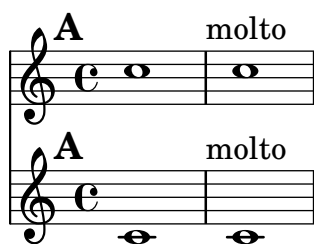
Sebbene le indicazioni testuali siano di norma collocate solo sopra il rigo più alto, è possibile farle apparire su ogni rigo.

```
\score {
  <<
    \new Staff { \mark \default c''1 \textMark "molto" c'' }
    \new Staff { \mark \default c'1 \textMark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove Mark_engraver
      \remove Text_mark_engraver
      \remove Staff_collecting_engraver
    }
  }
}
```

```

    }
    \context {
      \Staff
      \consists Mark_engraver
      \consists Text_mark_engraver
      \consists Staff_collecting_engraver
    }
  }
}

```



Vedi anche

Guida alla notazione: [Segni di chiamata], pagina 119, Sezione 1.8.2 [Formattazione del testo], pagina 260, [Notazione musicale nel blocco markup], pagina 270, Sezione A.8 [Il font Emmen-taler], pagina 693.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextMarkEvent” in *Guida al Funzionamento Interno*, Sezione “Text_mark_engraver” in *Guida al Funzionamento Interno*, Sezione “TextMark” in *Guida al Funzionamento Interno*.

Testo separato

Un blocco `\markup` può esistere di per sé, fuori da qualsiasi blocco `\score`, come un’ “espressione di livello superiore”. Questa sintassi è descritta in Sezione 3.1.5 [Struttura del file], pagina 497.

```

\markup {
  Tomorrow, and tomorrow, and tomorrow...
}

```

Tomorrow, and tomorrow, and tomorrow...

Ciò permette di stampare il testo in modo autonomo dalla musica, ed è utile soprattutto quando il file di input contiene vari brani musicali, come è spiegato in Sezione 3.1.2 [Molteplici partiture in un libro], pagina 494.

```

\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}

```



Tomorrow, and tomorrow, and tomorrow...



Blocchi di testo separati possono essere estesi per molte pagine, rendendo possibile la realizzazione di documenti o libri interamente con LilyPond. Questa funzionalità, e la sintassi specifica che richiede, è descritta in [Testo formattato su più pagine], pagina 273.

Comandi predefiniti

`\markup`, `\markuplist`.

Frammenti di codice selezionati

Testo separato su due colonne

Il testo separato può essere disposto su varie colonne con i comandi di `\markup`:

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
  }
  \hspace #2
  \column \italic {
    \line { O sacred feast }
    \line { in which Christ is received, }
    \line { the memory of His Passion is renewed, }
    \line { the mind is filled with grace, }
    \line { and a pledge of future glory is given to us. }
    \line { Amen. }
  }
}
\hspace #1
}
```

O sacrum convivium
in quo Christus sumitur,
recolitur memoria passionis ejus,
mens impletur gratia,
futurae gloriae nobis pignus datur.
Amen.

O sacred feast
in which Christ is received,
the memory of His Passion is renewed,
the mind is filled with grace,
and a pledge of future glory is given to us.
Amen.

Vedi anche

Guida alla notazione: Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione 3.1.5 [Struttura del file], pagina 497, Sezione 3.1.2 [Molteplici partiture in un libro], pagina 494, [Testo formattato su più pagine], pagina 273.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

1.8.2 Formattazione del testo

Questa sezione presenta la formattazione del testo basilare e quella avanzata, usando la sintassi specifica della modalità `\markup`.

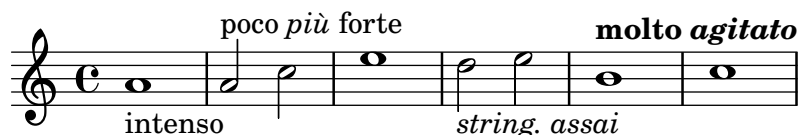
Introduzione al testo a margine

Un blocco `\markup` permette di comporre del testo con un’ampia sintassi chiamata “modalità `markup`”.

La sintassi di `markup` è simile alla solita sintassi di LilyPond: un’espressione `\markup` viene racchiusa tra parentesi graffe `{...}`. Una singola parola viene considerata un’espressione minima, e quindi non è necessario racchiuderla tra parentesi.

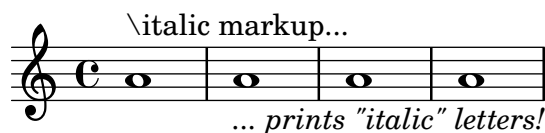
Diversamente dalle indicazioni testuali “tra virgolette”, i blocchi `\markup` possono contenere espressioni o comandi di `markup` annidati, inseriti col carattere di barra inversa `\`. Tali comandi hanno effetto solo sulla prima espressione che segue.

```
\relative {
  a'1-\markup intenso
  a2^\markup { poco \italic più forte }
  c e1
  d2_\markup { \italic "string. assai" }
  e
  b1^\markup { \bold { molto \italic agitato } }
  c
}
```



Un blocco `\markup` può contenere anche stringhe di testo tra virgolette. Tali stringhe vengono trattate come espressioni testuali minime, e quindi qualsiasi comando di `markup` o carattere speciale (come `\` e `#`) apparirà alla lettera senza influenzare la formattazione del testo. Le stesse doppie virgolette possono essere stampate facendole precedere da una barra inversa.

```
\relative {
  a'1^\markup { \italic markup... }
  a_\markup { \italic "... prints \"italic\" letters!" }
  a a
}
```



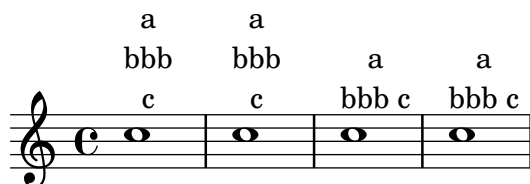
Perché sia trattata come un’espressione distinta, una lista di parole deve essere racchiusa tra virgolette doppie o preceduta da un comando. Il modo in cui le espressioni musicali sono definite influenza il modo in cui saranno sistemate, centrate e allineate; nell’esempio seguente, la seconda espressione di `\markup` viene trattata nello stesso modo della prima:

```
\relative c'' {
  c1^\markup { \center-column { a bbb c } }
```

```

c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { bbb c } } }
c1^\markup { \center-column { a "bbb c" } }
}

```



I markup possono essere salvati in delle variabili, che possono poi essere attaccate direttamente alle note:

```

allegro = \markup { \bold \large Allegro }

{
  d''8.\allegro
  d'16 d'4 r2
}

```



Una lista completa dei comandi specifici di `\markup` si trova in Sezione A.12 [Comandi per *markup*], pagina 713.

Vedi anche

Guida alla notazione: Sezione A.12 [Comandi per *markup*], pagina 713.

Frammenti: Sezione “Text” in *Frammenti di codice*.

File installati: `scm/markup.scm`.

Problemi noti e avvertimenti

I messaggi di errore di sintassi relativi alla modalità `markup` possono essere poco chiari.

Scelta del tipo di carattere e della dimensione

La modalità `markup` permette di cambiare il tipo di carattere:

```

\relative {
  d''1^\markup {
    \bold { Più mosso }
    \italic { non troppo \underline Vivo }
  }
  r2 r4 r8
  d,\markup { \italic quasi \smallCaps Tromba }
  f1 d2 r
}

```



Si può modificare la dimensione del tipo di carattere, rispetto alla dimensione globale del rigo, in vari modi.

Si può impostare su una dimensione predefinita.

```
\relative b' {
  b1_\markup { \huge Sinfonia }
  b1^\markup { \teeny da }
  b1-\markup { \normalsize camera }
}
```



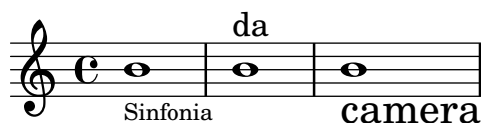
oppure in modo proporzionale rispetto al valore precedente.

```
\relative b' {
  b1_\markup { \larger Sinfonia }
  b1^\markup { \smaller da }
  b1-\markup { \magnify #0.6 camera }
}
```



Può essere aumentata o diminuita rispetto al valore impostato per la dimensione globale del rigo.

```
\relative b' {
  b1_\markup { \fontsize #-2 Sinfonia }
  b1^\markup { \fontsize #1 da }
  b1-\markup { \fontsize #3 camera }
}
```



Si può impostare anche su una dimensione fissa (in punti), indipendentemente dalla dimensione globale del rigo.

```
\relative b' {
  b1_\markup { \abs-fontsize #20 Sinfonia }
  b1^\markup { \abs-fontsize #8 da }
  b1-\markup { \abs-fontsize #14 camera }
}
```



Se il testo contiene degli spazi, è meglio racchiuderlo tutto tra virgolette, in modo che la dimensione di ciascun spazio sia adatta alla dimensione degli altri caratteri.

```
\markup \fontsize #6 \bold { Sinfonia da camera }
```

```
\markup \fontsize #6 \bold { "Sinfonia da camera" }
```

Sinfonia da camera

Sinfonia da camera

È possibile stampare il testo come pedice o apice. Per impostazione predefinita, questo appaiono in corpo più piccolo, ma si può usare anche un corpo normale:

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement }
    \sub { (part two) } }
}
```

1st movement
1st movement_(part two)

La modalità di markup fornisce un modo semplice per scegliere famiglie di caratteri diverse. Se non specificato altrimenti, viene scelto automaticamente il carattere tipografico con grazie (il tipo romano); nell'ultima linea dell'esempio seguente non c'è differenza tra la prima e la seconda parola.

```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \serif Valentine and Proteus. }
  }
}
```

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Alcune di queste famiglie di caratteri, usate per elementi specifici come i numeri o le dinamiche, non forniscono tutti i caratteri, come accennato in [Nuove indicazioni dinamiche], pagina 140, e [Indicazioni di ripetizione manuali], pagina 169.

Se usati all'interno di una parola, alcuni comandi che cambiano il tipo di carattere o la formattazione potrebbero produrre uno spazio vuoto indesiderato. Si può facilmente risolvere concatenando insieme gli elementi testuali:

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
    }
  }
}
```

```

        \italic { con dolce espressione }
    }
}

```

1st movement

p, con dolce espressione

Una lista completa dei comandi per cambiare il tipo di carattere o per usare tipi di carattere personalizzati si trova in Sezione A.12.1 [Font markup], pagina 713.

È possibile anche definire i propri gruppi di tipi di carattere, come è spiegato in Sezione 1.8.3 [Tipi di carattere], pagina 273.

Comandi predefiniti

\teeny, \tiny, \small, \normalsize, \large, \huge, \smaller, \larger.

Vedi anche

Guida alla notazione: Sezione A.12.1 [Font markup], pagina 713, [Nuove indicazioni dinamiche], pagina 140, [Indicazioni di ripetizione manuali], pagina 169, Sezione 1.8.3 [Tipi di carattere], pagina 273.

File installati: scm/define-markup-commands.scm.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

L’uso dei comandi di dimensionamento dei caratteri \teeny, \tiny, \small, \normalsize, \large e \huge produce una spaziatura della linea imprevedibile rispetto all’uso di \fontsize.

Allineamento del testo

Questa sottosezione spiega come posizionare il testo nella modalità markup. Gli oggetti markup possono anche essere spostati interamente tramite la sintassi descritta in Sezione “Spostare gli oggetti” in *Manuale di Apprendimento*.

Gli oggetti di markup possono essere allineati in vari modi. Per impostazione predefinita, l’indicazione testuale è allineata rispetto al suo margine sinistro: nell’esempio seguente, non c’è differenza tra il primo e il secondo markup.

```

\relative {
  d''1-\markup { poco }
  f
  d-\markup { \left-align poco }
  f
  d-\markup { \center-align { poco } }
  f
  d-\markup { \right-align poco }
}

```



L’allineamento orizzontale può essere ritoccato usando un valore numerico:

```

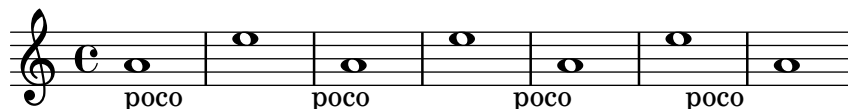
\relative {

```

```

a'1-\markup { \halign #-1 poco }
e'
a,-\markup { \halign #0 poco }
e'
a,-\markup { \halign #0.5 poco }
e'
a,-\markup { \halign #2 poco }
}

```



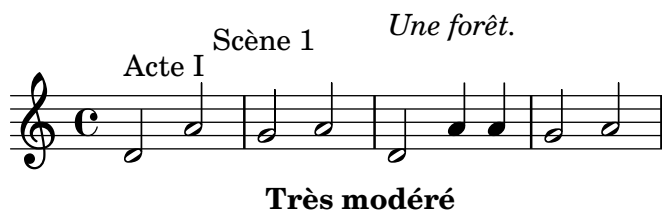
Alcuni oggetti possono avere proprie procedure di allineamento, e dunque non sono influenzate da questi comandi. È possibile spostare tali oggetti di markup tutti insieme, come mostrato ad esempio in [Indicazioni testuali], pagina 256.

L'allineamento verticale è un po' più complesso. Come si è detto prima, gli oggetti di markup possono essere spostati tutti insieme; tuttavia è anche possibile spostare elementi specifici all'interno di un blocco markup. In questo caso l'elemento da spostare deve essere preceduto da un *punto di riferimento*, che può essere un altro elemento markup o un oggetto invisibile. L'esempio seguente illustra queste due possibilità; l'ultimo markup in questo esempio non ha un punto di riferimento e di conseguenza non si muove.

```

\relative {
  d'2^\markup {
    Acte I
    \raise #2 { Scène 1 }
  }
  a'
  g_\markup {
    \null
    \lower #4 \bold { Très modéré }
  }
  a
  d,^\markup {
    \raise #4 \italic { Une forêt. }
  }
  a'4 a g2 a
}

```



Alcuni comandi possono cambiare l'allineamento sia orizzontale che verticale degli oggetti testuali in modalità markup. Qualsiasi oggetto interessato da questi comandi deve essere preceduto da un punto di riferimento:

```

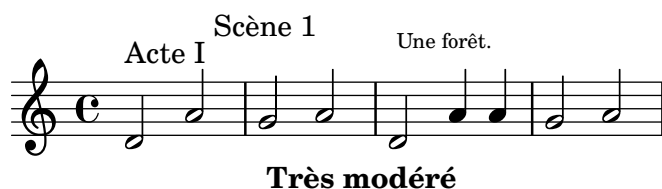
\relative {
  d'2^\markup {
    Acte I
    \translate #'(-1 . 2) "Scène 1"
  }
}

```

```

}
a'
g_\markup {
  \null
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,^\markup {
  \null
  \translate-scaled #'(-1 . 2) \teeny "Une forêt."
}
a'4 a g2 a
}

```



Un oggetto markup può includere varie linee di testo. Nell'esempio seguente, ogni elemento o espressione viene posizionato sulla sua linea, allineato a sinistra o centrato:

```

\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}

```

a	a
b c	b c
d e f	d e f

Analogamente, una lista di elementi o espressioni può essere distesa per riempire l'intera larghezza orizzontale della linea (se c'è un solo elemento, verrà centrato sulla pagina). Queste espressioni possono a loro volta includere del testo multilinea o una qualsiasi altra espressione di markup:

```

\markup {
  \fill-line {
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
  }
  \line { Sir Arthur Sullivan }
}

```

```

    }
  }
  \markup {
    \fill-line { 1885 }
  }

```

William S. Gilbert

THE MIKADO
or
THE TOWN OF TITIPU
1885

Sir Arthur Sullivan

Indicazioni testuali lunghe possono andare a capo automaticamente in base alla larghezza della linea specificata. Possono essere allineate a sinistra o giustificate, come mostra l'esempio seguente.

```

\markup {
  \column {
    \line \smallCaps { La vida breve }
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada. Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada. Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}

```

LA VIDA BREVE

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Una lista completa dei comandi di allineamento del testo si trova in Sezione A.12.2 [Markup for text alignment], pagina 725.

Vedi anche

Manuale d'apprendimento: Sezione “Spostare gli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione A.12.2 [Markup for text alignment], pagina 725, [Indicazioni testuali], pagina 256.

File installati: scm/define-markup-commands.scm.

Frammenti: Sezione “Text” in *Frammenti di codice*.

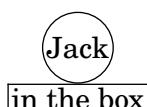
Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Notazione grafica nel blocco markup

Si possono aggiungere vari oggetti grafici a una partitura attraverso i comandi di markup.

Alcuni comandi di markup consentono di decorare gli elementi testuali con degli elementi grafici, come è illustrato nell'esempio seguente.

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```



Erik Satie [1866 - 1925]

Prelude

Alcuni comandi possono richiedere un aumento del padding intorno al testo; per farlo si usano dei comandi di markup, descritti in modo esaustivo in Sezione A.12.2 [Markup for text alignment], pagina 725.

```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
```

```

\box \pad-to-box #'(-5 . 20) #'(0 . 5)
  \bold "Largo to Presto"
}
\pad-around #3
  "String quartet keeps very even time,
  Flute quartet keeps very uneven time."
}

```

Charles Ives (1874 - 1954)

THE UNANSWERED QUESTION

A Cosmic Landscape

Largo to Presto

String quartet keeps very even time, Flute quartet keeps very uneven time.

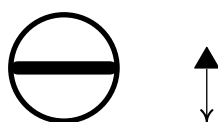
Si possono produrre altri elementi grafici o simboli che non richiedono alcun testo. Come con qualsiasi espressione di markup, tali oggetti possono essere combinati.

```

\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}

```



Le funzionalità grafiche avanzate comprendono la possibilità di includere file di immagini convertite nel formato Encapsulated PostScript (*eps*), oppure di inserire la grafica direttamente nel file di input, usando del codice PostScript nativo. In tal caso, può essere utile specificare esplicitamente la dimensione del disegno, come è mostrato sotto:

```

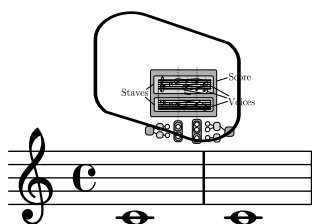
c'1~\markup {
  \combine
    \epsfile #X #10 "./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript "
      -2 3 translate
      2.7 2 scale

```

```

newpath
2 -1 moveto
4 -2 4 1 1 arct
4 2 3 3 1 arct
0 4 0 3 1 arct
0 0 1 -1 1 arct
closepath
stroke"
}
c'

```



Una lista completa dei comandi specifici per la grafica si trova in Sezione A.12.3 [Graphical markup], pagina 744.

Vedi anche

Guida alla notazione: Sezione A.12.2 [Markup for text alignment], pagina 725, Sezione 5.4.4 [Dimensioni], pagina 645, Sezione 1.7 [Note editoriali], pagina 231, Sezione A.12.3 [Graphical markup], pagina 744.

File installati: scm/define-markup-commands.scm, scm/stencil.scm.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Notazione musicale nel blocco markup

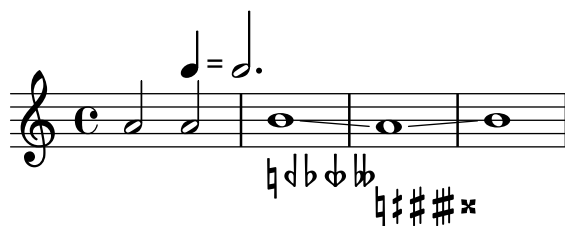
Si possono aggiungere vari elementi della notazione musicale dentro un oggetto markup.

Per le note e le alterazioni esistono dei comandi markup appositi:

```

a'2 a'^\markup {
  \note {4} #1
  =
  \note-by-number #1 #1 #1.5
}
b'1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a'1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b'

```



Anche altri oggetti della notazione possono essere stampati in modalità markup:

```
\relative {
  g1 bes
  ees\finger \markup \tied-lyric "4~1"
  fis_\markup { \dynamic rf }
  bes^\markup {
    \beam #8 #0.1 #0.5
  }
  cis
  d-\markup {
    \markalphabet #8
    \markletter #8
  }
}
```



Più in generale, qualsiasi simbolo musicale disponibile può essere incluso separatamente in un oggetto markup, come è illustrato sotto. Una lista completa di questi simboli e dei loro nomi si trova in Sezione A.8 [Il font Emmentaler], pagina 693.

```
\relative {
  c''2
  c'\markup { \musicglyph "eight" }
  c,4
  c,8.\markup { \musicglyph "clefs.G_change" }
  c16
  c2^\markup { \musicglyph "timesig.neomensural94" }
}
```



Un altro modo per stampare glifi non testuali è descritto in [Tipi di carattere in dettaglio], pagina 274. È utile per stampare parentesi di varie dimensioni.

La modalità markup supporta anche i diagrammi per strumenti specifici:

```
\relative {
  c''1^\markup {
    \fret-diagram-terse "x;x;o;2;3;2;"
  }
}
```

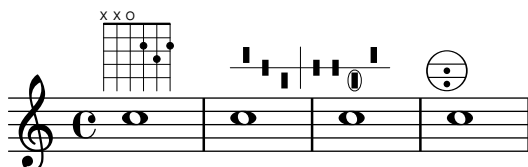
```

}
c^\markup {
  \harp-pedal "^-v|--ov^"
}
c
c^\markup {
  \combine
    \musicglyph "accordion.discant"
  \combine

\raise #0.5 \musicglyph "accordion.dot"

\raise #1.5 \musicglyph "accordion.dot"
}
}

```



Questi diagrammi sono documentati in Sezione A.12.6 [Instrument-specific markup], pagina 768.

È possibile annidare perfino un'intera partitura in un oggetto markup:

```

\relative {
  c'4 d^\markup {
    \score {
      \relative { c'4 d e f }
    }
  }
  e f |
  c d e f
}

```



Una lista completa dei comandi relativi alla notazione musicale si trova in Sezione A.12.4 [Markup for music and musical symbols], pagina 755.

Vedi anche

Guida alla notazione: Sezione A.12.4 [Markup for music and musical symbols], pagina 755, Sezione A.8 [Il font Emmentaler], pagina 693, [Tipi di carattere in dettaglio], pagina 274.

File installati: scm/define-markup-commands.scm, scm/fret-diagrams.scm, scm/harp-pedals.scm.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

La spaziatura verticale di un blocco `\score` interno a un oggetto markup è regolata da `baseline-skip`. Le impostazioni in `\paper` vengono ignorate.

Testo formattato su più pagine

Sebbene gli oggetti di markup standard non possano avere interruzioni, una specifica sintassi permette di inserire linee di testo che possono estendersi per varie pagine:

```
\markuplist {
  \justified-lines {
    Un testo molto lungo di linee giustificate.
    ...
  }
  \wordwrap-lines {
    Un altro paragrafo molto lungo.
    ...
  }
  ...
}
```

Un testo molto lungo di linee giustificate. ...

Un altro paragrafo molto lungo. ...

...

Questa sintassi accetta una lista di oggetti di markup, che possono essere

- il risultato di un comando `\markuplist`,
- una lista di markup,
- una lista di `\markuplists`.

Una lista completa dei comandi che si possono usare con `\markuplist` si trova in Sezione A.13 [Comandi per una lista di *markup*], pagina 789.

Vedi anche

Guida alla notazione: Sezione A.13 [Comandi per una lista di *markup*], pagina 789.

Estendere LilyPond: Sezione “New markup list command definition” in *Estendere*.

File installati: `scm/define-markup-commands.scm`.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*.

Comandi predefiniti

`\markuplist`.

1.8.3 Tipi di carattere

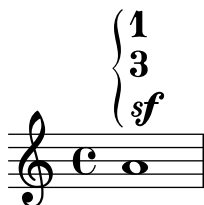
Questa sezione presenta il modo in cui sono gestiti i tipi di carattere e come possono essere modificati nelle partiture.

Tipi di carattere in dettaglio

I tipi di carattere vengono gestiti attraverso varie librerie. FontConfig rileva i tipi di carattere disponibili nel sistema; i tipi selezionati sono riprodotti con Pango.

I tipi di carattere della notazione musicale possono essere descritti come un insieme di glifi specifici, ordinati in varie famiglie. La seguente sintassi permette di usare vari glifi *Feta* di LilyPond direttamente nella modalità markup:

```
a'1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup "brace120"
    \override #'(font-encoding . fetaText)
    \column { 1 3 sf }
    \override #'(font-encoding . fetaMusic)
    \lookup "noteheads.s0petrucci"
  }
}
```



Tuttavia, tutti questi glifi, ad eccezione delle graffe di varie dimensioni contenute in *fetaBraces*, sono già utilizzabili con la sintassi ben più semplice descritta in [Notazione musicale nel blocco markup], pagina 270.

Quando si usano i glifi contenuti in *fetaBraces*, la dimensione della graffa viene specificata dalla parte numerica del nome del glifo, in unità arbitrarie. Può essere specificato qualsiasi numero intero da 0 a 575 compresi, dove 0 corrisponde alla graffa più piccola. Il valore ottimale deve essere determinato per tentativi. Questi glifi sono tutte graffe sinistre; le graffe destre si possono ottenere con la rotazione, vedi Sezione 5.4.9 [Rotazione degli oggetti], pagina 659.

Sono disponibili tre famiglie di font: il font *roman* (con grazie), il font *sans* (senza grazie) e il font monospaziato *typewriter*.

Per il backend *svg*:

Famiglia	Font predefinito
<i>roman</i>	serif
<i>sans</i>	sans-serif
<i>typewriter</i>	monospace

serif, *sans-serif*, e *monospace* sono generic-family (famiglie generiche) nelle specifiche SVG e CSS.

Per gli altri backend:

Famiglia	Font predefinito (alias)	Elenchi di definizione di alias
<i>roman</i>	LilyPond Serif	TeX Gyre Schola, C059, Century SchoolBook URW, Century Schoolbook L, DejaVu Serif, ..., serif
<i>sans</i>	LilyPond Sans Serif	TeX Gyre Heros, Nimbus Sans, Nimbus Sans L, DejaVu Sans, ..., sans-serif
<i>typewriter</i>	LilyPond Monospace	TeX Gyre Cursor, Nimbus Mono PS, Nimbus Mono, Nimbus Mono L, DejaVu Sans Mono, ..., monospace

LilyPond Serif, LilyPond Sans Serif, e LilyPond Monospace sono font alias definiti nel file di configurazione FontConfig specifico per LilyPond 00-lilypond-fonts.conf. Se un carattere non esiste nel primo font elencato, il font successivo dell'elenco verrà usato al suo posto per quel carattere. I dettagli delle definizioni degli alias si trovano nel file 00-lilypond-fonts.conf all'interno della directory di installazione.

Ogni famiglia può avere forme e serie differenti. L'esempio seguente illustra la possibilità di scegliere famiglie, forme, serie e dimensioni alternative. Il valore specificato per font-size è la modifica relativa alla dimensione predefinita.

```
\override Score.RehearsalMark.font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript.font-shape = #'italic
\override Voice.TextScript.font-series = #'bold
d''2.^{\markup "Allegro"}
\override Voice.TextScript.font-size = #-3
c''4^smaller
```



Una sintassi simile si usa nella modalità markup; tuttavia in questo caso è preferibile usare la sintassi più semplice spiegata in [Scelta del tipo di carattere e della dimensione], pagina 261:

```
\markup {
  \column {
    \line {
      \override #'((font-shape . italic) (font-size . 4))
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter)
      {
        \override #'(font-series . bold)
        re
        di
      }
    }
    \override #'(font-family . sans)
    Creta
  }
}
```

Idomeneo,
re di Creta

Quando si usano i font OpenType, sono disponibili le funzionalità dei font. Nota bene: non tutti i font OpenType hanno a disposizione tutte le funzioni. Se si richiede una funzionalità che non esiste nel font scelto, tale funzionalità viene semplicemente ignorata.

```
% Maiuscoletto vero
\markup { Stile normale: Hello HELLO }
\markup { \caps { Maiuscoletto: Hello } }
```



```

\markup { \override #'(font-features . ("smcp"))
          { Maiuscoletto vero: Hello } }

% Stili numerici
\markup { Stile numerico normale: 0123456789 }
\markup { \override #'(font-features . ("onum"))
          { Stile numerico vecchio: 0123456789 } }

% Alternative stilistiche
\markup { \override #'(font-features . ("salt 0"))
          { Alternative stilistiche 0:  $\epsilon\phi\pi\rho\theta$  } }
\markup { \override #'(font-features . ("salt 1"))
          { Alternative stilistiche 1:  $\epsilon\phi\pi\rho\theta$  } }

% Funzionalità multiple
\markup { \override #'(font-features . ("onum" "smcp" "salt 1"))
          { Funzionalità multiple: Hello 0123456789  $\epsilon\phi\pi\rho\theta$  } }

```

Stile normale: Hello HELLO

MAIUSCOLETTTO: HELLO

Maiuscoletto vero: Hello

Stile numerico normale: 0123456789

Stile numerico vecchio: 0123456789

Alternative stilistiche 0: $\epsilon\phi\pi\rho\theta$

Alternative stilistiche 1: $\epsilon\phi\pi\rho\theta$

Funzionalità multiple: Hello 0123456789 $\epsilon\phi\pi\rho\theta$

L'elenco completo delle funzionalità del font OpenType è consultabile qui: <https://www.microsoft.com/typography/otspec/featurelist.htm>

Per identificare le funzionalità dei font OpenType leggere qui: <https://lists.gnu.org/archive/html/lilypond-devel/2017-08/msg00004.html>

Sebbene sia semplice passare a un tipo di carattere preconfigurato, è anche possibile usare altri tipi, come viene spiegato nelle sezioni successive: [Tipi di carattere per singolo oggetto], pagina 277, e [Tipi di carattere per l'intero documento], pagina 277.

Vedi anche

Guida alla notazione: Sezione A.8 [Il font Emmentaler], pagina 693, [Notazione musicale nel blocco markup], pagina 270, Sezione 5.4.9 [Rotazione degli oggetti], pagina 659, [Scelta del tipo di carattere e della dimensione], pagina 261, Sezione A.12.1 [Font markup], pagina 713.

Tipi di carattere per singolo oggetto

Si può usare nella partitura qualsiasi tipo di carattere che sia installato nel sistema operativo e riconosciuto da FontConfig, usando la seguente sintassi:

```
\override Staff.TimeSignature.font-name = "Bitstream Charter"
\override Staff.TimeSignature.font-size = #2
\time 3/4

a'1_\markup {
  \override #'(font-name . "Bitstream Vera Sans,sans-serif, Oblique Bold")
  { Vera Oblique Bold }
}
```



font-name può essere definito da una lista separata da virgola di ‘font’ e una lista separata da spazi di ‘stili’. Se il ‘font’ nella lista è installato e contiene il glifo richiesto, verrà usato, altrimenti sarà usato al suo posto il font *successivo*.

Lanciando lilypond con la seguente opzione si ottiene un elenco di tutti i tipi di carattere disponibili nel sistema operativo:

```
lilypond -dshow-available-fonts
```

Vedi anche

Guida alla notazione: [Tipi di carattere in dettaglio], pagina 274, [Tipi di carattere per l'intero documento], pagina 277.

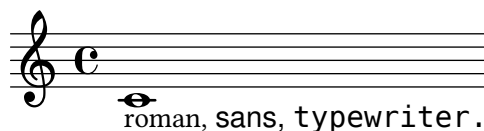
Frammenti: Sezione “Text” in *Frammenti di codice*.

Tipi di carattere per l'intero documento

È possibile modificare i tipi di carattere usati come tipi predefiniti nelle famiglie *roman*, *sans* e *typewriter* specificandoli, in questo ordine, come è mostrato nell'esempio seguente, che ridimensiona automaticamente i caratteri col valore impostato per la dimensione globale del rigo. In modo analogo a [Tipi di carattere per singolo oggetto], pagina 277, si può indicare con una lista separata da virgole di ‘font’. Gli ‘stili’ dei font, invece, non possono essere definiti. I tipi di carattere sono spiegati in [Tipi di carattere in dettaglio], pagina 274.

```
\paper {
  property-defaults.fonts.serif = "Linux Libertine O"
  property-defaults.fonts.sans = "Nimbus Sans, Nimbus Sans L"
  property-defaults.fonts.typewriter = "DejaVu Sans Mono"
}

\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



Nota: `make-pango-font-tree` reimposta i font della notazione sui predefiniti Emmentaler.

La sintassi seguente permette di cambiare font specifici, lasciando gli altri ai valori predefiniti. L'esempio seguente produce lo stesso risultato dell'esempio precedente di `make-pango-font-tree`. Come con `make-pango-font-tree`, è possibile specificare un elenco separato da virgola di "font" per le categorie `roman`, `sans` e `typewriter`. Se non si cambia la dimensione del rigo dal valore predefinito di 20 pt, `#:factor (/ staff-height pt 20)` non è necessario.

```
\paper {
  #(define fonts
    (set-global-fonts
      #:roman "Linux Libertine O"
      #:sans "Nimbus Sans, Nimbus Sans L"
      #:typewriter "DejaVu Sans Mono"
      #:factor (/ staff-height pt 20) ; non necessario se la dimensione del rigo è predefinita
    ))
}
```

È possibile specificare anche i font della notazione. Il prossimo esempio ha anch'esso lo stesso risultato degli esempi precedenti, perché imposta i font della notazione predefiniti. Maggiori informazioni in Sezione 3.4.4 [Cambiare il tipo di carattere della notazione], pagina 535.

```
\paper {
  #(define fonts
    (set-global-fonts
      #:music "emmentaler" ; predefinito
      #:brace "emmentaler" ; predefinito
      #:roman "Linux Libertine O"
      #:sans "Nimbus Sans, Nimbus Sans L"
      #:typewriter "DejaVu Sans Mono"
      #:factor (/ staff-height pt 20) ; non necessario se la dimensione del rigo è predefinita
    ))
}
```

Nota: ogni chiamata della funzione `set-global-fonts` reimposta completamente i font sia della notazione principale che del testo. Se una categoria non è specificata, allora verrà usato il font predefinito per quella categoria. Ogni chiamata di `set-global-fonts` cambia i font di ciascun blocco `\book` che la segue. Se ci sono molteplici blocchi `\book` e si vogliono usare font diversi per ciascuno, basta ripetere `set-global-fonts`, in questo modo:

```
\paper {
  #(define fonts
    (set-global-fonts
      ...
    ))
}
\book {
  ...
}

\paper {
  #(define fonts
    (set-global-fonts
      ...
    ))
}
```

```
}  
\book {  
  ...  
}
```

Vedi anche

Guida alla notazione: [Tipi di carattere in dettaglio], pagina 274, [Tipi di carattere per singolo oggetto], pagina 277, [Scelta del tipo di carattere e della dimensione], pagina 261, Sezione A.12.1 [Font markup], pagina 713, Sezione 3.4.4 [Cambiare il tipo di carattere della notazione], pagina 535.

2 Notazione specialistica

Questo capitolo spiega come creare la notazione musicale per particolari tipi di strumento o per stili specifici.

La musica d'orchestra e di insieme non è trattata in una sezione separata, ma si possono trovare varie nozioni usate comunemente nelle partiture orchestrali in [Riferimenti per opera e musical], pagina 323.

2.1 Musica vocale

Recitativo
Baritono

216 O Freun - - de, nicht die - se Töne!

222 Sondern laßt uns an - - ge -

228 nehmere an - stimmen, und freu -

232 - - - - - *ad libitum* denvollere!

Questa sezione spiega come scrivere la musica vocale e assicurarsi che il testo sia allineato con le note della melodia.

2.1.1 Notazione comune per la musica vocale

Questa sezione tratta le questioni relative alla maggior parte delle tipologie di musica vocale.

Riferimenti per la musica vocale

Questa sezione indica dove trovare informazioni dettagliate sulle questioni comuni a qualsiasi tipo di musica vocale.

- La maggior parte degli stili di musica vocale usa il testo semplice per le parti vocali. Un'introduzione a questo tipo di notazione è disponibile in Sezione "Impostare canzoni semplici" in *Manuale di Apprendimento*.
- La musica vocale richiede solitamente l'uso della modalità markup, sia per il testo musicale che per altri elementi testuali (i nomi dei personaggi, etc.). Questa sintassi è spiegata in [Introduzione al testo a margine], pagina 260.
- L'*Ambitus* può essere aggiunto all'inizio dei righe per la voce, come è spiegato in [Ambitus], pagina 38.
- Le indicazioni dinamiche sono di norma posizionate sotto il rigo, ma nella musica corale sono posizionate solitamente sopra il rigo per evitare il testo, come è spiegato in [Struttura di una partitura corale], pagina 320.

Vedi anche

Glossario musicale: Sezione “ambitus” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Impostare canzoni semplici” in *Manuale di Apprendimento*.

Guida alla notazione: [Introduzione al testo a margine], pagina 260, [Ambitus], pagina 38, [Struttura di una partitura corale], pagina 320.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Inserimento del testo vocale

Il testo vocale viene inserito in una speciale modalità di input, che può essere introdotta dai comandi `\lyricmode`, `\addlyrics` o `\lyricsto`. In questa speciale modalità, l’input `d` non viene analizzato come l’altezza *Re*, ma come una sillaba di una lettera. In altre parole, le sillabe si inseriscono come le note, ma le altezze sono sostituite dal testo.

Per esempio:

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Ci sono due metodi principali per determinare il posizionamento orizzontale delle sillabe. Si può indicare la durata di ogni sillaba esplicitamente, come nell’esempio precedente; oppure si può lasciare che il testo sia allineato automaticamente a una melodia o a un’altra voce del brano, usando `\addlyrics` o `\lyricsto`. Il primo metodo è descritto sotto in [Durate manuali delle sillabe], pagina 287; il secondo in [Durate automatiche delle sillabe], pagina 284.

Una parola o sillaba del testo inizia con un carattere alfabetico (seguito da altri caratteri, vedi dopo) ed è terminata da uno spazio bianco o da un numero. I caratteri successivi al primo nella sillaba possono essere un qualsiasi carattere che non sia un numero o uno spazio bianco.

Dato che qualsiasi carattere che non sia un numero o uno spazio viene considerato come parte della sillaba, una parola è valida anche se finisce con `}`, causando spesso il seguente errore:

```
\lyricmode { lah lah lah }
```

In questo esempio, la parentesi `}` è inclusa nella sillaba finale, dunque la parentesi iniziale non viene chiusa e la compilazione del file probabilmente non riuscirà. Le parentesi devono quindi essere sempre circondate da uno spazio:

```
\lyricmode { lah lah lah }
```

La punteggiatura, i caratteri accentati, quelli di lingue diverse dall’inglese e i caratteri speciali (come il simbolo del cuore o le virgolette oblique) possono essere inseriti direttamente nel file di input, purché sia salvato nella codifica UTF-8. Ulteriori informazioni in Sezione 3.3.3 [Caratteri speciali], pagina 531.

```
\relative { d''8 c16 a bes8 f e' d c4 }
```

```
\addlyrics { „Schad’ um das schö -- ne grü -- ne Band, }
```



Le virgolette normali possono essere usate nel testo vocale, ma devono essere precedute da un carattere di barra inversa e l’intera sillaba deve essere racchiusa tra altre virgolette. Per esempio,

```
\relative { \time 3/4 e'4 e4. e8 d4 e d c2. }
```

```
\addlyrics { "\"I" am so lone -- "ly,\"" said she }
```



La definizione completa di come può iniziare una parola nella modalità testo è un po' più complessa. Può iniziare con un carattere alfabetico, `_`, `?`, `!`, `:`, `'`, i caratteri di controllo `^A` fino a `^F`, `^Q` fino a `^W`, `^Y`, `^Z`, qualsiasi carattere a 8-bit con un codice ASCII superiore a 127 oppure una combinazione a due caratteri di una barra inversa seguita da ```, `'`, `"` o `^`.

Per avere un maggior controllo sull'aspetto del testo si può usare `\markup` all'interno del testo. Per una spiegazione delle molte opzioni disponibili leggere Sezione 1.8.2 [Formattazione del testo], pagina 260.

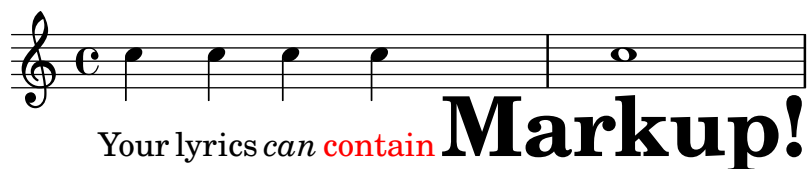
Frammenti di codice selezionati

Formattazione delle sillabe del testo vocale

La modalità markup può essere usata per formattare le singole sillabe del testo vocale.

```
mel = \relative c'' { c4 c c c c1 }
lyr = \lyricmode {
  Your lyrics \markup { \italic can }
  \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

<<
  \new Voice = "melody" \mel
  \new Lyrics \lyricsto "melody" \lyr
>>
```



Vedi anche

Manuale di apprendimento: Sezione “Canzoni” in *Manuale di Apprendimento*.

Guida alla notazione: [Durate automatiche delle sillabe], pagina 284, Sezione 1.8.3 [Tipi di carattere], pagina 273, Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione 5.4.1 [Modalità di inserimento], pagina 641, [Durate manuali delle sillabe], pagina 287, Sezione 3.3.3 [Caratteri speciali], pagina 531.

Guida al funzionamento interno: Sezione “LyricText” in *Guida al Funzionamento Interno*.

Frammenti: Sezione “Text” in *Frammenti di codice*.

Allineamento del testo alla melodia

Il testo vocale viene interpretato in `\lyricmode` e stampato in un contesto Lyrics, vedi Sezione 5.1.1 [Tutto sui contesti], pagina 605.

```
\new Lyrics \lyricmode { ... }
```

Due varianti di `\lyricmode` impostano un contesto associato usato per sincronizzare le sillabe del testo con la musica. Il più comodo `\addlyrics` segue immediatamente il contenuto musicale del contesto della voce con cui deve essere sincronizzato, creando implicitamente un contesto Lyrics. Il più versatile `\lyricsto` richiede sia di specificare il contesto della voce associata con un nome sia di creare esplicitamente un contesto Lyrics che contenga il testo. Maggiori dettagli in [Durate automatiche delle sillabe], pagina 284.

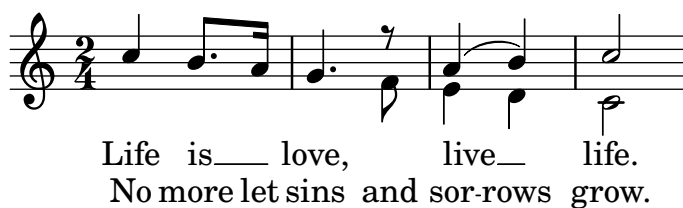
Ci sono due modi per allineare il testo a una melodia:

- Il testo può essere allineato automaticamente in modo che le durate delle sillabe siano prese da un'altra voce o (in circostanze speciali) da una melodia associata, usando `\addlyrics`, `\lyricsto` o impostando la proprietà `associatedVoice`. Ulteriori informazioni in [Durate automatiche delle sillabe], pagina 284.

```
<<
\new Staff <<
  \time 2/4
  \new Voice = "one" \relative {
    \voiceOne
    c''4 b8. a16 g4. r8 a4 ( b ) c2
  }
  \new Voice = "two" \relative {
    \voiceTwo
    s2 s4. f'8 e4 d c2
  }
  }
>>

% prende le durate e l'allineamento dalle note in "one"
\new Lyrics \lyricsto "one" {
  Life is __ _ love, live __ life.
}

% prende le durate e l'allineamento dalle note in "one" inizialmente
% poi passa a "two"
\new Lyrics \lyricsto "one" {
  No more let
  \set associatedVoice = "two" % deve essere impostato prima della sillaba
  sins and sor -- rows grow.
}
>>
```



La prima strofa mostra il modo normale di inserire il testo.

La seconda strofa mostra come cambiare la voce da cui prendere le durate per le sillabe del testo. Ciò è utile se le parole di strofe diverse si distribuiscono lungo le note in modo differente e tutte le durate sono disponibili nei contesti della voce. Ulteriori dettagli in Sezione 2.1.3 [Strofe], pagina 310.

- Il testo può essere allineato indipendentemente dalla durata delle note se le durate delle sillabe vengono specificate esplicitamente e inserite con `\lyricmode`.

```
<<
\new Voice = "one" \relative {
  \time 2/4
  c''4 b8. a16 g4. f8 e4 d c2
}
>>
```



```

% usa la durata esplicita precedente di 2;
\new Lyrics \lyricmode {
  Joy to the earth!
}

% durate esplicite, impostate su un ritmo diverso
\new Lyrics \lyricmode {
  Life4 is love,2. live4 life.2
}
>>

```



La prima strofa non è allineata con le note perché le durate non sono state specificate, e il valore precedente di 2 viene usato per ogni parola.

La seconda strofa mostra come le parole possano essere allineate in modo del tutto indipendente dalle note. Ciò è utile se le parole di strofe diverse si distribuiscono lungo le note in modo differente e le durate necessarie non sono disponibili in un contesto musicale. Ulteriori dettagli in [Durate manuali delle sillabe], pagina 287. Questa tecnica è utile anche quando si imposta un dialogo sopra una musica, come si può vedere negli esempi in [Dialogo sopra la musica], pagina 329.

Vedi anche

Manuale di apprendimento: Sezione “Allineare il testo alla melodia” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.1.1 [Tutto sui contesti], pagina 605, [Durate automatiche delle sillabe], pagina 284, Sezione 2.1.3 [Strofe], pagina 310, [Durate manuali delle sillabe], pagina 287, [Dialogo sopra la musica], pagina 329, [Durate manuali delle sillabe], pagina 287.

Guida al funzionamento interno: Sezione “Lyrics” in *Guida al Funzionamento Interno*.

Durate automatiche delle sillabe

Il testo vocale può essere allineato automaticamente alle note di una melodia in tre modi:

- specificando il nome del contesto Voice contenente la melodia con `\lyricsto`,
- introducendo il testo con `\addlyrics` e inserendolo subito dopo il contesto Voice contenente la melodia,
- impostando la proprietà `associatedVoice`, l’allineamento del testo può essere cambiato su un contesto Voice con un nome diverso in qualsiasi momento musicale.

In tutti questi tre metodi è possibile disegnare dei trattini tra le sillabe di una parola e delle linee di estensione oltre la fine di una parola. Maggiori dettagli in [Estensori e trattini], pagina 292.

Il contesto Voice contenente la melodia al quale il testo si deve allineare non deve essere “morto”, altrimenti il testo successivo a quel punto verrà perduto. Questo può accadere se ci sono dei momenti in cui quella voce non ha nulla da fare. Metodi per mantenere attivi i contesti sono descritti in Sezione 5.1.3 [Conservazione di un contesto], pagina 611.

Uso di `\lyricsto`

Il testo vocale può essere allineato a una melodia automaticamente specificando il nome del contesto voce con `\lyricsto`:

```
<<
  \new Voice = "melody" \relative {
    a'1 a4. a8 a2
  }
  \new Lyrics \lyricsto "melody" {
    These are the words
  }
>>
```



In questo modo il testo viene allineato alle note del contesto `Voice` con quel nome, che deve già esistere. Ecco perché di solito il contesto `Voice` viene indicato prima, seguito dal contesto `Lyrics`. Il testo segue il comando `\lyricsto`. Il comando `\lyricsto` invoca la modalità testo automaticamente. Per impostazione predefinita, il testo viene posizionato sotto le note. Altri posizionamenti sono descritti in [Posizionamento verticale del testo], pagina 295.

Uso di `\addlyrics`

Il comando `\addlyrics` è solo una comoda scorciatoia da usare per evitare di impostare il testo tramite una struttura più complessa.

```
{ MUSICA }
\addlyrics { TESTO VOCALE }
```

equivale a

```
\new Voice = "blah" { MUSICA }
\new Lyrics \lyricsto "blah" { TESTO VOCALE }
```

Ecco un esempio,

```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
}
```



Si possono aggiungere più strofe aggiungendo più sezioni `\addlyrics`:

```
{
  \time 3/4
  \relative { c'2 e4 g2. }
  \addlyrics { play the game }
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```



Il comando `\addlyrics` non può gestire le impostazioni polifoniche. Inoltre non può essere usato per associare il testo alla voce per intavolatura (TabVoice). In questi casi bisogna usare `\lyricsto`.

Uso di `associatedVoice`

La melodia a cui allineare il testo vocale può essere cambiata impostando la proprietà `associatedVoice`,

```
\set associatedVoice = "lala"
```

Il valore della proprietà (qui: "lala") deve essere il nome di un contesto Voice. Per ragioni tecniche, il comando `\set` deve essere posizionato una sillaba prima di quella alla quale si riferisce il cambio di voce.

Ecco un esempio che ne dimostra l'utilizzo:

```
<<
\new Staff <<
  \time 2/4
  \new Voice = "one" \relative {
    \voiceOne
    c' '4 b8. a16 g4. r8 a4 ( b ) c2
  }
  \new Voice = "two" \relative {
    \voiceTwo
    s2 s4. f'8 e8 d4. c2
  }
>>
% inizialmente prende le note e le durate da "one"
% poi passa a "two"
\new Lyrics \lyricsto "one" {
  No more let
  \set associatedVoice = "two" % must be set one syllable early
  sins and sor -- rows grow.
}
>>
```



Vedi anche

Guida alla notazione: [Estensori e trattini], pagina 292, Sezione 5.1.3 [Conservazione di un contesto], pagina 611, [Posizionamento verticale del testo], pagina 295.

Durate manuali delle sillabe

In alcune musiche vocali complesse, si potrebbe voler posizionare il testo in modo totalmente indipendente dalle note. In tali casi non bisogna usare `\lyricsto` o `\addlyrics` e nemmeno impostare `associatedVoice`. Le sillabe verranno invece inserite come se fossero note, ma col testo al posto delle altezze, e la durata di ogni sillaba sarà indicata esplicitamente dopo la sillaba.

Si possono mostrare le linee tratteggiate tra le sillabe come sempre, mentre le linee di estensione non appaiono se non c'è una voce associata.

Ecco due esempi:

```
<<
\new Voice = "melody" \relative {
  c''2 a f f e e
}
\new Lyrics \lyricmode {
  c4. -- a -- f -- f -- e2. -- e
}
>>
```



```
<<
\new Staff {
  \relative {
    c''2 c2
    d1
  }
}
\new Lyrics {
  \lyricmode {
    I2 like4. my8 cat!1
  }
}
\new Staff {
  \relative {
    c'8 c c c c c c c
    c8 c c c c c c c
  }
}
>>
```



Questa tecnica è utile quando si scrivono dialoghi, vedi [Dialogo sopra la musica], pagina 329. Per cambiare l'allineamento delle sillabe, basta impostare la proprietà `self-alignment-X`:

```
<<
```

```

\new Voice = "melody" \relative {
  \time 3/4
  c'2 e4 g2 f
}
\new Lyrics \lyricmode {
  \override LyricText.self-alignment-X = #LEFT
  play1 a4 game4
}
>>

```



Vedi anche

Guida alla notazione: [Dialogo sopra la musica], pagina 329.

Guida al funzionamento interno: Sezione “Lyrics” in *Guida al Funzionamento Interno*, Sezione “Voice” in *Guida al Funzionamento Interno*.

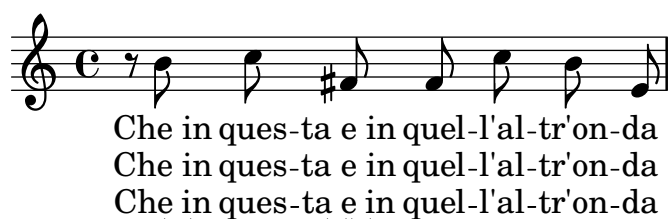
Più sillabe in una nota

Per assegnare più di una sillaba a una singola nota mantenendo uno spazio tra le sillabe, occorre mettere la frase tra virgolette o usare il carattere `_`. Altrimenti si può usare il simbolo tilde (`~`) per ottenere una legatura di valore per il testo.

```

{
  \relative {
    \autoBeamOff
    r8 b' c fis, fis c' b e,
  }
  \addlyrics
  {
    % Assicurarsi che i trattini siano visibili
    \override LyricHyphen.minimum-distance = #1.0
    Che_in ques -- ta_e_in quel -- l'al -- tr'on -- da
  }
  \addlyrics { "Che in" ques -- "ta e in" quel -- l'al -- tr'on -- da }
  \addlyrics { Che~in ques -- ta~e~in quel -- l'al -- tr'on -- da }
}

```



Vedi anche

Guida al funzionamento interno: Sezione “LyricCombineMusic” in *Guida al Funzionamento Interno*.

Più note in una sillaba

Talvolta, in particolare nella musica medievale e barocca, molte note vengono cantate in una sillaba; si chiama melisma (vedi Sezione “melisma” in *Glossario Musicale*). La sillaba di un melisma viene solitamente allineata a sinistra della prima nota del melisma.

Quando un melisma si trova su una sillaba diversa dall’ultima in una parola, quella sillaba di solito viene collegata a quella successiva con una linea tratteggiata. Ciò si indica inserendo un doppio trattino --, subito dopo la sillaba.

Altrimenti, se un melisma si trova sull’ultima o unica sillaba in una parola, solitamente appare una linea di estensione dalla fine della sillaba all’ultima nota del melisma. Ciò si indica inserendo un doppio trattino basso, __, subito dopo la parola.

Esistono cinque modi per indicare i melismi:

- I melismi vengono creati automaticamente sulle note legate insieme:

```
<<
\new Voice = "melody" \relative {
  \time 3/4
  f' '4 g2 ~ |
  4 e2 ~ |
  8
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e __
}
>>
```



- I melismi possono essere creati automaticamente dalla musica inserendo delle legature di portamento sulle note di ogni melisma. Questo è il modo più comune di inserire il testo:

```
<<
\new Voice = "melody" \relative {
  \time 3/4
  f' '4 g8 ( f e f )
  e8 ( d e2 )
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e __
}
>>
```



Attenzione: le legature di frase non causano la creazione di melismi.

- Le note sono considerate un melisma se le loro travature sono disposte manualmente, purché la travatura automatica sia disabilitata. Vedi [Impostare il comportamento delle travature automatiche], pagina 91.

```
<<
```

```

\new Voice = "melody" \relative {
  \time 3/4
  \autoBeamOff
  f''4 g8[ f e f]
  e2.
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e
}
>>

```



Ovviamente ciò non è adatto ai melismi su note più lunghe degli ottavi.

- Un gruppo di note privo di legature sarà trattato come un melisma se sono comprese tra `\melisma` `\melismaEnd`.

```

<<
\new Voice = "melody" \relative {
  \time 3/4
  f''4 g8
  \melisma
  f e f
  \melismaEnd
  e2.
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e
}
>>

```



- Un melisma può essere definito interamente nel testo inserendo un solo trattino basso, `_`, per ogni nota ulteriore da aggiungere al melisma.

```

<<
\new Voice = "melody" \relative {
  \time 3/4
  f''4 g8 f e f
  e8 d e2
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- _ _ _ e _ _ _
}
>>

```



È possibile avere legature di portamento e di valore e travature manuali nella melodia senza che indichino i melismi. Per farlo si imposta `melismaBusyProperties`:

```
<<
\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'()
  c'4 d ( e )
  g8 [ f ] f4 ~ 4
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- e e -- le -- i -- son
}
>>
```



Altre impostazioni di `melismaBusyProperties` possono essere usate per includere o escludere selettivamente legature di valore e di portamento e travature dal rilevamento automatico del melisma; consultare `melismaBusyProperties` in Sezione “Tunable context properties” in *Guida al Funzionamento Interno*.

Altrimenti, se si vuole ignorare tutte le indicazioni di melisma, basta impostare `ignoreMelismata` su vero; vedi [Strofe con ritmi diversi], pagina 312.

Se un melisma è necessario in un passaggio in cui `melismaBusyProperties` è attivo, lo si può indicare inserendo un singolo trattino basso nel testo per ogni nota che debba essere inclusa nel melisma:

```
<<
\new Voice = "melody" \relative {
  \time 3/4
  \set melismaBusyProperties = #'()
  c'4 d ( e )
  g8 [ f ] ~ f4 ~ 4
}
\new Lyrics \lyricsto "melody" {
  Ky -- ri -- _ e _ _ _ _
}
>>
```



Comandi predefiniti

`\autoBeamOff`, `\autoBeamOn`, `\melisma`, `\melismaEnd`.

Vedi anche

Glossario musicale: Sezione “melisma” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Allineare il testo alla melodia” in *Manuale di Apprendimento*.

Guida alla notazione: [Allineamento del testo alla melodia], pagina 282, [Durate automatiche delle sillabe], pagina 284, [Impostare il comportamento delle travature automatiche], pagina 91, [Strofe con ritmi diversi], pagina 312.

Guida al funzionamento interno: Sezione “Tunable context properties” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Le linee di estensione sotto i melismi non vengono creati automaticamente; devono essere inseriti a mano con un doppio trattino basso.

Estensori e trattini

Nell’ultima sillaba di una parola, i melismi sono talvolta indicati con una lunga linea orizzontale che inizia dalla sillaba del melisma e termina in quella successiva. Tale linea viene chiamata linea di estensione e si inserisce con ‘`--`’ (notare gli spazi prima e dopo i due caratteri).

Nota: I melismi appaiono nella partitura come linee di estensione, che si inseriscono con un doppio trattino basso; ma i melismi brevi si possono inserire anche saltando note singole but short melismata can also be entered by skipping individual notes, which are entered as single underscore characters; these do not make an extender line to be typeset by default.

Il trattino centrato si inserisce con ‘`--`’ tra le sillabe di una stessa parola (notare gli spazi prima e dopo i due caratteri). Il trattino sarà centrato tra le sillabe e la sua lunghezza sarà regolata a seconda dello spazio tra le sillabe.

Nelle partiture compresse i trattini possono essere eliminati. Questo comportamento è controllato da due proprietà di LyricHyphen: `minimum-distance` (distanza minima tra le due sillabe) e `minimum-length` (soglia sotto la quale i trattini vengono rimossi).

Per impostazione predefinita, un trattino non viene ripetuto dopo un a capo quando la linea successiva inizia con una nuova sillaba. Per far sì che il trattino appaia in tali situazioni si deve impostare la proprietà `after-line-breaking` su `#t`.

Vedi anche

Guida al funzionamento interno: Sezione “LyricExtender” in *Guida al Funzionamento Interno*, Sezione “LyricHyphen” in *Guida al Funzionamento Interno*.

Cambi graduali di vocale

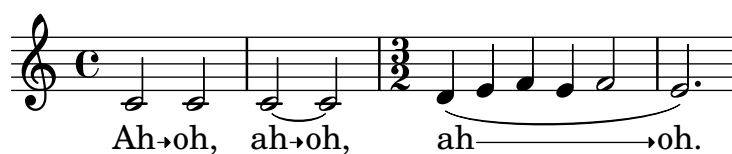
Le transizioni vocali (cambi graduali delle vocali o consonante sostenuta), che possono essere indicate da frecce tra le sillabe, si inseriscono col comando `\vowelTransition` (vedi Gould pp. 452–453). La freccia indica la lunghezza della transizione, e per impostazione predefinita viene sempre disegnata (viene aggiunto dello spazio se necessario). Le legature di valore tra note di altezza invariata o le legature di portamento mostrano che non ci sono nuove articolazioni, nonostante un cambio di vocale. La lunghezza minima delle frecce si può regolare con la proprietà `minimum-length` di `VowelTransition`.

{

```

c'2 c'
\set melismaBusyProperties = #'()
c'2 ~ c'
\time 3/2
d'4( e' f' e' f'2
e'2.)
}
\addlyrics
{
  Ah \vowelTransition oh,
  ah \vowelTransition oh,
  ah \vowelTransition _ _ _ _
  oh.
}

```



Vedi anche

Glossario musicale: Sezione “vowel transition” in *Glossario Musicale*.

Guida al funzionamento interno: Sezione “VowelTransition” in *Guida al Funzionamento Interno*.

2.1.2 Tecniche specifiche per il testo vocale

Lavorare con testo e variabili

Si possono creare delle variabili contenenti il testo vocale, ma questo deve essere inserito in modalità testo vocale:

```

musicOne = \relative {
  c''4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
  <<
    \new Voice = "one" {
      \time 2/4
      \musicOne
    }
    \new Lyrics \lyricsto "one" {
      \verseOne
    }
  >>
}

```



Non è necessario aggiungere le durate se la variabile viene richiamata con `\addlyrics` o `\lyricsto`. Se la musica ha un ordine diverso e più complesso, conviene definire prima le variabili che contengono la musica e il testo, poi impostare la gerarchia di righe e testo, omettendo il testo stesso, e infine aggiungere il testo all'interno di un blocco `\context`. Ciò garantisce che le voci richiamate da `\lyricsto` siano sempre state definite prima. Per esempio:

```
sopranoMusic = \relative { c'4 c c c }
contraltoMusic = \relative { a4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        \sopranoWords
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        \contraltoWords
      }
    }
  }
  >>
}
```



Vedi anche

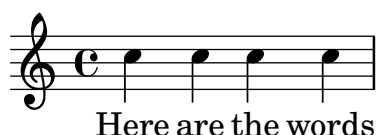
Guida alla notazione: [Posizionamento verticale del testo], pagina 295.

Guida al funzionamento interno: Sezione “LyricCombineMusic” in *Guida al Funzionamento Interno*, Sezione “Lyrics” in *Guida al Funzionamento Interno*.

Posizionamento verticale del testo

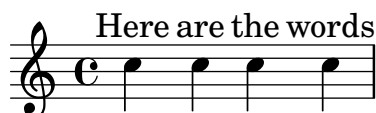
A seconda del tipo di musica, il testo può trovarsi sopra o sotto il rigo oppure tra i righi. Posizionare il testo sotto il rigo associato è il modo più semplice, infatti basta definire il contesto Lyrics sotto il contesto Staff;

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative { c' '4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Per posizionare il testo sopra il rigo ci sono due metodi. Il metodo più semplice (e preferito) consiste nell'usare la stessa sintassi precedente e indicare in modo esplicito la posizione del testo:

```
\score {
  <<
    \new Staff = "staff" {
      \new Voice = "melody" {
        \relative { c' '4 c c c }
      }
    }
    \new Lyrics \with { alignAboveContext = "staff" } {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



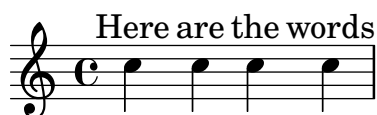
Altrimenti si possono usare due passaggi. Prima si dichiara il contesto Lyrics (privo di contenuto) prima dei contesti Staff e Voice, poi il comando `\lyricsto` viene posizionato dopo la voce a cui si riferisce tramite `\context`. Ecco un esempio:

```
\score {
  <<
    \new Lyrics = "lyrics" \with {
      % il testo sopra un rigo deve avere questo override
```

```

    \override VerticalAxisGroup.staff-affinity = #DOWN
  }
  \new Staff {
    \new Voice = "melody" {
      \relative { c' '4 c c c }
    }
  }
  \context Lyrics = "lyrics" {
    \lyricsto "melody" {
      Here are the words
    }
  }
  >>
}

```



Quando ci sono due voci in righe separate, si può posizionare il testo tra i righe usando uno di questi metodi. Ecco un esempio del secondo metodo:

```

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative { c' '4 c c c }
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos" \with {
      % il testo sopra un rigo deve avere questo override
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "contraltos" {
        \relative { a' 4 a a a }
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        Sop -- ra -- no words
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        Con -- tral -- to words
      }
    }
  }
  >>
}

```



Si possono generare altre combinazioni di testo e righe a partire da questi esempi oppure studiando i modelli nel Manuale di apprendimento, vedi Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.

Vedi anche

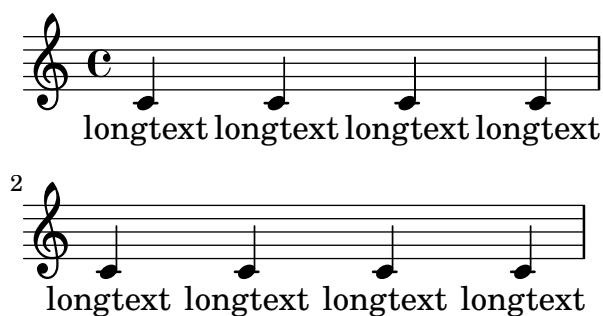
Manuale di apprendimento: Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.1.7 [Ordine di disposizione dei contesti], pagina 623, Sezione 5.1.2 [Creazione e citazione di un contesto], pagina 607.

Posizionamento orizzontale delle sillabe

Per aumentare lo spazio tra le righe del testo, si imposta la proprietà `minimum-distance` di `LyricSpace`.

```
\relative c' {
  c c c c
  \override Lyrics.LyricSpace.minimum-distance = #1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```



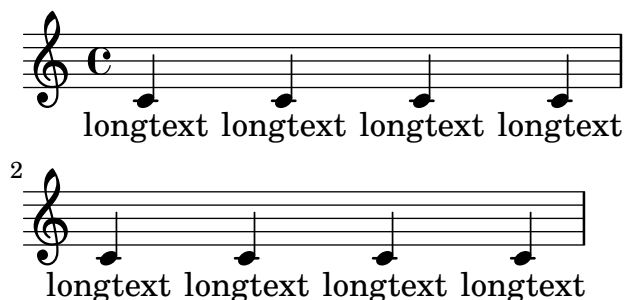
Per applicare questa modifica a tutti i testi della partitura, impostare la proprietà nel blocco `\layout`.

```
\score {
  \relative {
    c' c c c
    c c c c
  }
  \addlyrics {
    longtext longtext longtext longtext
    longtext longtext longtext longtext
  }
}
```

```

}
\layout {
  \context {
    \Lyrics
    \override LyricSpace.minimum-distance = #1.0
  }
}
}

```



Frammenti di codice selezionati

Allineamento del testo vocale

L'allineamento orizzontale del testo vocale si imposta attraverso la proprietà `self-alignment-X` dell'oggetto `LyricText`. #-1 è sinistra, #0 è centro e #1 è destra; si possono usare anche `#LEFT`, `#CENTER` e `#RIGHT`.

```

\layout {
  \ragged-right = ##f
}

\relative c'' {
  c1 c c c
}

\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "centered"
  \once \override LyricText.self-alignment-X = 1
  "right-aligned"
  \once \override LyricText.self-alignment-X = #-1.5
  "very right"
}

```



Problemi noti e avvertimenti

Verificare che le annotazioni testuali e il testo vocale stiano dentro i margini richiede ulteriori calcoli. Se si desidera velocizzare un po' l'elaborazione, tale funzionalità può essere disabilitata:

```

\override Score.PaperColumn.keep-inside-line = ##f

```

Per far sì che il testo eviti anche le stanghette, usare

```
\layout {
  \context {
    \Lyrics
    \consists Bar_engraver
    \consists Separating_line_group_engraver
    \hide BarLine
  }
}
```

Testo e ripetizioni

Ripetizioni semplici

Le ripetizioni in *musica* sono trattate in un'altra sezione: Sezione 1.4 [Ripetizioni], pagina 160. Questa sezione spiega come aggiungere del testo vocale a delle parti musicali ripetute.

Il testo che si riferisce a una sezione musicale ripetuta deve avere lo stesso costrutto di ripetizione della musica, se le parole non sono modificate.

```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Non ri -- petu -- to.
        \repeat volta 2 { Ri -- petu -- to due. }
      }
    }
  >>
}
```



Le parole verranno poi espanse correttamente se le ripetizioni sono ripetute.

```
\score {
  \unfoldRepeats {
    <<
      \new Staff {
        \new Voice = "melody" {
          \relative {
            a'4 a a a
            \repeat volta 2 { b4 b b b }
          }
        }
      }
    }
}
```



```

    }
    \new Lyrics {
      \lyricsto "melody" {
        Non ri -- petu -- to.
        \repeat volta 2 { Ri -- petu -- to due. }
      }
    }
  >>
}

```



Se la sezione ripetuta deve essere ripetuta di nuovo con parole diverse, è sufficiente inserire tutte le parole:

```

\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat unfold 2 { b4 b b b }
        }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Non ri -- petu -- to.
      Parole della prima volta.
      Parole della seconda volta.
    }
  }
  >>
}

```



Quando le parole che si riferiscono a una ripetizione sono diverse, le parole di ogni ripetizione devono essere inserite in contesti Lyrics separati, annidati correttamente in sezioni parallele:

```

\score {
  <<
    \new Staff {

```

```

\new Voice = "melody" {
  \relative {
    a'4 a a a
    \repeat volta 2 { b4 b b b }
  }
}
\new Lyrics \lyricsto "melody" {
  Non ri -- petu -- to.
  <<
{ Parole della prima volta. }
\new Lyrics {
  \set associatedVoice = "melody"
  Parole della seconda volta.
}
  >>
}
>>
}

```



Si possono aggiungere più strofe in modo analogo:

```

\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics \lyricsto "singleVoice" {
      Non ri -- petu -- to.
      <<
{ Parole della prima volta. }
\new Lyrics {
  \set associatedVoice = "singleVoice"
  Parole della seconda volta.
}
\new Lyrics {
  \set associatedVoice = "singleVoice"
  Parole della terza volta.
}
      >>
      La sezione fi -- nale.
    }
  >>
}

```

```

    }
  >>
}

```



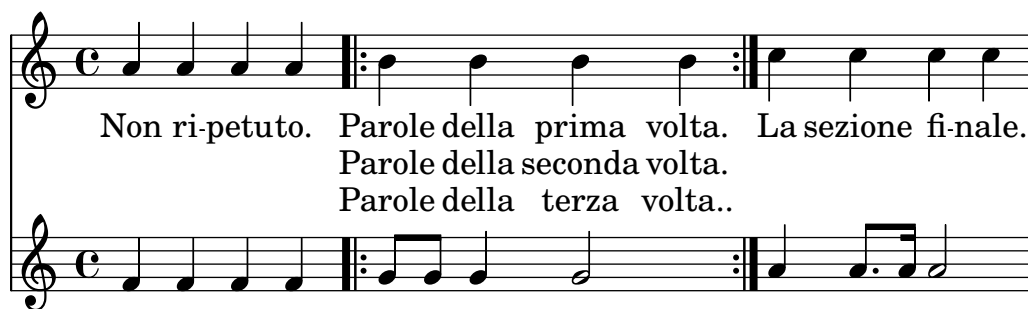
Tuttavia, se questo costrutto si trova all'interno di un contesto multirigo come `ChoirStaff`, il testo della seconda e terza strofa apparirà sotto il rigo inferiore.

Per posizzionarli correttamente usare `alignBelowContext`:

```

\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics = "firstVerse" \lyricsto "melody" {
      Non ri -- petu -- to.
    }
    { Parole della prima volta. }
    \new Lyrics = "secondVerse"
    \with { alignBelowContext = "firstVerse" } {
      \set associatedVoice = "melody"
      Parole della seconda volta.
    }
    \new Lyrics = "thirdVerse"
    \with { alignBelowContext = "secondVerse" } {
      \set associatedVoice = "melody"
      Parole della terza volta..
    }
  }
  >>
  La sezione fi -- nale.
}
\new Voice = "harmony" {
  \relative {
    f'4 f f f \repeat volta 2 { g8 g g4 g2 } a4 a8. a16 a2
  }
}
>>
}

```



Ripetizioni con finali alternativi

Se le parole della sezione ripetuta sono le stesse, e nessuno dei finali alternativi inizia con una pausa, si può usare la stessa identica struttura sia per il testo che per la musica. Ciò comporta il vantaggio che `unfoldRepeats` espanderà correttamente sia la musica che il testo vocale.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          a'4 a a a
          \repeat volta 2 { b4 b }
          \alternative {
            \volta 1 { b b }
            \volta 2 { b c }
          }
        }
      }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      Non ri -- petu -- to.
      \repeat volta 2 { Ripe -- tu -- }
      \alternative {
        \volta 1 { to due. }
        \volta 2 { to due. }
      }
    }
  }
  >>
}
```



Ma quando la sezione ripetuta ha parole diverse o uno dei blocchi `\alternative` inizia con una pausa, non si può usare il costrutto della ripetizione per le parole e bisogna inserire manualmente i comandi `\skip` per le note delle sezioni alternative prive di un testo corrispondente.

Attenzione: non usare il trattino basso, `_`, per saltare le note, perché il trattino basso indica un melisma e fa sì che la sillaba precedente sia allineata a sinistra.

Nota: Il comando `\skip` deve essere seguito da un numero, ma questo numero viene ignorato se nel testo vocale la durata delle sillabe deriva dalla durata delle note in una melodia associata attraverso `\addlyrics` o `\lyricsto`. Ogni `\skip` salta una singola nota di un qualsiasi valore, senza tener conto del valore del numero che segue.

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b }
          \alternative {
            \volta 1 { b b }
            \volta 2 { b c }
          }
        }
        c4 c
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Parole della prima volta.
        \repeat unfold 2 { \skip 1 }
        Termina qui.
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Pa -- role
        \repeat unfold 2 { \skip 1 }
        seconda volta.
      }
    }
  >>
}

```



Quando una nota è legata verso uno o più finali alternativi, si usa una legatura di valore per portare quella nota nel primo finale alternativo e `\repeatTie` per portarla nel secondo e nei successivi. Questa struttura comporta dei problemi di difficile allineamento quando è presente il testo; si può ottenere un risultato più accettabile aumentando la lunghezza delle sezioni alternative in modo che le note legate siano contenute interamente al loro interno.

La legatura crea un melisma nella prima alternativa, ma non nella seconda e nelle successive, dunque per allineare il testo correttamente occorre disabilitare la creazione automatica dei melismi dopo la sezione delle volte e inserire dei salti manuali.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b'4 b ~}
          \alternative {
            \volta 1 { b b }
            \volta 2 { b \repeatTie c }
          }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        \repeat volta 2 { Ecco una __ }
        \alternative {
          \volta 1 { \skip 1 strofa }
          \volta 2 { \skip 1 sec }
        }
      }
      onda strofa.
    }
  >>
}
```



Se `\unfoldRepeats` precede una sezione contenente `\repeatTie`, bisogna togliere `\repeatTie` per evitare che appaiano entrambi i tipi di legatura.

Quando la sezione ripetuta ha parola diverse, non si può mettere il testo in un blocco `\repeat` e bisogna inserire manualmente i comandi `\skip`, come si è visto prima.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative {
          \repeat volta 2 { b'4 b ~}
          \alternative {
            \volta 1 { b b }
            \volta 2 { b \repeatTie c }
          }
        }
      }
    }
  >>
}
```

```

    }
    c4 c
  }
  }
  }
  \new Lyrics {
    \lyricsto "melody" {
Ecco una __ strofa.
\repeat unfold 2 { \skip 1 }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
Eccone un'
\repeat unfold 2 { \skip 1 }
altra da cantare.
    }
  }
  >>
}

```



Se si desidera mostrare gli estensori e i trattini subito prima o dopo un finale alternativo, questi vanno inseriti a mano.

```

\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
\relative {
  \repeat volta 2 { b'4 b ~}
  \alternative {
    \volta 1 { b b }
    \volta 2 { b \repeatTie c }
  }
}
c4 c
}
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
Ecco una __ strofa.
\repeat unfold 2 { \skip 1 }
    }
  }
  \new Lyrics {
    \lyricsto "melody" {

```

```

Ecco "una_"
\skip 1
"_" sec -- onda strofa.
}
}
>>
}

```



Vedi anche

Guida alla notazione: Sezione 5.1.3 [Conservazione di un contesto], pagina 611, Sezione 1.4 [Ripetizioni], pagina 160.

Testi alternati

Quando solo le parole e le durate di due parti differiscono mentre le altezze sono le stesse, conviene disabilitare temporaneamente la rilevazione automatica dei melismi e indicare il melisma nel testo cantato:

```

\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDown
        \slurDashed
        e4 e8 ( e ) c4 c |
        \unset melismaBusyProperties
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "melody" {
      We will _
    }
  >>
}

```



Quando sia la musica che le parole differiscono, è meglio nominare i contesti della voce in cui musica e testo sono diversi e assegnare il testo a quei contesti specifici:

```

\score {

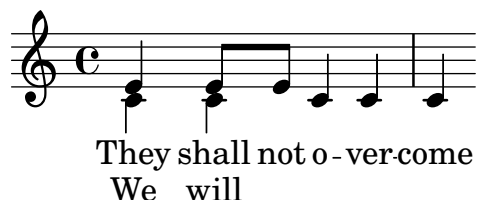
```



```

<<
  \new Voice = "melody" {
    \relative {
<<
  {
    \voiceOne
    e'4 e8 e
  }
  \new Voice = "splitpart" {
    \voiceTwo
    c4 c
  }
>>
\oneVoice
c4 c |
c
  }
}
  \new Lyrics \lyricsto "melody" {
    They shall not o -- ver -- come
  }
  \new Lyrics \lyricsto "splitpart" {
    We will
  }
>>
}

```



Nella musica corale è comune avere una voce divisa in varie misure. Il costrutto `<< {...} \\ {...} >>`, in cui due (o più) espressioni musicali sono separate dalla doppia barra inversa, potrebbe sembrare il modo giusto di impostare le voci divise. Tuttavia questo costrutto assegna **tutte** le espressioni al suo interno a **NUOVI contesti Voice**, dunque *nessun testo cantato* sarà impostato per esse perché il testo si collegherà al contesto della voce originale. Di norma, non è ciò che si desidera. Il costrutto adatto in questa situazione è il passaggio polifonico temporaneo, spiegato nella sezione *Passaggi polifonici temporanei* in [Polifonia su un solo rigo], pagina 184.

Polifonia con testo in comune

Quando due voci contenenti durate diverse condividono lo stesso testo cantato, allineare il testo a una delle voci può creare dei problemi nell'altra voce. Per esempio, il secondo estensore del testo nell'esempio seguente è troppo corto, perché il testo è allineato solo alla voce superiore:

```

soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice = "sopranoVoice" { \voiceOne \soprano }

```

```

\new Voice { \voiceTwo \alto }
\new Lyrics \lyricsto "sopranoVoice" \words
>>

```



Per ottenere il risultato desiderato, si allinea il testo a un nuovo contesto `NullVoice` contenente una giusta combinazione delle due voci. Le note del contesto `NullVoice` non appaiono nello spartito, ma servono soltanto ad allineare il testo nel modo corretto:

```

soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice { \voiceOne \soprano }
  \new Voice { \voiceTwo \alto }
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>

```



Questo metodo si può usare anche con la funzione `\partCombine`, che di per sé non accetta il testo cantato:

```

soprano = \relative { b'8( c d c) d2 }
alto = \relative { g'2 b8( a g a) }
aligner = \relative { b'8( c d c) b( a g a) }
words = \lyricmode { la __ la __ }

\new Staff <<
  \new Voice \partCombine \soprano \alto
  \new NullVoice = "aligner" \aligner
  \new Lyrics \lyricsto "aligner" \words
>>

```



Problemi noti e avvertimenti

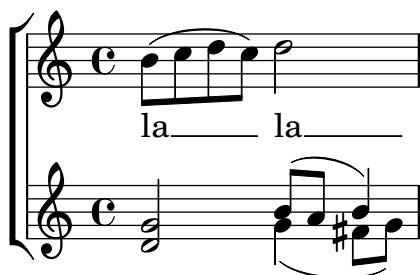
La funzione `\addlyrics` funziona solo con testi vocali collegati a contesti `Voice` e non può essere usata con `NullVoice`.

La funzione `\partCombine` è descritta in [Combinazione automatica delle parti], pagina 194.

Infine, si può usare questo metodo anche quando le voci si trovano su righe diversi e non è limitata a due sole voci:

```
soprano = \relative { b'8( c d c) d2 }
altoOne = \relative { g'2 b8( a b4) }
altoTwo = \relative { d'2 g4( fis8 g) }
aligner = \relative { b'8( c d c) d( d d d) }
words = \lyricmode { la __ la __ }

\new ChoirStaff \with {\accepts NullVoice } <<
  \new Staff <<
    \soprano
    \new NullVoice = "aligner" \aligner
  >>
  \new Lyrics \lyricsto "aligner" \words
  \new Staff \partCombine \altoOne \altoTwo
>>
```

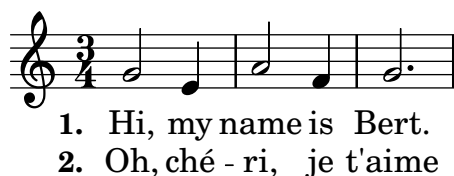


2.1.3 Strofe

Aggiungere i numeri di strofa

I numeri di strofa si aggiungono impostando stanza:

```
\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = "1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = "2. "
  Oh, ché -- ri, je t'aime
}
```



Questi numeri appaiono prima dell'inizio della prima sillaba. Due linee di una strofa possono anche essere raggruppate insieme, per esempio in caso di una ripetizione con testo diverso:

```
leftbrace = \markup {
  \override #'(font-encoding . fetaBraces)
  \lookup "brace80"
}
```

```
stanzaOneOne = \lyricmode {
  \set stanza = \markup {
    \column { \vspace #.33 "1. " }
    \leftbrace
  }
  Child, you're mine and I love you.
  Lend thine ear to what I say.
}
```

```
stanzaOneThree = \lyricmode {
  Child, I have no great -- er joy
  Than to have you walk in truth.
}
```

```
\new Voice {
  \repeat volta 2 {
    c'8 c' c' c' c' c' c'4
    c'8 c' c' c' c' c' c'4
  }
}
\addlyrics { \stanzaOneOne }
\addlyrics { \stanzaOneThree }
```

1. { Child, you're mine and I love you.
Child, I have no greater joy
Lend thine ear to what I say.
Than to have you walk in truth.

Aggiungere le dinamiche alle strofe

Le strofe che hanno un volume diverso possono essere indicate con un segno di dinamica all'inizio di ogni strofa. In LilyPond, tutto ciò che si trova di fronte a una strofa va nell'oggetto StanzaNumber; lo stesso vale per i segni di dinamica. Per ragioni tecniche, bisogna impostare la strofa fuori da `\lyricmode`:

```
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}
```

```
<<
\new Voice = "tune" {
  \time 3/4
```

```

      g'4 c'2
    }
\new Lyrics \lyricsto "tune" \text
>>

```



Aggiungere i nomi dei cantanti alle strofe

Si possono aggiungere anche i nomi dei cantanti. Appariranno all'inizio del rigo, proprio come per i nomi degli strumenti. Si creano impostando `vocalName`. Una versione abbreviata si inserisce con `shortVocalName`.

```

\new Voice \relative {
  \time 3/4 g'2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = "Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = "Ernie "
  Oh, ché -- ri, je t'aime
}

```



Strofe con ritmi diversi

Spesso, strofe diverse di una canzone sono collegate a una melodia in modi leggermente diversi. Tali variazioni possono essere colte con `\lyricsto`.

Ignorare i melismi

Può capitare ad esempio che il testo abbia un melisma in una strofa, ma varie sillabe in un'altra. Una possibile soluzione consiste nel far sì che la voce più veloce ignori il melisma, impostando `ignoreMelismata` nel contesto `Lyrics`.

```

<<
\relative \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f
  c'4
  \slurDotted
  f8.[( g16)]
  a4
}
\new Lyrics \lyricsto "lahlah" {
  più len -- ta
}
\new Lyrics \lyricsto "lahlah" {
  più
}

```

```

\set ignoreMelismata = ##t
velo -- ce
\unset ignoreMelismata
ancora
}
>>

```



Problemi noti e avvertimenti

Diversamente dalla maggior parte dei comandi `\set`, `\set ignoreMelismata` non funziona se preceduto da `\once`. Bisogna usare `\set` e `\unset` per contrassegnare il testo in cui il melisma deve essere ignorato.

Aggiungere le sillabe agli abbellimenti

Per impostazione predefinita, gli abbellimenti (ovvero le note inserite con `\grace`) non sono assegnati alle sillabe quando si usa `\lyricsto`, ma tale comportamento può essere modificato:

```

<<
\new Voice = melody \relative {
  f'4 \appoggiatura a32 b4
  \grace { f16 a16 } b2
  \afterGrace b2 { f16[ a16] }
  \appoggiatura a32 b4
  \acciaccatura a8 b4
}
\new Lyrics
\lyricsto melody {
  normal
  \set includeGraceNotes = ##t
  case,
  gra -- ce case,
  after -- grace case,
  \set ignoreMelismata = ##t
  app. case,
  acc. case.
}
>>

```



Problemi noti e avvertimenti

Come per `associatedVoice`, `includeGraceNotes` deve essere impostato al più tardi una sillaba prima di quella da mettere sotto un abbellimento. In caso di un abbellimento proprio all'inizio di un brano, si consiglia di usare un blocco `\with` o un blocco `\context` all'interno di `\layout`:

```

<<

```

```

\new Voice = melody \relative c' {
  \grace { c16( d e f }
  g1) f
}
\new Lyrics \with { includeGraceNotes = ##t }
\lyricsto melody {
  Ah __ fa
}
>>

```



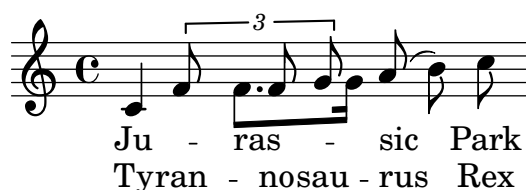
Passare a una melodia alternativa

Sono possibili variazioni più complesse nell'impostare testo e musica. La melodia su cui è impostato il testo può essere modificata all'interno del contesto del testo impostando la proprietà `associatedVoice`:

```

<<
\relative \new Voice = "lahlah" {
  \set Staff.autoBeaming = ##f
  c'4
  <<
    \new Voice = "alternative" {
\voiceOne
\tuplet 3/2 {
  % mostra chiaramente le associazioni.
  \override NoteColumn.force-hshift = #-3
  f8 f g
}
    }
    {
\voiceTwo
f8.[ g16]
\oneVoice
    } >>
    a8( b) c
  }
\new Lyrics \lyricsto "lahlah" {
  Ju -- ras -- sic Park
}
\new Lyrics \lyricsto "lahlah" {
  % Complicato: bisogna impostare associatedVoice
  % una sillaba prima di quella cui si applica!
  \set associatedVoice = "alternative" % si applica a "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = "lahlah" % si applica a "rus"
  sau -- rus Rex
} >>

```



Il testo per la prima strofa viene impostato sulla melodia 'lahlah' nel solito modo, ma la seconda strofa è impostata inizialmente sul contesto lahlah e passa poi alla melodia alternative per le sillabe da 'ran' a 'sau':

```
\set associatedVoice = "alternative" % si applica a "ran"
Ty --
ran --
no --
\set associatedVoice = "lahlah" % si applica a "rus"
sau -- rus Rex
```

In questo esempio `alternative` è il nome del contesto Voice contenente la terzina.

Attenzione al posizionamento del comando `\set associatedVoice`: appare una sillaba troppo presto, ma ciò è corretto.

Nota: Il comando `\set associatedVoice` deve essere inserito una sillaba *prima* di quella in cui deve verificarsi il passaggio alla nuova voce. In altre parole, il passaggio alla voce associata accade una sillaba dopo quella che ci si aspetterebbe. Ciò è dovuto a ragioni tecniche e non è un difetto di LilyPond.

Stampare le strofe alla fine

Talvolta si allinea una sola strofa alla musica e le strofe rimanenti appaiono in forma di versi alla fine del brano. Per ottenere ciò si aggiungono le strofe ulteriori in un blocco `\markup` esterno al blocco della partitura. Esistono vari modi di forzare le interruzioni di linea in un blocco `\markup`. Per inserire una stringa intera si può usare `\string-lines` con `\n` inseriti manualmente oppure interruzioni di linea automatiche così come si scrivono oppure `\wordwrap-string`. Se si usa la formattazione all'interno, è consigliata una combinazione di `\line` e `\column`.

```
melody = \relative {
  e' d c d | e e e e |
  d d e d | c1 |
}

text = \lyricmode {
  \set stanza = "1." Ma- ry had a lit- tle lamb,
  its fleece was white as snow.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}
\markup {
```



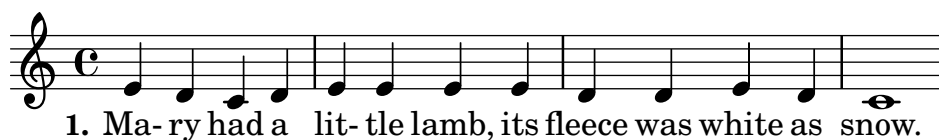
```

\column {
  \string-lines
  "Verse 2. \n Everywhere that Mary went \n The lamb was sure to go."
}
}
\markup {
  \column {
    \string-lines
    "Verse 3.
    All the children laughed and played,
    To see a lamb at school."
  }
}
\markup {
  \column {
    \line \italic { Verse 4. }
    \line { And so the teacher turned it out, }
    \line { But still it lingered near. }
  }
}
\markup {
  \wordwrap-string "
  Verse 5.

  Mary took it home again,

  It was against the rule."
}

```



Verse 2.

Everywhere that Mary went
The lamb was sure to go.

Verse 3.

All the children laughed and played,
To see a lamb at school.

Verse 4.

And so the teacher turned it out,
But still it lingered near.

Verse 5.

Mary took it home again,
It was against the rule.

Stampare le strofe alla fine in molteplici colonne

Quando un brano ha molte strofe, queste sono spesso stampate in molteplici colonne lungo la pagina. Un numero di strofa rientrato spesso introduce ciascuna strofa. L'esempio seguente mostra come riprodurre questo output in LilyPond.

```

melody = \relative {
  c'4 c c c | d d d d
}

text = \lyricmode {
  \set stanza = "1."
  This is verse one.
  It has two lines.
}

\score {
  <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
}

\markup {
  \fill-line {
    % sposta la colonna in avanti rispetto al margine sinistro;
    % può essere tolto se lo spazio nella pagina è stretto
    \hspace #0.1
    \column {
      \line { \bold "2."
        \column {
          "This is verse two."
          "It has two lines."
        }
      }
      % aggiunge spazio verticale tra le strofe
      \combine \null \vspace #0.1
      \line { \bold "3."
        \column {
          "This is verse three."
          "It has two lines."
        }
      }
    }
  }
  % aggiunge spazio orizzontale tra le colonne
  \hspace #0.1
  \column {
    \line { \bold "4."
      \column {
        "This is verse four."
        "It has two lines."
      }
    }
  }
  % aggiunge spazio verticale tra le strofe
  \combine \null \vspace #0.1
  \line { \bold "5."
    \column {

```

```

        "This is verse five."
        "It has two lines."
    }
}
}
% dà ulteriore spazio sul margine destro
% può essere tolto se lo spazio nella pagina è stretto
\hspace #0.1
}
}

```



1. This is verse one. It has two lines.

2. This is verse two.
It has two lines.

3. This is verse three.
It has two lines.

4. This is verse four.
It has two lines.

5. This is verse five.
It has two lines.

Vedi anche

Guida al funzionamento interno: Sezione “LyricText” in *Guida al Funzionamento Interno*, Sezione “StanzaNumber” in *Guida al Funzionamento Interno*.

2.1.4 Canzoni

Riferimenti per canzoni

Le canzoni si scrivono solitamente su tre righe: la melodia per il cantante nel rigo superiore e due righe per l’accompagnamento di pianoforte in basso. Il testo della prima strofa appare immediatamente sotto il rigo superiore. Se ci sono solo poche altre strofe, queste possono essere poste subito sotto la prima; ma se ci sono più strofe di quante ne possano essere contenute lì la seconda strofa e le successive vengono stampate dopo la musica come testo separato.

Tutti gli elementi della notazione necessari per scrivere canzoni sono descritti dettagliatamente in altre parti della documentazione:

- Per costruire la struttura del rigo: Sezione 1.6.1 [Aspetto del rigo], pagina 202.
- Per scrivere la musica per pianoforte: Sezione 2.2 [Tastiera e altri strumenti multirigo], pagina 343.
- Per scrivere il testo da associare a una linea melodica: Sezione 2.1.1 [Notazione comune per la musica vocale], pagina 280.
- Per posizionare il testo: [Posizionamento verticale del testo], pagina 295.
- Per inserire le strofe: Sezione 2.1.3 [Strofe], pagina 310.
- Le canzoni hanno spesso i nomi degli accordi che appaiono sopra i righe. Questo argomento è trattato in Sezione 2.7.2 [Visualizzazione accordi], pagina 437.
- Per stampare i diagrammi degli accordi per l’accompagnamento per chitarra o altri strumenti a tasti: “Fret diagram markups” in Sezione 2.4.1 [Notazione comune per strumenti a corde con tasti], pagina 358.

Vedi anche

Manuale di apprendimento: Sezione “Canzoni” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 2.1.1 [Notazione comune per la musica vocale], pagina 280, Sezione 2.7.2 [Visualizzazione accordi], pagina 437, Sezione 1.6.1 [Aspetto del rigo], pagina 202, Sezione 2.2 [Tastiera e altri strumenti multirigo], pagina 343, [Posizionamento verticale del testo], pagina 295, Sezione 2.1.3 [Strofe], pagina 310.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Canzonieri

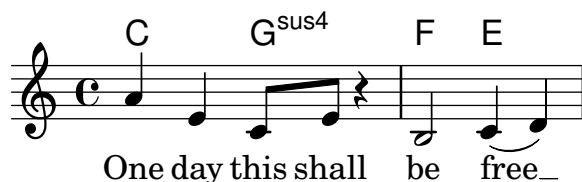
I canzonieri (in inglese *lead sheet*) si ottengono combinando le parti vocali con la ‘modalità per accordi’; la sintassi è spiegata in Sezione 2.7 [Notazione per accordi], pagina 431.

Frammenti di codice selezionati

Canzoniere semplice

Mettendo insieme nomi degli accordi, melodia e testo si ottiene un canzoniere (in inglese “lead sheet”):

```
<<
\chords { c2 g:sus4 f e }
\new Staff \relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



Vedi anche

Guida alla notazione: Sezione 2.7 [Notazione per accordi], pagina 431.

2.1.5 Musica corale

Questa sezione tratta le questioni che hanno a che fare più direttamente con la musica corale, compresi inni, cori polifonici, oratori, etc.

Riferimenti per musica corale

La musica corale viene solitamente rappresentata con due, tre o quattro righe in un gruppo ChoirStaff. L’accompagnamento, se richiesto, è posto sotto in un gruppo PianoStaff, che viene solitamente rimpicciolito per le prove di opere corali *a cappella*. Le note di ogni parte vocale sono inserite in un contesto Voice e a ogni rigo viene assegnato una sola parte vocale (ovvero un contesto Voice) o una coppia di parti vocali (due contesti Voice).

Le parole vengono poste nei contesti Lyrics, o sotto ciascun rigo musicale corrispondente oppure una strofa sopra e una sotto il rigo musicale se questo contiene musica per due parti.

Vari argomenti comuni nella musica corale sono trattati dettagliatamente in altre sezioni della documentazione:

- Un’introduzione alla creazione di una partitura vocale SATB si trova nel manuale di apprendimento: Sezione “Partitura vocale a quattro parti SATB” in *Manuale di Apprendimento*. È disponibile anche un modello integrato che semplifica la scrittura di musica vocale SATB: Sezione “Modelli integrati” in *Manuale di Apprendimento*.

- Vari modelli adatti per vari stili di musica corale si trovano anche nel manuale di apprendimento: Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.
- Le informazioni relative a `ChoirStaff` e `PianoStaff` si trovano in [Raggruppare i rigi], pagina 203.
- Le teste di nota a forma variabile, come quelle usate nello stile Sacred Harp e notazione simile, sono descritte in [Teste di nota a forma variabile], pagina 44.
- Quando due parti vocali si trovano su uno stesso rigo, i gambi, le legature, etc. della parte più alta sono dirette in su e quelle della parte più bassa sono rivolte in giù. Per farlo si usa `\voiceOne` e `\voiceTwo`, come è spiegato in [Polifonia su un solo rigo], pagina 184.
- Quando una parte vocale si divide temporaneamente, si devono usare i *passaggi polifonici temporanei* (vedi [Polifonia su un solo rigo], pagina 184).

Comandi predefiniti

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

Vedi anche

Manuale di apprendimento: Sezione “Partitura vocale a quattro parti SATB” in *Manuale di Apprendimento*, Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.1.7 [Ordine di disposizione dei contesti], pagina 623, [Raggruppare i rigi], pagina 203, [Teste di nota a forma variabile], pagina 44, [Polifonia su un solo rigo], pagina 184.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “ChoirStaff” in *Guida al Funzionamento Interno*, Sezione “Lyrics” in *Guida al Funzionamento Interno*, Sezione “PianoStaff” in *Guida al Funzionamento Interno*.

Struttura di una partitura corale

La musica corale contenente quattro rigi, con o senza accompagnamento del pianoforte, viene solitamente disposta in due sistemi per pagina. A seconda della dimensione della pagina, ottenere ciò può richiedere modifiche a varie impostazioni predefinite. Occorre considerare le seguenti impostazioni:

- La dimensione globale del rigo può essere modificata per cambiare la dimensione degli elementi della partitura. Vedi Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.
- Le distanze tra i sistemi, i rigi e il testo cantato possono essere tutte regolate in modo indipendente. Vedi Sezione 4.4 [Spaziatura verticale], pagina 574.
- Le dimensioni delle variabili di disposizione verticale possono essere mostrate per facilitare la regolazione della spaziatura verticale. Questa e altre possibilità per far entrare la musica in meno pagine sono descritte in Sezione 4.6 [Riduzione del numero di pagine di una partitura], pagina 601.
- Se il numero dei sistemi per pagina cambia da uno a due, è uso comune indicare ciò con un separatore di sistema tra i due sistemi. Vedi [Separare i sistemi], pagina 209.
- Per maggiori informazioni sulle proprietà di formattazione della pagina, leggere Sezione 4.1 [Formattazione della pagina], pagina 552.

Le indicazioni dinamiche sono poste, per impostazione predefinita, sotto il rigo, ma nella musica corale sono poste solitamente sopra il rigo per evitare il testo cantato. Il comando predefinito `\dynamicUp` permette di ottenere ciò per le indicazioni dinamiche in un singolo contesto `Voice`. Se ci sono molti contesti `Voice`, questo comando dovrebbe essere posto in ciascuno di essi. Ma

esiste la possibilità di usare una sola volta la sua forma estesa, che sposta tutte le indicazioni dinamiche dell'intera partitura sopra i loro rispettivi righi, come mostrato in questo esempio:

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
        \relative { g'4\ff g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative { d'4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override DynamicText.direction = #UP
      \override DynamicLineSpanner.direction = #UP
    }
  }
}
```



Comandi predefiniti

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Vedi anche

Guida alla notazione: Sezione 4.6.2 [Modificare la spaziatura], pagina 602, Sezione 4.6.1 [Visualizzare la spaziatura], pagina 601, Sezione 4.6 [Riduzione del numero di pagine di una partitura], pagina 601, Sezione 4.1 [Formattazione della pagina], pagina 552, Sezione 4.2 [Formattazione della partitura], pagina 563, [Separare i sistemi], pagina 209, Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565, Sezione 4.3 [Interruzioni], pagina 567, Sezione 4.4 [Spaziatura verticale], pagina 574.

Guida al funzionamento interno: Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno*, Sezione “StaffGrouper” in *Guida al Funzionamento Interno*.

Frammenti di codice selezionati

Using a bracket to clarify divisi

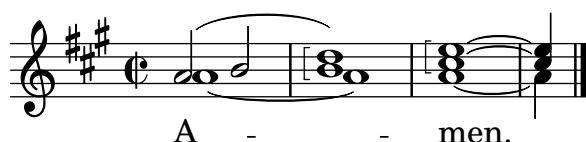
The `\nonArpeggiato` command can be used to indicate the division of voices where there are no stems to provide the information. This is often seen in choral music.

```
\include "english.ly"
```

```

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
    \new Voice = "upper" <<
    {
      \voiceOne
      a2( b2
      <b d>1\nonArpeggiato)
      <cs e>\nonArpeggiato ~
      <cs e>4
      \fine
    }
    \addlyrics { \lyricmode { A -- men. } }
  >>
  \new Voice = "lower" {
    \voiceTwo
    a1 ~
    a
    a ~
    a4
    \fine
  }
  >>
}

```



Vedi anche

Guida alla notazione: Sezione 1.3.3 [Indicazioni espressive lineari], pagina 149.

2.1.6 Opera e musical

La musica, il testo cantato e i dialoghi delle opere e dei musical sono di solito impostati in una o più delle seguenti forme:

- Una *partitura del direttore* contenente l'intera partitura orchestrale e le parti vocali, insieme a un libretto con le battute d'entrata se ci sono dei passaggi parlati.
- Le *parti orchestrali* contenenti la musica degli strumenti individuali dell'orchestra o della band.
- Una *partitura vocale* contenente tutte le parti vocali con accompagnamento del pianoforte. L'accompagnamento di solito è una riduzione orchestrale e in questo caso il nome dello strumento orchestrale originale viene spesso indicato. Le partiture vocali talvolta hanno anche le didascalie e il libretto con le battute d'entrata.
- Un *libro vocale* contenente soltanto le parti vocali (nessun accompagnamento), talvolta insieme al libretto.

- Un *libretto* contenente i passaggi estesi dei dialoghi parlati solitamente presenti nei musical, insieme alle parole delle parti cantate. Le didascalie sono solitamente incluse. È possibile usare LilyPond per creare i libretti, ma dato che non contengono musica è preferibile usare altri metodi.

Le sezioni della documentazione di LilyPond che trattano gli argomenti necessari per creare partiture negli stili più comuni nell'opera e nei musical sono indicati nei riferimenti seguenti. A questi seguono sezioni che trattano le tecniche specifiche che sono peculiari per le partiture di opera e musical.

Riferimenti per opera e musical

Oltre che ai gruppi vocali e teatrali, la maggior parte delle seguenti nozioni si applica a quasi qualsiasi musica d'orchestra e di insieme:

- La partitura di un direttore d'orchestra contiene molti rigi e testi vocali raggruppati. I modi per raggruppare i rigi sono mostrati in [Raggruppare i rigi], pagina 203. Per annidare gruppi di rigi leggere [Gruppi di rigi annidati], pagina 207.
- La stampa di rigi vuoti nelle partiture musicali e vocali dei direttori è solitamente soppressa. Per creare tale partitura leggere [Nascondere i rigi], pagina 217.
- La scrittura di parti orchestrali è trattata in Sezione 1.6.3 [Scrittura delle parti], pagina 219. Altre sezioni del capitolo Notazione specialistica potrebbero essere rilevanti, a seconda dell'orchestrazione usata. Molti strumenti sono strumenti traspositori, vedi [Trasporto strumentale], pagina 27.
- Se il numero di sistemi per pagina cambia da pagina a pagina, è di uso comune separare i sistemi con un segno separatore di sistemi. Vedi [Separare i sistemi], pagina 209.
- Per maggiori informazioni sulle proprietà di formattazione della pagina leggere Sezione 4.1 [Formattazione della pagina], pagina 552.
- Si possono inserire suggerimenti di dialogo, didascalie e note a piè di pagina, vedi Sezione 3.2.4 [Creazione di note a piè di pagina], pagina 512, e Sezione 1.8 [Testo], pagina 253. Didascalie estese possono essere aggiunte anche con una sezione di markup indipendenti tra i due blocchi `\score`, vedi [Testo separato], pagina 258.

Vedi anche

Glossario musicale: Sezione “Partitura senza rigi vuoti” in *Glossario Musicale*, Sezione “Rigo temporaneo” in *Glossario Musicale*, Sezione “Strumento traspositore” in *Glossario Musicale*.

Guida alla notazione: Sezione 3.2.4 [Creazione di note a piè di pagina], pagina 512, [Raggruppare i rigi], pagina 203, [Nascondere i rigi], pagina 217, [Trasporto strumentale], pagina 27, [Gruppi di rigi annidati], pagina 207, Sezione 4.1 [Formattazione della pagina], pagina 552, [Separare i sistemi], pagina 209, [Trasposizione], pagina 11, Sezione 1.6.3 [Scrittura delle parti], pagina 219, Sezione 1.8.1 [Inserimento del testo], pagina 253.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Nomi dei personaggi

I nomi dei personaggi sono solitamente mostrati a sinistra del rigo quando il rigo è dedicato a quel personaggio soltanto:

```
\score {
  <<
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative {
```



```

\clef "G_8"
c'4 c c c
\break
c4 c c c
}
}
\new Staff {
  \set Staff.vocalName = \markup \smallCaps Melchior
  \set Staff.shortVocalName = \markup \smallCaps Mel
  \clef "bass"
  \relative {
a4 a a a
a4 a a a
  }
}
>>
}

```

Quando due o più personaggi condividono lo stesso rigo, il nome del personaggio è solitamente collocato sopra il rigo all'inizio di ogni sezione appartenente a quel personaggio. È possibile fare ciò con i markup. Spesso si usa un tipo di carattere preciso a questo scopo.

```

\relative c' {
  \clef "G_8"
  c4~\markup \fontsize #1 \smallCaps Kaspar
  c c c
  \clef "bass"
  a4~\markup \fontsize #1 \smallCaps Melchior
  a a a
  \clef "G_8"
  c4~\markup \fontsize #1 \smallCaps Kaspar
  c c c
}

```

Altrimenti, se ci sono molti cambi di personaggi, è più semplice impostare una variabile per salvare le definizioni di ogni personaggio, in modo che il cambio di personaggio possa essere indicato in modo facile e conciso.

```

kaspar = {
  \clef "G_8"
  \set Staff.shortVocalName = "Kas."
  \set Staff.midiInstrument = "voice oohs"
  <>^\markup \smallCaps "Kaspar"
}

melchior = {
  \clef "bass"
  \set Staff.shortVocalName = "Mel."
  \set Staff.midiInstrument = "choir aahs"
  <>^\markup \smallCaps "Melchior"
}

\relative c' {
  \kaspar
  c4 c c c
  \melchior
  a4 a a a
  \kaspar
  c4 c c c
}

```



Vedi anche

Manuale di apprendimento: Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.8 [Testo], pagina 253, Sezione A.12 [Comandi per *markup*], pagina 713.

Suggerimenti musicali

I suggerimenti musicali possono essere inseriti nelle partiture vocali, nei libri vocali e nelle parti orchestrali per indicare quale musica in un'altra parte precede immediatamente un'entrata. I suggerimenti sono spesso inseriti anche nella riduzione per pianoforte nelle partiture vocali per indicare cosa sta suonando ogni strumento dell'orchestra. Ciò aiuta il direttore quando non è disponibile la partitura completa.

Il meccanismo di base per inserire i suggerimenti è spiegato dettagliatamente in [Citare altre voci], pagina 222, e [Formattazione delle notine], pagina 226. Ma quando si devono inserire molti suggerimenti, per esempio per aiutare il direttore in una partitura vocale, il nome dello strumento deve essere posizionato attentamente proprio prima dell'inizio delle citazioni in corpo più piccolo (“cue notes”). L'esempio seguente mostra come si fa.

```

flute = \relative {
  s4 s4 e'' g
}
\addQuote "flute" { \flute }

pianoRH = \relative {
  c''4. g8
}

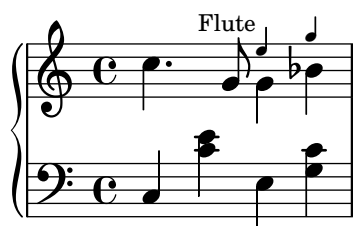
```

```

% posiziona il nome dello strumento citato proprio prima delle citazioni in corpo pi
% e sopra il rigo
<>~\markup { \right-align { \tiny "Flute" } }
\cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}

```



Se viene citato uno strumento traspositore, la parte strumentale deve specificare la sua armatura di chiave in modo che la conversione delle sue notine sia fatta automaticamente. Il prossimo esempio mostra questa trasposizione per un clarinetto in Si bemolle. Le note in questo esempio si trovano in basso nel rigo, quindi viene specificato DOWN in `\cueDuring` (in modo che i gambi vadano giù) e il nome dello strumento è posizionato sotto il rigo.

```

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
  \transposition c'
  % posiziona il nome dello strumento citato sotto il rigo
  <>~\markup { \right-align { \tiny "Clar." } }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH

```

```

    }
    }
    \new Staff {
    \clef "bass"
    \pianoLH
    }
    >>
    >>
}

```



Da questi due esempi è evidente che inserire molte citazioni in corpo piccolo in una partitura vocale sarebbe noioso, e le note della parte per pianoforte sarebbero confuse. Tuttavia, come mostra il frammento seguente, è possibile definire una funzione musicale per ridurre l'input e rendere più chiare le note per pianoforte.

Frammenti di codice selezionati

Aggiungere citazioni orchestrali a una partitura vocale

L'esempio seguente mostra un approccio per semplificare l'aggiunta di citazioni orchestrali a una riduzione per pianoforte di una partitura vocale. La funzione musicale `\cueWhile` prende quattro argomenti: la musica da cui prendere la citazione, come è definita da `\addQuote`, il nome da inserire prima delle notine, poi o `#UP` o `#DOWN` per specificare o `\voiceOne` col nome sopra il rigo o `\voiceTwo` col nome sotto il rigo, e infine la musica per pianoforte che deve apparire in parallelo alle notine. Il nome dello strumento citato è posto a sinistra delle notine. Molti passaggi possono essere citati, ma non possono sovrapporsi l'un l'altro nel tempo.

```

cueWhile =
#(define-music-function
  (instrument name dir music)
  (string? string? ly:dir? ly:music?)
  #{
    \cueDuring $instrument #dir {
      \once \override TextScript.self-alignment-X = #RIGHT
      \once \override TextScript.direction = $dir
      <>-\markup { \tiny #name }
      $music
    }
  })

flute = \relative c' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {

```

```

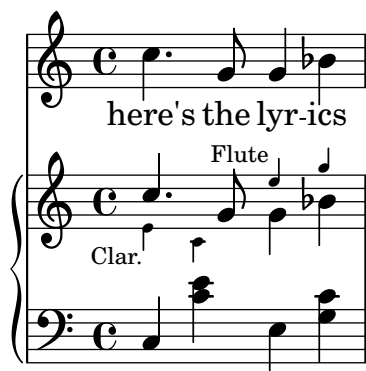
\transposition bes
fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    }
  >>
}

```



Vedi anche

Glossario musicale: Sezione “Notine o Citazioni in corpo più piccolo” in *Glossario Musicale*.

Guida alla notazione: Sezione 5.5.1 [Allineamento degli oggetti], pagina 660, Sezione 5.4.2 [Direzione e posizionamento], pagina 643, [Formattazione delle notine], pagina 226, [Citare altre voci], pagina 222, Sezione 5.6 [Uso delle funzioni musicali], pagina 674.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “CueVoice” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

`\cueDuring` inserisce automaticamente un contesto CueVoice e tutte le notine sono poste in quel contesto. Ciò significa che non è possibile avere due sequenze sovrapposte di notine con questa tecnica. Si possono inserire sequenze sovrapposte solo dichiarando esplicitamente contesti CueVoice distinti e usando `\quoteDuring` per estrarre e inserire le notine.

Musica parlata

Effetti come il ‘parlato’ o ‘Sprechgesang’ chiedono all’esecutore di parlare senza intonare note ma andando comunque a ritmo; tali effetti si indicano con teste di nota barrate, come è illustrato in [Teste di nota speciali], pagina 41.

Dialogo sopra la musica

Il dialogo parallelo alla musica appare solitamente sopra i rigi in corsivo, con l’inizio di ogni frase collegato a un momento musicale ben preciso.

In caso di brevi intromissioni può bastare un semplice `\markup`.

```
\relative {
  a'4~\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
  a4 a~\markup { \smallCaps { Bethan - } \italic Where? } a
  a4 a a a
}
```



In caso di frasi più lunghe può essere necessario espandere la musica per poter far entrare le parole. Non c’è modo di fare ciò del tutto automaticamente in LilyPond e sarà necessario anche qualche intervento manuale per formattare la pagina.

In caso di frasi lunghe o di passaggi con molti dialoghi serrati, l’uso di un contesto Lyrics darà risultati migliori. Il contesto Lyrics non deve essere associato a una voce musicale; occorre invece assegnare a ogni parte del dialogo una durata esplicita. Se c’è un vuoto nel dialogo, la parola finale deve essere separata dal resto e la durata divisa tra le due così che la musica sottostante abbia spazio sufficiente.

Se il dialogo si estende per più di un rigo sarà necessario inserire manualmente dei `\break` e aggiustare il posizionamento del dialogo per evitare di entrare nel margine destro. La parola finale dell’ultima misura di un rigo deve anche essere separata, come mostrato prima.

Ecco un esempio che illustra come fare.

```
music = \relative {
  \repeat unfold 3 { a'4 a a a }
}
```

```

dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
  }4*7
  "two"4 |
  \break
  "and this over measure"4*3
  "three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText.font-shape = #'italic
      \override LyricText.self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}

```



Vedi anche

Guida alla notazione: [Durate manuali delle sillabe], pagina 287, Sezione 1.8 [Testo], pagina 253.

Internal Reference: Sezione “LyricText” in *Guida al Funzionamento Interno*.

2.1.7 Canti salmi e inni

La musica e le parole per canti, sali e inni di solito segue un formato ben definito in ciascuna chiesa. Sebbene i formati possano differire da chiesa a chiesa, i problemi tipografici che si possono incontrare sono generalmente simili e sono trattati in questa sezione.

Riferimenti per canti e salmi

La composizione tipografica dei canti gregoriani in vari stili di notazione antica è descritta in Sezione 2.9 [Notazione antica], pagina 454.

Vedi anche

Guida alla notazione: Sezione 2.9 [Notazione antica], pagina 454.

Frammenti: Sezione “Vocal music” in *Frammenti di codice*.

Impostare un canto

Le impostazioni per il canto moderno usano la notazione moderna insieme ad alcuni elementi della notazione antica. Alcuni di questi elementi e metodi sono mostrati in questa sezione.

I canti usano spesso note di un quarto senza gambi per indicare l'altezza, mentre le durate vengono dal ritmo parlato delle parole.

```
stemOff = { \hide Staff.Stem }
```

```
\relative c' {  
  \stemOff  
  a'4 b c2 |  
}
```



Nei canti le stanghette sono spesso omesse oppure si usano delle stanghette più brevi o punteggiate per indicare le pause nella musica. Per omettere tutte le stanghette da tutti i righi si disattiva l'incisore delle stanghette:

```
\score {  
  \new StaffGroup <<  
    \new Staff {  
      \relative {  
a'4 b c2 |  
a4 b c2 |  
a4 b c2 |  
      }  
    }  
    \new Staff {  
      \relative {  
a'4 b c2 |  
a4 b c2 |  
a4 b c2 |  
      }  
    }  
  >>  
  \layout {  
    \context {  
      \Staff  
      \remove Bar_engraver  
    }  
  }  
}
```

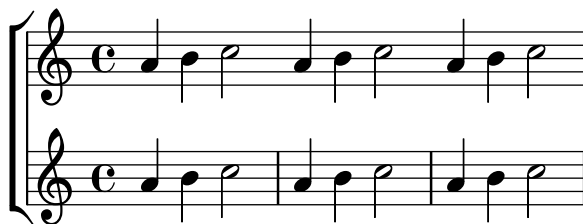


Le stanghette possono anche essere tolte solo in certi righi:


```

\score {
  \new ChoirStaff <<
    \new Staff
      \with { \remove Bar_engraver } {
        \relative {
          a'4 b c2 |
          a4 b c2 |
          a4 b c2 |
        }
      }
    \new Staff {
      \relative {
        a'4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
}

```



Per togliere le stanghette da una sezione musicale soltanto basta trattarla come una cadenza. Se la sezione è lunga potrebbe essere necessario inserire delle stanghette fittizie con `\bar ""` per mostrare dove si deve interrompere la linea.

```

\relative a' {
  a4 b c2 |
  \cadenzaOn
  a4 b c2
  a4 b c2
  \bar ""
  a4 b c2
  a4 b c2
  \cadenzaOff
  a4 b c2 |
  a4 b c2 |
}

```



Nei canti le pause si indicano con stanghette modificate.

```

\relative a' {
  a4
  \cadenzaOn
  b c2
}

```

```

a4 b c2
\bar " " "
a4 b c2
a4 b c2
\bar ";"
a4 b c2
\bar "!"
a4 b c2
\bar "||"
}

```

```

        \remove Bar_engraver
    }
}
}

```



Nei canti viene solitamente omessa l'indicazione di tempo e spesso anche la chiave.

```

\score {
  \new Staff {
    \relative {
      a'4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
      \Staff
      \remove Bar_engraver
      \remove Time_signature_engraver
      \remove Clef_engraver
    }
  }
}

```



I canti per salmi della tradizione anglicana sono solitamente o *singoli*, con 7 battute musicali, oppure *doppi*, con due gruppi di 7 battute. Ogni gruppo di 7 battute è diviso a metà, che corrispondono alle metà di ciascun verso, di solito separato da una doppia stanghetta. Si usano solo semibreve e minime. La prima battuta di ogni metà contiene sempre un solo accordo di semibreve, che viene chiamato la “nota recitativa”. I canti sono centrati sulla pagina.

```

SopranoMusic = \relative {
  g'1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

```

```

AltoMusic = \relative {
  e'1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

```

```

TenorMusic = \relative {
  c'1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

```

```

BassMusic = \relative {

```

```

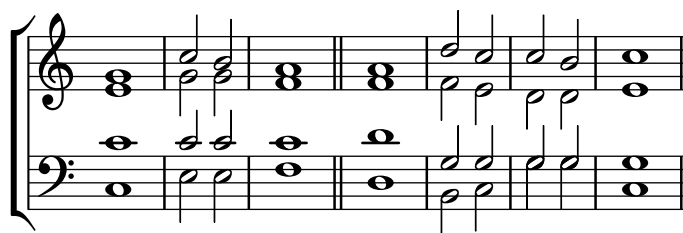
    c1 | e2 e | f1 |
    d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

% Si usa un blocco markup per centrare il canto sulla pagina
\markup {
  \fill-line {
    \score { % centered
      <<
      \new ChoirStaff <<
        \new Staff <<
          \global
          \clef "treble"
          \new Voice = "Soprano" <<
            \voiceOne
            \SopranoMusic
          >>
          \new Voice = "Alto" <<
            \voiceTwo
            \AltoMusic
          >>
        >>
        \new Staff <<
          \clef "bass"
          \global
          \new Voice = "Tenor" <<
            \voiceOne
            \TenorMusic
          >>
          \new Voice = "Bass" <<
            \voiceTwo
            \BassMusic
          >>
        >>
      >>
    }
    \layout {
      \context {
        \Score
        \override SpacingSpanner.base-shortest-duration = \musicLength 2
      }
      \context {
        \Staff
        \remove Time_signature_engraver
      }
    }
  } % End score
}

```

```
} % End markup
```



Altri approcci per impostare un canto simile sono illustrati nel primo dei seguenti frammenti.

Frammenti di codice selezionati

Chant or psalm notation

This form of notation is used for psalm chant, where verses are not always of the same length.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \section
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \section
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \fine
  }
}
```



I cantici e altri testi liturgici possono essere impostati in modo più libero e possono usare elementi della notazione della musica antica. Le parole sono spesso mostrate sotto e allineate con le note. In questo caso le note sono spaziate in base alle sillabe invece che alle durate delle note.

Ancient notation template – modern transcription of Gregorian music

This example demonstrates how to do modern transcription of Gregorian music. Gregorian music has no measure, no stems; it uses only half and quarter note heads, and special marks, indicating rests of different length.

```
chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g f) a2 \finalis \break
  f4 a2 \divisioMinima
```

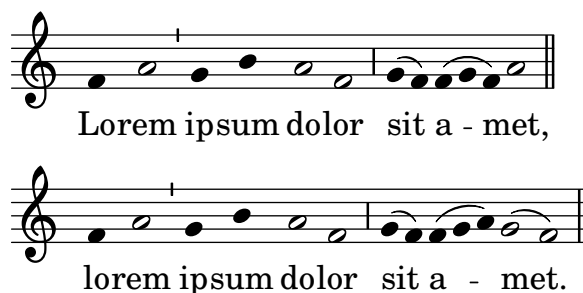
```

g4 b a2 f2 \divisioMaior
g4( f) f( g a) g2( f) \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met,
  lo -- rem ip -- sum do -- lor sit a -- met.
}

\score {
  \new GregorianTranscriptionStaff <<
    \new GregorianTranscriptionVoice = "melody" \chant
    \new GregorianTranscriptionLyrics = "one" \lyricsto melody \verba
  >>
}

```



Vedi anche

Manuale di apprendimento: Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*, Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 2.9 [Notazione antica], pagina 454, [Stanghette], pagina 105, Sezione 5.1.4 [Modifica dei componenti aggiuntivi di un contesto], pagina 613, Sezione 2.9.4 [Scrivere il canto gregoriano], pagina 465, [Musica in tempo libero], pagina 79, Sezione 5.4.7 [Visibilità degli oggetti], pagina 652.

Salmi

Le parole di un salmo anglicano sono solitamente stampate in versi separati centrati sotto il canto.

I canti singoli (7 battute) si ripetono per ogni verso. I canti doppi (con 14 battute) si ripetono per ogni coppia di versi. Sono inseriti dei segni nelle parole per mostrare come si combinano col canto. Ogni verso è diviso a metà e tale divisione è indicata solitamente dai due punti, che corrispondono alla doppia stanghetta in musica. Le parole che precedono i due punti sono cantate insieme alle prime tre battute della musica; quelle successive insieme alle restanti quattro battute.

Stanghette singole (o in alcuni libri di salmi una virgola inversa o segno simile) sono inserite tra le parole per indicare dove cadono le stanghette nella musica. In modalità markup una stanghetta singola può essere inserita usando il simbolo di controllo battuta |.

```

\markup {
  \fill-line {
    \column {
      \left-align {
        \line { 0 come let us sing | unto the | Lord : let }
        \line { us heartily rejoice in the | strength of | our }
      }
    }
  }
}

```

```

\line { sal- | -vation. }
}
}
}
}

```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Altri simboli potrebbero richiedere i glifi dei tipi di carattere fetaMusic. Maggiori informazioni in Sezione 1.8.3 [Tipi di carattere], pagina 273.

```

tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { 0 come let us sing \tick unto the \tick Lord : let }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}

```

O come let us sing ' unto the ' Lord : let
us heartily rejoice in the ' strength of ' our
sal ' vation.

Se c'è una semibreve in una battuta, tutte le parole che si riferiscono a quella battuta sono recitate su quella nota singola col ritmo del parlato. Dove ci sono due note in una battuta ci saranno solo una o due sillabe corrispondenti. Se ci sono più di due sillabe, si inserisce solitamente un punto per indicare dove si trova il cambio di nota.

```

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          0 come let us sing \tick unto \dot the \tick Lord : let
        }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
      }
    }
  }
}

```

```

}
\line { sal \tick vation. }
}
}
}
}

```

O come let us sing 'unto • the ' Lord : let
us heartily rejoice in the ' strength of ' our
sal ' vation.

In alcuni libri di salmi si usa un asterisco, al posto di una virgola, per indicare una pausa in una sezione recitata, mentre le sillabe accentate o leggermente allungate sono indicate in grassetto.

```

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { Today if ye will hear his voice * }
        \line {
          \concat { \bold hard en }
          | not your | hearts : as in the pro-
        }
        \line { vocation * and as in the \bold day of tempt- | }
        \line { -ation | in the | wilderness. }
      }
    }
  }
}
}

```

Today if ye will hear his voice *
harden | not your | hearts : as in the pro-
vocation * and as in the **day** of tempt- |
-ation | in the | wilderness.

In altri libri di salmi si usa un simbolo di accento sopra la sillaba per indicare l'accento.

```

tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us \concat {
            si \combine \tick ng

```



```

    }
    | unto the | Lord : let
  }
  \line {
    us heartily \concat {
      rejo \combine \tick ice
    }
    in the | strength of | our
  }
  \line { sal- | -vation. }
    }
  }
}

```

O come let us *síng* | unto the | Lord : let
 us heartily *rejoíce* in the | strength of | our
 sal- | -vation.

L'uso di `\markup` per centrare il testo e disporre le linee in colonne è descritto in Sezione 1.8.2 [Formattazione del testo], pagina 260.

La maggior parte di questi elementi sono mostrati in uno dei due versi del modello Sezione “Salmi” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Salmi” in *Manuale di Apprendimento*, Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.8.3 [Tipi di carattere], pagina 273, Sezione 1.8.2 [Formattazione del testo], pagina 260.

Misure parziali nei motivi degli inni

Le melodie degli inni spesso iniziano e terminano ogni rigo musicale con misure parziali, così che ciascun rigo musicale corrisponda esattamente a un rigo di testo. Per fare ciò è necessario un comando `\partial` all’inizio della musica e dei comandi `\bar " | "` o `\bar " | | "` alla fine di ogni linea.

Modello per inno

Il codice seguente presenta un modo di impostare un inno in cui ogni verso inizia e finisce con una misura parziale. Mostra anche come aggiungere delle strofe come testo separato sotto la musica.

```

Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \caesura \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \fine
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

```

```

}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic = \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
  \key g \major
}

\score { % Start score
  \new PianoStaff << % Start pianostaff
    \new Staff << % Start Staff = RH
      \global
      \clef "treble"
      \new Voice = "Soprano" << % Start Voice = "Soprano"
        \Timeline
        \voiceOne
        \SopranoMusic
      >> % End Voice = "Soprano"
      \new Voice = "Alto" << % Start Voice = "Alto"
        \Timeline
        \voiceTwo
        \AltoMusic
      >> % End Voice = "Alto"
    >> % End Staff = RH

  \new Staff << % Start Staff = LH
    \global
    \clef "bass"
    \new Voice = "Tenor" << % Start Voice = "Tenor"
      \Timeline
      \voiceOne
      \TenorMusic
    >> % End Voice = "Tenor"
    \new Voice = "Bass" << % Start Voice = "Bass"
      \Timeline
      \voiceTwo
      \BassMusic
    >> % End Voice = "Bass"
  >> % End Staff = LH
}

```

```

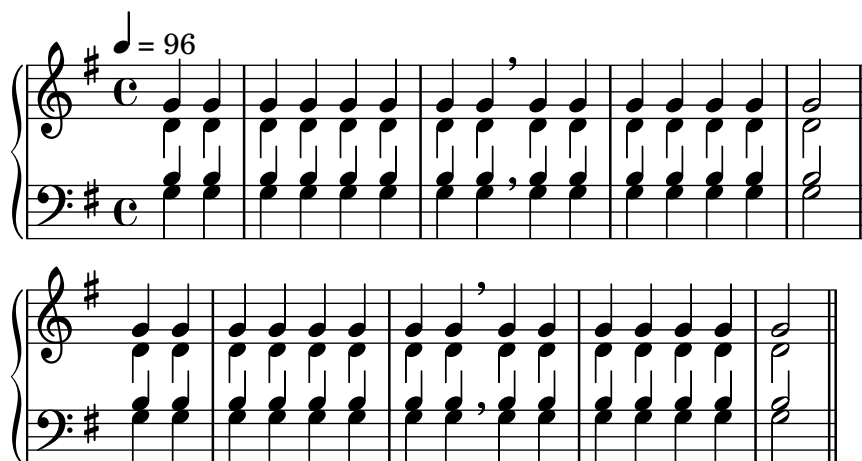
>> % End pianostaff
} % End score

\markup \fill-line {
  \left-column {
    "This is line one of the first verse"
    "This is line two of the same"
    \null
    "And here's line one of the second verse"
    "And the next line of the same"
  }
}

\layout {
  \context {
    \Score
    caesuraType = #'((bar-line . "||"))
    fineBarType = "||"
  }
}

\paper { % Start paper block
  indent = 0 % don't indent first system
  line-width = 130 % shorten line length to suit music
  tagline = ##f % Don't print tag line, can be removed
} % End paper block

```



This is line one of the first verse
 This is line two of the same

And here's line one of the second verse
 And the next line of the same

2.1.8 Musica vocale antica

È supportata la musica vocale antica, come spiegato in Sezione 2.9 [Notazione antica], pagina 454.

Vedi anche

Guida alla notazione: Sezione 2.9 [Notazione antica], pagina 454.

2.2 Tastiera e altri strumenti multirigo

Un peu retenu
très expressif

ppp

rall. - - - - *a tempo*

long

pp

ppp

8va

ral - - len - - tan - - do - - - - Lent

Questa sezione tratta vari aspetti della notazione musicale che sono specifici degli strumenti a tastiera e di altri strumenti che vengono scritti su molti righi, come arpe e vibrafoni. Ai fini di questa sezione, questo gruppo di strumenti multirigo viene chiamato “tastiere” per brevità, anche se alcuni di questi strumenti non hanno una tastiera.

2.2.1 Notazione comune per tastiere

Questa sezione tratta le questioni di notazione che riguardano la maggior parte degli strumenti a tastiera.

Riferimenti per tastiere

Gli strumenti a tastiera vengono rappresentati solitamente con righi per pianoforte, ovvero due o più righi normali raggruppati da una parentesi graffa. La stessa notazione viene usata anche

per altri strumenti dotati di tasti. La musica per organo viene scritta normalmente con due righi all'interno di un gruppo PianoStaff più un terzo rigo normale per i pedali.

I righi nella musica per tastiera sono molto indipendenti, ma talvolta le voci possono attraversare i due righi. Questa sezione tratta tecniche di notazione specifiche della musica per tastiera.

Varie questioni comuni della musica per tastiera sono discusse altrove:

- La musica per tastiera contiene solitamente varie voci, il cui numero può cambiare regolarmente, come è descritto in [Risoluzione delle collisioni], pagina 188.
- La musica per tastiera può essere scritta in parallelo, come è descritto in [Scrivere la musica in parallelo], pagina 199.
- Le dinamiche possono essere poste in un contesto Dynamics, tra i due contesti Staff, per allineare le indicazioni dinamiche su una linea orizzontale centrata tra i righi; vedi [Dinamiche], pagina 132.
- Le diteggiature sono indicate con [Indicazioni di diteggiatura], pagina 235.
- Le indicazioni per i pedali dell'organo sono inserite come articolazioni, vedi Sezione A.15 [Elenco delle articolazioni], pagina 793.
- Le linee della griglia verticali possono essere mostrate con le [Linee della griglia], pagina 248.
- La musica per tastiera contiene spesso legature di valore *Laissez vibrer* così come legature di valore su arpeggi e tremoli, descritti in [Legature di valore], pagina 58.
- Il posizionamento di arpeggi attraverso molteplici voci e rigi è trattato in [Arpeggio], pagina 154.
- I segni di tremolo sono descritti in [Ripetizioni con tremolo], pagina 176.
- Varie modifiche manuali che possono rendersi necessarie nella musica per tastiera sono illustrati in Sezione “Esempio musicale” in *Manuale di Apprendimento*.
- Si possono usare le note nascoste per produrre delle legature di valore che attraversano le voci, come mostrato in Sezione “Altri usi delle modifiche con tweak” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Esempio musicale” in *Manuale di Apprendimento*, Sezione “Altri usi delle modifiche con tweak” in *Manuale di Apprendimento*.

Guida alla notazione: [Raggruppare i rigi], pagina 203, [Nomi degli strumenti], pagina 219, [Risoluzione delle collisioni], pagina 188, [Scrivere la musica in parallelo], pagina 199, [Indicazioni di diteggiatura], pagina 235, Sezione A.15 [Elenco delle articolazioni], pagina 793, [Linee della griglia], pagina 248, [Legature di valore], pagina 58, [Arpeggio], pagina 154, [Ripetizioni con tremolo], pagina 176.

Guida al funzionamento interno: Sezione “PianoStaff” in *Guida al Funzionamento Interno*.

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Modifica manuale del rigo

Le voci possono essere passate a un altro rigo manualmente, usando il comando

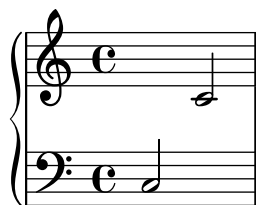
```
\change Staff = nomerigo
```

La stringa *nomerigo* è il nome del rigo. Fa passare la voce corrente dal suo rigo corrente al rigo chiamato *nomerigo*. Valori tipici di *nomerigo* sono "su" e "giù" o "MD" (mano destra) e "MS" (mano sinistra).

Il rigo a cui viene passata la voce deve esistere nel momento dello scambio. Se necessario, i rigi devono essere “conservati”, vedi Sezione 5.1.3 [Conservazione di un contesto], pagina 611,

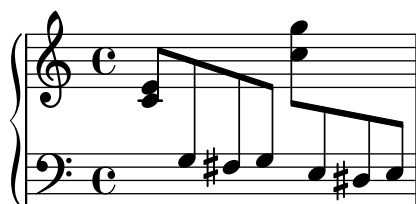
o istanziati esplicitamente, per esempio usando l'accordo vuoto, <>, vedi [Note in un accordo], pagina 178.

```
\new PianoStaff <<
  \new Staff = "up" {
    % forza la creazione di tutti i contesti in questo momento
    <>
    \change Staff = "down" c2
    \change Staff = "up" c'2
  }
  \new Staff = "down" {
    \clef bass
    % conserva il rigo
    s1
  }
>>
```



Le note trasversali al rigo hanno la travatura automatica:

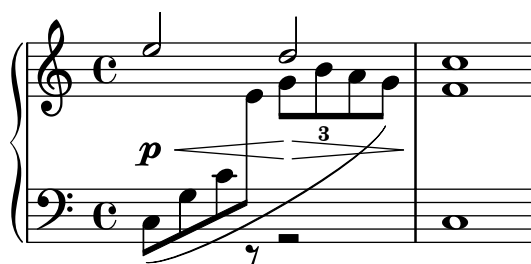
```
\new PianoStaff <<
  \new Staff = "su" {
    <e' c'>8
    \change Staff = "giù"
    g8 fis g
    \change Staff = "su"
    <g'' c''>8
    \change Staff = "giù"
    e8 dis e
    \change Staff = "su"
  }
  \new Staff = "giù" {
    \clef bass
    % conserva il contesto
    s1
  }
>>
```



Se la disposizione delle travature deve essere modificata, modificare prima le direzioni dei gambi. Le posizioni delle travature vengono quindi misurate dal centro del rigo che è più vicino alla travatura. Un semplice esempio di modifica manuale della travatura è descritto in Sezione “Correggere elementi della notazione sovrapposti” in *Manuale di Apprendimento*.

Alcuni elementi della notazione potranno sovrapporsi quando le voci sono trasversali ai righi:

```
\new PianoStaff <<
  \new Staff = "su" {
    \voiceOne
    % Fai spazio alla ditekgiatura nella voce trasversale ai righi
    \once\override DynamicLineSpanner.staff-padding = #4
    e''2\p\< d''\>
    c''1\!
  }
  \new Staff = "giù" <<
  {
    \clef bass
    s4. e,8\rest g,2\rest
    c1
  } \ {
    c8\< g c'
    \change Staff = "su"
    e' g' b'-3 a' g'\)
    f'1
  }
>>
>>
```



Il gambo e la legatura di portamento si sovrappongono alla linea intermedia della dinamica, perché la risoluzione automatica delle collisioni è sospesa per travature, legature di portamento e altri estensori che collegano note di righi diversi, così come per gambi e articolazioni se il loro posizionamento è influenzato da un estensore trasversale ai righi. Le collisioni risultanti devono essere risolte manualmente, laddove necessario, usando i metodi descritti in Sezione “Correggere elementi della notazione sovrapposti” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Correggere elementi della notazione sovrapposti” in *Manuale di Apprendimento*.

Guida alla notazione: [Gambi], pagina 244, [Travature automatiche], pagina 88, Sezione 5.1.3 [Conservazione di un contesto], pagina 611.

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Beam” in *Guida al Funzionamento Interno*, Sezione “ContextChange” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Non è possibile evitare la collisione delle travature per le travature automatiche che terminano appena prima di un cambio di rigo. In questo caso occorre usare le travature manuali.

Modifica automatica del rigo

È possibile scambiare automaticamente le voci tra rigo superiore e inferiore. La sintassi è la seguente:

```
\autoChange ...musica...
```

Ciò creerà due righi dentro il gruppo di righi corrente (solitamente un PianoStaff), chiamati "up" (su) e "down" (giù). Il rigo più basso sarà in chiave di basso. La funzione di scambio automatico entra in azione in base all'altezza (il Do centrale è il punto di svolta), e guarda in avanti saltando le pause per poter scambiare in anticipo.

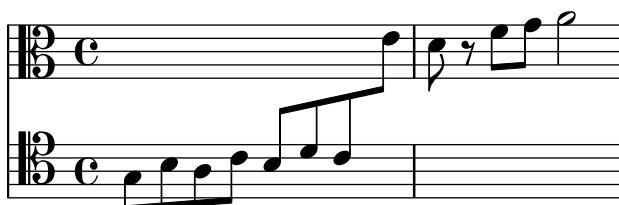
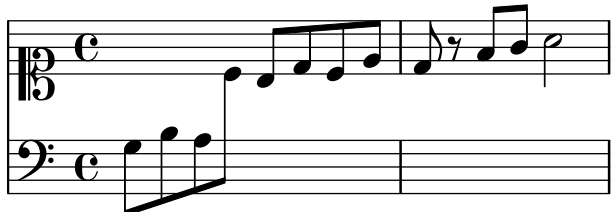
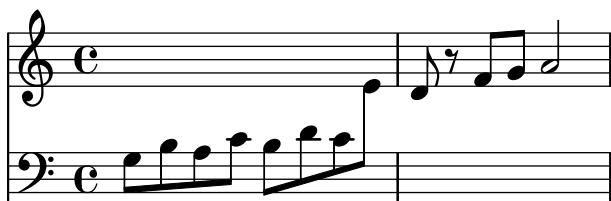
```
\new PianoStaff {
  \autoChange {
    g4 a b c'
    d'4 r a g
  }
}
```



Si possono specificare altre altezze per il punto di svolta. Se i righi non sono istanziati esplicitamente, si possono usare altre chiavi.

```
music = {
  g8 b a c' b8 d' c'8 e'
  d'8 r f' g' a'2
}
```

```
\autoChange d' \music
\autoChange b \with { \clef soprano } \music
\autoChange d' \with { \clef alto } \with { \clef tenor } \music
```



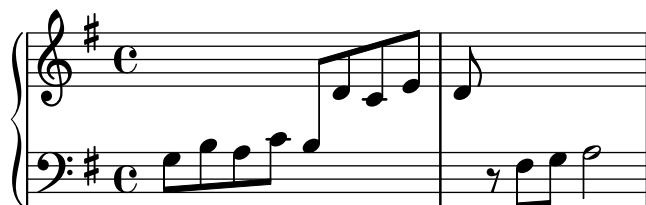
Una sezione `\relative` che si trova fuori da `\autoChange` non ha effetto sulle altezze della musica. Dunque, se necessario, mettere `\relative` dentro `\autoChange`.

Se è richiesto ulteriore controllo sui righi individuali, possono essere creati manualmente coi nomi "up" e "down". Il comando `\autoChange` scambierà allora la sua voce tra i righi esistenti.

Nota: Se i righi sono creati manualmente, *devono* chiamarsi "up" e "down".

Per esempio, i righi devono essere creati manualmente per posizionare un'armatura di chiave nel rigo inferiore:

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melOne" {
      \key g \major
      \autoChange \relative {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
>>
```



Vedi anche

Guida alla notazione: [Modifica manuale del rigo], pagina 344.

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Problemi noti e avvertimenti

Gli scambi di rigo potrebbero finire in posizioni non ottimali. Per ottenere un risultato di alta qualità, i cambi di rigo devono essere specificati manualmente.

Gli accordi non vengono divisi tra i righi; vengono assegnati a un rigo in base alla prima nota nominata nel costrutto dell'accordo.

Linee del cambio rigo

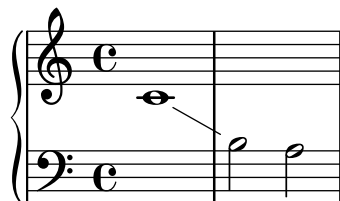
Quando una voce passa a un altro rigo, è possibile stampare automaticamente una linea connettiva:

```
\new PianoStaff <<
  \new Staff = "uno" {
    \showStaffSwitch
    c'1
```

```

\change Staff = "due"
b2 a
}
\new Staff = "due" {
\clef bass
s1*2
}
>>

```

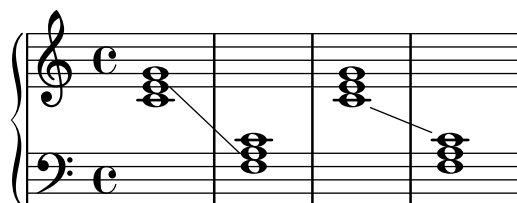


Una linea di cambio rigo tra accordi collega le “ultime note” degli accordi come sono scritte nel file sorgente; ciò può essere utile per regolare velocemente le posizioni verticali di inizio e fine della linea.

```

\new PianoStaff <<
\new Staff = "uno" {
<c' e' g'>1
\showStaffSwitch
\change Staff = "due"
<a c' f>1
\hideStaffSwitch
\change Staff = "uno"
<e' g' c'>1
\showStaffSwitch
\change Staff = "due"
<f a c'>1
}
\new Staff = "due" {
\clef bass
s1*4
}
>>

```



Comandi predefiniti

`\showStaffSwitch`, `\hideStaffSwitch`.

Vedi anche

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Note_head_line_engraver” in *Guida al Funzionamento Interno*, Sezione “VoiceFollower” in *Guida al Funzionamento Interno*.

Frammenti di codice selezionati

Cross-staff stems

This snippet shows how to use `Span_stem_engraver` and `\crossStaff` to connect stems across staves automatically.

The stem lengths need not be specified, as the variable distance between noteheads and staves is calculated automatically. However, it is important that `\crossStaff` is applied to the correct voice or staff (i.e., on the opposite side of where a beam is or would be positioned) to get the desired effect.

```
\layout {
  \context {
    \PianoStaff
    \consists "Span_stem_engraver"
  }
}

\new PianoStaff <<
  \new Staff {
    <b d'>4 r d'16\> e'8. g8 r\! |
    e'8 f' g'4
    \voiceTwo
    % Down to lower staff
    \crossStaff { e'8 e'8 } e'4 |
  }

  \new Staff {
    \clef bass
    \voiceOne
    % Up to upper staff
    \crossStaff { <e g>4 e, g16 a8. c8 } d |
    g8 f g4 \voiceTwo g8 g g4 |
  }
>>
```



Indicating cross-staff chords with a bracket

An non-arpeggiato bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the `PianoStaff` must be set to accept cross-staff brackets.

The following example typesets measure 65 of Debussy's prelude *Les collines d'Anacapri*.

```
\new PianoStaff <<
  \once \set PianoStaff.connectChordBrackets = ##t

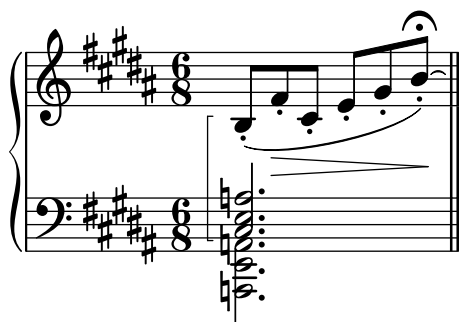
  \new Staff \relative c' {
    \key b \major
```

```

\time 6/8
b8-.( \nonArpeggiato fis'-.\> cis-.
  e-. gis-. b-. ) \!\fermata^\laissezVibrer
\section
}

\new Staff \relative c' {
  \clef bass
  \key b \major
  << { <a e cis>2. \nonArpeggiato } \\\
    { <a, e a,>2. } >>
  \section
}
>>

```



Vedi anche

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Stem” in *Guida al Funzionamento Interno*.

2.2.2 Pianoforte

Questa sezione tratta le questioni di notazione che hanno a che fare più direttamente col pianoforte.

Pedali del pianoforte

I pianoforti hanno generalmente tre pedali che alterano il modo in cui il suono viene prodotto: *risonanza*, *tonale* o *sostenuto* (*sos.*) e *una corda* (*U.C.*). I pedali di risonanza si trovano anche nei vibrafoni e nelle celeste.

```

\relative {
  c''4 \sustainOn d e g
  <c, f a>1 \sustainOff
  c4 \sostenutoOn e g c,
  <bes d f>1 \sostenutoOff
  c4 \unaCorda d e g
  <d fis a>1 \treCorde
}

```



Ci sono tre stili per indicare i pedali: text (testo), bracket (parentesi quadra) e mixed (misto). Il pedale di risonanza e il pedale una corda usano il testo come stile predefinito, mentre il pedale tonale usa lo stile misto.

```
\relative {
  c' '4\sustainOn g c2\sustainOff
  \set Staff.pedalSustainStyle = #'mixed
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2\sustainOff
  \set Staff.pedalSustainStyle = #'bracket
  c4\sustainOn g c d
  d\sustainOff\sustainOn g, c2
  \bar "|."
}
```



Il posizionamento dei comandi del pedale corrisponde ai movimenti fisici del pedale di risonanza durante l'esecuzione musicale. Per indicare che il pedale è attivo fino alla stanghetta finale si omette il comando finale che disattiva il pedale.

Le indicazioni del pedale possono essere poste in un contesto Dynamics, che le allinea su una linea orizzontale.

Vedi anche

Guida alla notazione: [Legature di valore], pagina 58.

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “SustainPedal” in *Guida al Funzionamento Interno*, Sezione “SustainPedalLineSpanner” in *Guida al Funzionamento Interno*, Sezione “SustainEvent” in *Guida al Funzionamento Interno*, Sezione “SostenutoPedal” in *Guida al Funzionamento Interno*, Sezione “SostenutoPedalLineSpanner” in *Guida al Funzionamento Interno*, Sezione “SostenutoEvent” in *Guida al Funzionamento Interno*, Sezione “UnaCordaPedal” in *Guida al Funzionamento Interno*, Sezione “UnaCordaPedalLineSpanner” in *Guida al Funzionamento Interno*, Sezione “UnaCordaEvent” in *Guida al Funzionamento Interno*, Sezione “PianoPedalBracket” in *Guida al Funzionamento Interno*, Sezione “Piano_pedal_engraver” in *Guida al Funzionamento Interno*.

2.2.3 Fisarmonica

Questa sezione tratta la notazione specifica della fisarmonica.

Simboli di discanto

Le fisarmoniche sono spesso costruite con più di un insieme di ance che possono essere all'unisono, un'ottava sopra o un'ottava sotto rispetto all'altezza scritta. Ogni costruttore di fisarmoniche ha nomi diversi per i registri che selezionano le varie combinazioni di ance, come *oboe*, *musette* o *bandoneon*. È stato quindi ideato un sistema di simboli per semplificare le istruzioni di esecuzione.

Un elenco completo di tutti i registri per fisarmonica disponibili si trova in Sezione A.12.7 [Accordion registers], pagina 773.

Frammenti di codice selezionati

Simboli di registro della fisarmonica

I simboli di registro della fisarmonica sono disponibili sia come `\markup` sia come eventi musicali autonomi (perché i cambi di registro capitano solitamente tra reali eventi musicali). I registri bassi non sono troppo standardizzati. I comandi disponibili si trovano nella sezione «Registri della fisarmonica» della Guida alla notazione.

```
#(use-modules (lily accreg))

\new PianoStaff <<
  \new Staff \relative {
    \clef treble
    \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    <<
      { r16 <f bes> r <e a> r <d g> }
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>~"b" r <e a c>~"am" r <d g bes>~"gm" |
        <e a cis>1~"a" }
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```



Vedi anche

Frammenti: Sezione “Keyboard and other multi-staff instruments” in *Frammenti di codice*.

2.2.4 Arpa

Questa sezione tratta le questioni di notazione specifiche dell’arpa.

Riferimenti per arpe

Alcune caratteristiche comuni della musica per arpa sono trattate altrove:

- Il glissando è la tecnica più caratteristica dell’arpa, vedi [Glissando], pagina 149.
- Un *bisbigliando* è scritto come un tremolo, vedi [Ripetizioni con tremolo], pagina 176.
- Gli armonici naturali sono trattati in [Armonici], pagina 356.
- Per arpeggi direzionali e non-arpeggi, vedere [Arpeggio], pagina 154.

Vedi anche

Guida alla notazione: [Ripetizioni con tremolo], pagina 176, [Glissando], pagina 149, [Arpeggio], pagina 154, [Armonici], pagina 356.

Pedali dell’arpa

Le arpe hanno sette corde per ottava che possono risuonare all’altezza naturale, bemolle o diesis. Nelle arpe con leva, ogni corda viene regolata individualmente, ma nelle arpe a pedale ogni corda con la stessa altezza è regolata da un unico pedale. I pedali, da sinistra a destra rispetto all’esecutore, sono D, C e B a sinistra e E, F, G e A a destra. La posizione dei pedali può essere indicata con dei segni testuali:

```
\textLengthOn
cis''1_\markup \concat \vcenter {
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c''!1_\markup \concat \vcenter {
  [ C \natural ] }
```



o con dei diagrammi dei pedali:

```
\textLengthOn
cis''1_\markup { \harp-pedal "^v-|vv-^" }
c''!1_\markup { \harp-pedal "^o--|vv-^" }
```



Il comando `\harp-pedal` accetta una stringa di caratteri, dove `^` è la posizione più alta del pedale (altezza bemolle), `-` è la posizione centrale del pedale (altezza naturale), `v` è la posizione più bassa del pedale (altezza diesis) e `|` è il divisore. `o` posto prima di un simbolo produrrà un cerchio intorno al simbolo di pedale.

Vedi anche

Guida alla notazione: [Scritte], pagina 253, Sezione A.12.6 [Instrument-specific markup], pagina 768.

2.3 Strumenti a corde senza tasti

1 **lentement**

fatigué s. vib. n. p. vib. s. vib.

IV 1) n. 2) s.p. IV IV

mf *mf* *mf* *ff* *pp*

accel... s.p. n. s.p. n. p. vib.

IV IV IV IV

mf *ff*

s.p. n. s.p. n. p. vib. m. vib.

IV IV IV

ppp

The image displays three staves of musical notation for string instruments without keys. The first staff is marked 'lentement' and 'fatigué', featuring a sequence of notes with vibrato (s. vib.) and dynamic markings (mf, ff, pp). The second staff shows a triplet of notes with an 'accel...' marking, followed by a sequence of notes with dynamic markings (mf, ff). The third staff includes a 'ritar...' marking and a sequence of notes with dynamic markings (ppp). Various technical markings like 'IV', 'n.', 's.p.', and 'p. vib.' are present throughout the notation.

Questa sezione fornisce informazioni e riferimenti utili per scrivere musica per strumenti a corde senza tasti, principalmente archi da orchestra.

2.3.1 Notazione comune per strumenti a corde senza tasti

La notazione specialistica per strumenti a corde senza tasti è esigua. La musica viene scritta su un solo rigo e di solito è richiesta una sola voce. Due voci potrebbero essere necessarie solo per alcuni passaggi a doppia corda o i divisi.

Riferimenti per archi senza tasti

La maggior parte della notazione utile per gli archi da orchestra e altri strumenti ad arco è trattata altrove:

- Le indicazioni testuali come “pizz.” e “arco” sono inserite come testo semplice, vedi [Scritte], pagina 253.
- Le diteggiature, incluse le indicazioni per il pollice, sono descritte in [Indicazioni di diteggiatura], pagina 235.
- I numeri di corda si inseriscono (generalmente in numeri romani per gli strumenti ad arco) come è spiegato in [Indicazioni di numero di corda], pagina 359.
- La doppia corda viene normalmente indicata scrivendo un accordo, vedi [Note in un accordo], pagina 178. Si possono inserire delle direttive su come eseguire gli accordi, vedi [Arpeggio], pagina 154.
- I modelli per i quartetti d’archi si trovano in Sezione “Modelli per quartetto d’archi” in *Manuale di Apprendimento*. Altri sono illustrati nei frammenti.

Vedi anche

Manuale di apprendimento: Sezione “Modelli per quartetto d’archi” in *Manuale di Apprendimento*.

Guida alla notazione: [Scritte], pagina 253, [Indicazioni di diteggiatura], pagina 235, [Note in un accordo], pagina 178, [Arpeggio], pagina 154.

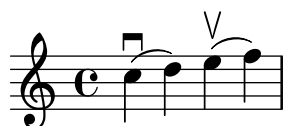
Frammenti: Sezione “Unfretted string instruments” in *Frammenti di codice*.

Indicazioni di arcata

Le indicazioni di arcata vengono create come articolazioni, descritte in [Articolazioni e abbellimenti], pagina 129.

I comandi di arcata, `\upbow` (su) e `\downbow` (giù), vengono usati con le legature di portamento nel seguente modo:

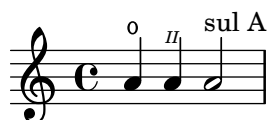
```
\relative { c' '4(\downbow d) e(\upbow f) }
```



Si possono usare i numeri romani come numeri di corda (al posto dei numeri arabi racchiusi in un cerchio), come è spiegato in [Indicazioni di numero di corda], pagina 359.

Altrimenti, le indicazioni di corda possono essere mostrate anche tramite comandi markup; le articolazioni possono indicare anche corde vuote.

```
a'4 \open
\romanStringNumbers
a'\2
a'2^\markup { \small "sul A" }
```



Comandi predefiniti

`\downbow`, `\upbow`, `\open`, `\romanStringNumbers`.

Vedi anche

Guida alla notazione: [Articolazioni e abbellimenti], pagina 129, [Indicazioni di numero di corda], pagina 359, [Legature di portamento], pagina 142.

Armonici

Armonici naturali

Gli armonici naturali possono essere rappresentati in vari modi. Una testa di nota romboidale indica generalmente che bisogna toccare la corda dove si prenderebbe la nota se questa non fosse romboidale.

```
\relative d' ' {
  d4 e4.
  \harmonicsOn
  d8 e e
  d4 e4.
  \harmonicsOff
}
```

```
d8 e e
}
```



Altrimenti si può mostrare una testa di nota normale all'altezza da riprodurre insieme a un piccolo cerchio che indica che deve essere suonata come armonico:

```
d''2^\flageolet d''_\flageolet
```



Si può creare anche un cerchio più piccolo, vedi l'elenco di frammenti in [Riferimenti per archi senza tasti], pagina 355.

Armonici artificiali

Gli armonici artificiali sono rappresentati da due note, una con una testa normale per indicare la posizione in cui sfiorare la corda e un'altra con una testa romboidale vuota per indicare la posizione dell'armonico.

Gli armonici artificiali indicati con `\harmonic` non mostrano i punti. Occorre impostare la proprietà di contesto `harmonicDots` per far apparire i punti.

```
\relative e' {
  <e a\harmonic>2. <c g'\harmonic>4
  \set harmonicDots = ##t
  <e a\harmonic>2. <c g'\harmonic>4
}
```



Vedi anche

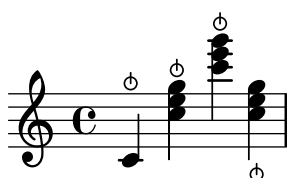
Glossario musicale: Sezione “armonici” in *Glossario Musicale*.

Guida alla notazione: [Teste di nota speciali], pagina 41, [Riferimenti per archi senza tasti], pagina 355.

Pizzicato alla Bartók

Il pizzicato alla Bartók, in inglese “snap pizzicato” (pizzicato a schiocco), è un tipo di pizzicato in cui la corda viene deliberatamente pizzicata verticalmente (invece che di lato) in modo che sbatta sulla tastiera.

```
\relative {
  c'4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



2.4 Strumenti a corde con tasti

The image displays five staves of musical notation for stringed instruments with keys, such as guitar or banjo. The notation includes various techniques and dynamics:

- Staff 1:** Features a treble clef with a key signature of two sharps (F# and C#). It shows a series of eighth-note triplets, with dynamics *fp* (fortissimo piano) and a first fingering '1' indicated.
- Staff 2:** Continues with eighth-note triplets, including a *rit.* (ritardando) marking and a *dim.* (diminuendo) dynamic. It then transitions to a new section marked *Andantino* in common time (C), starting with a piano (*p*) dynamic.
- Staff 3:** Labeled *il canto ben marcato*, it shows a melody with dotted rhythms and a *p dol.* (piano dolce) dynamic. Fingering numbers 1, 2, 3, and 4 are visible.
- Staff 4:** Continues the melodic line with various fingering numbers (1, 2, 3, 4) and a 4/4 time signature.
- Staff 5:** Shows further melodic development with a 4/4 time signature and a fermata over the final measure.

Questa sezione tratta vari aspetti della notazione musicale che riguardano esclusivamente gli strumenti a corde con tasti.

2.4.1 Notazione comune per strumenti a corde con tasti

Questa sezione presenta la notazione comune che riguarda esclusivamente gli strumenti a corde con tasti.

Riferimenti per strumenti a corde con tasti

La musica per strumenti a corde con tasti viene normalmente scritta su un rigo singolo, o nella notazione musicale tradizionale o in intavolatura. Talvolta i due tipi sono combinati; nella musica popolare è molto comune usare i diagrammi degli accordi sopra un rigo di notazione tradizionale. La chitarra e il banjo sono strumenti traspositori, che suonano un'ottava più bassa di quella in cui vengono scritti. Le partiture per questi strumenti devono usare la chiave "treble_8" (per ottenere l'output MIDI corretto usare `\transposition c`). Altri elementi relativi agli strumenti a corde con tasti sono trattati altrove:

- Le diteggiature vengono indicate come spiegato in [Indicazioni di diteggiatura], pagina 235.

- Istruzioni per le legature di valore *Laissez vibrer* così come per le legature di valore su arpeggi e tremoli si trovano in [Legature di valore], pagina 58.
- Istruzioni su come gestire voci multiple si trovano in [Risoluzione delle collisioni], pagina 188.
- Istruzioni per indicare gli armonici si trovano in [Armonici], pagina 356.

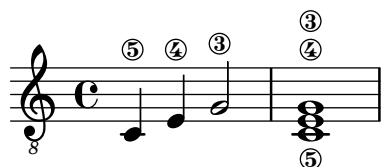
Vedi anche

Guida alla notazione: [Indicazioni di diteggiatura], pagina 235, [Legature di valore], pagina 58, [Risoluzione delle collisioni], pagina 188, [Nomi degli strumenti], pagina 219, [Scrivere la musica in parallelo], pagina 199, [Arpeggio], pagina 154, Sezione A.15 [Elenco delle articolazioni], pagina 793, [Chiave], pagina 17, [Trasporto strumentale], pagina 27.

Indicazioni di numero di corda

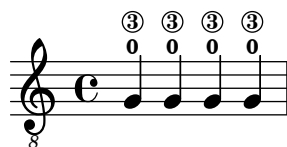
La corda sulla quale suonare una nota può essere indicata aggiungendo `\numero` a una nota.

```
\clef "treble_8"
c4\5 e\4 g2\3
<c\5 e\4 g\3>1
```



Se si usano insieme diteggiature e indicazioni di corda, il loro posizionamento è regolato dall'ordine in cui i due elementi appaiono nel codice *solo* se si trovano all'interno di un accordo esplicito: se applicate a un intero accordo o a singole note *fuori* da un accordo, le diteggiature vengono posizionate con un meccanismo diverso.

```
\clef "treble_8"
g4\3-0
g-0\3
<g\3-0>
<g-0\3>
```



I numeri di corda possono anche essere rappresentati da numeri romani, come è abituale con gli strumenti a corde senza tasti, e posizionati sotto il rigo invece che sopra.

```
\clef "treble_8"
c'2\2
a\3
\romanStringNumbers
c'\2
\set stringNumberOrientations = #'(down)
a\3
\arabicStringNumbers
g1\4
```



Frammenti di codice selezionati

La maggior parte dei comportamenti relativi alle indicazioni di numero di corda (oggetto `StringNumber`), incluso il loro posizionamento, possono essere impostati nello stesso modo valido per le diteggiature: vedi [Indicazioni di diteggiatura], pagina 235.

Comandi predefiniti

`\arabicStringNumbers`, `\romanStringNumbers`.

Vedi anche

Guida alla notazione: [Indicazioni di diteggiatura], pagina 235.

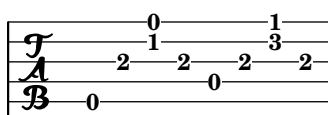
Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “StringNumber” in *Guida al Funzionamento Interno*, Sezione “Fingering” in *Guida al Funzionamento Interno*.

Intavolature predefinite

La musica per gli strumenti a corde pizzicate si scrive con un sistema di notazione basato sulle dita/tocco ovvero l'intavolatura (o tablatura). Diversamente dalla notazione tradizionale, le altezze non sono rappresentate da teste di nota, ma da numeri (o da simboli simili a lettere nell'intavolatura antica). Le linee del rigo nell'intavolatura indicano la corda su cui suonare la nota, mentre un numero posto su una linea del rigo indica il tasto su cui premere la corda corrispondente. Le note da suonare simultaneamente sono allineate verticalmente. Per impostazione predefinita, la corda 1 è la corda più alta e corrisponde alla linea più alta del rigo dell'intavolatura (`TabStaff`). L'accordatura predefinita delle corde di `TabStaff` è l'accordatura standard per chitarra (a sei corde). Le note appaiono in intavolatura, usando i contesti `TabStaff` e `TabVoice`. Viene aggiunta automaticamente una chiave calligrafica per l'intavolatura.

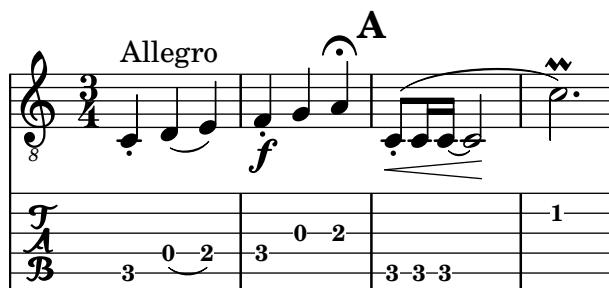
```
\new TabStaff \relative {
  a,8 a' <c e> a
  d,8 a' <d f> a
}
```



L'aspetto predefinito per le intavolature non prevede alcun simbolo di durata né altri simboli musicali, come per esempio i segni di espressione.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-. \f g a^ \fermata
  \mark \default
  c8_. \<\( c16 c~ 2\!
  c'2. \prall\
}

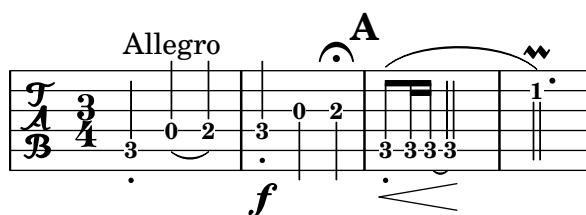
\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff { \symbols }
  >>
}
```



Se si desidera che tutti i simboli usati nella notazione tradizionale appaiano anche nell'intavolatura, occorre applicare il comando `\tabFullNotation` in un contesto `TabStaff`. Tenere presente che nell'intavolatura le note da due quarti hanno un doppio gambo per distinguerle dalle note di un quarto.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-. \f g a~\fermata
  \mark \default
  c8_. \<\( c16 c~ 2\!
  c'2. \prall\
}
```

```
\score {
  \new TabStaff {
    \tabFullNotation
    \symbols
  }
}
```



Di norma le altezze vengono assegnate alla posizione più bassa della tastiera (prima posizione). Viene data priorità alle corde a vuoto. Per far sì che una certa altezza venga suonata su una corda specifica, aggiungere un'indicazione di numero di corda al nome dell'altezza. Per nascondere le indicazioni di numero di corda nella notazione tradizionale, sovrascrivere lo stampo (stencil) corrispondente.

Spesso è più comodo definire la posizione sulla tastiera modificando la proprietà `minimumFret`, il cui valore predefinito è 0.

Anche quando `minimumFret` è impostato esplicitamente, vengono usate le corde a vuoto ogni volta che è possibile. È possibile modificare questo comportamento impostando `restrainOpenStrings` su `#t`.

```
\layout { \omit Voice.StringNumber }
\new StaffGroup <<
  \new Staff \relative {
    \clef "treble_8"
    \time 2/4
    c16 d e f g4
```

```

c,16\5 d\5 e\4 f\4 g4\4
c,16 d e f g4
}
\new TabStaff \relative {
  c16 d e f g4
  c,16\5 d\5 e\4 f\4 g4\4
  \set TabStaff.minimumFret = #5
  \set TabStaff.restrainOpenStrings = ##t
  c,16 d e f g4
}
>>

```

Gli accordi possono essere ripetuti col simbolo di ripetizione accordo `q`, che però rimuove i numeri di corda e le diteggiature. Se usato nelle intavolature, può quindi creare problemi. Converrà quindi eseguire il comando

```
\chordRepeats #'(string-number-event fingering-event)
```

esplicitamente sulle espressioni musicali dell'intavolatura che usano la [Ripetizione di un accordo], pagina 180. Questo comando è così comune che è disponibile come `\tabChordRepeats`.

```

guitar = \relative {
  r8 <gis-2 cis-3 b-0>~ q4 q8~ 8 q4
}

```

```

\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \guitar
  }
  \new TabStaff {
    \tabChordRepeats \guitar
  }
>>

```

Le legature di valore spezzate da un a capo appaiono tra parentesi. Lo stesso vale per la seconda alternativa di una ripetizione.

```
ties = \relative {
```



```

\repeat volta 2 {
  e'2. f4~
  2 g2~ }
\alternative {
  \volta 1 { g4 f2. }
  \volta 2 { g4\repeatTie c,2. }
}
b1~
\break
b1
\bar " | ."
}

\score {
  <<
    \new StaffGroup <<
      \new Staff {
        \clef "treble_8"
        \ties
      }
      \new TabStaff {
        \hideSplitTiedTabNotes
        \ties
      }
    >>
  >>
  \layout {
    indent = #0
    ragged-right = ##t
  }
}

```

The image displays a musical score for guitar. The top staff is a treble clef staff with a melody. The melody starts with a quarter note G4, followed by a dotted quarter note F4, then a quarter note G4, a quarter note A4, a quarter note B4, and a quarter note C5. There is a repeat sign after the C5, followed by a first ending (1.) and a second ending (2.). The first ending consists of a quarter note G4 and a quarter note F4. The second ending consists of a quarter note G4 and a quarter note A4. Below the melody staff is a guitar tablature staff. The tablature shows fret numbers: 0, 1, 3, 1, 1, 0. There is a bar line after the first four fret numbers. Below the tablature staff is a smaller staff with a treble clef and a single note G4.

Gli armonici possono essere inseriti in un'intavolatura nelle altezze in cui suonano:

```

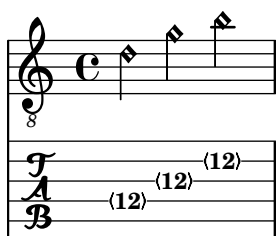
\layout { \omit Voice.StringNumber }
firstHarmonic = {

```

```

d'4\4\harmonic
g'4\3\harmonic
b'2\2\harmonic
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \firstHarmonic
    }
    \new TabStaff { \firstHarmonic }
  >>
}

```

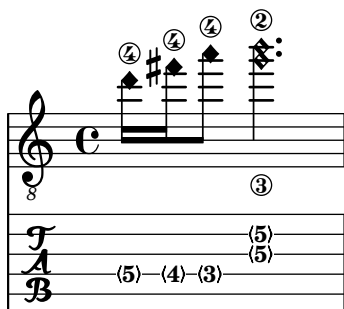


Il comando `\harmonic` deve essere sempre attaccato a note singole (possibilmente dentro a un costrutto per accordo) invece che ad accordi interi. Ha senso usarlo solo per gli armonici sul dodicesimo tasto. Tutti gli altri armonici devono essere calcolati da LilyPond. Ciò si ottiene indicando il tasto in cui il dito della mano che preme i tasti debba toccare una corda.

```

fretHarmonics = {
  \harmonicByFret #5 d16\4
  \harmonicByFret #4 d16\4
  \harmonicByFret #3 d8\4
  \harmonicByFret #5 <g\3 b\2>2.
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \fretHarmonics
    }
    \new TabStaff { \fretHarmonics }
  >>
}

```

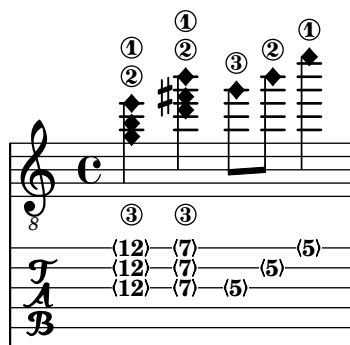


Gli armonici possono essere calcolati anche definendo il rapporto delle lunghezze della corda sopra e sotto la diteggiatura dell'armonico.

```

ratioHarmonics = {
  \harmonicByRatio #1/2 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/3 <g\3 b\2 e'\1>4
  \harmonicByRatio #1/4 { g8\3 b8\2 e'4\1 }
}
\score {
  <<
    \new Staff {
      \clef "treble_8"
      \ratioHarmonics
    }
    \new TabStaff { \ratioHarmonics }
  >>
}

```



I “bending” degli strumenti a corda sono supportati nell’intavolatura. Un bending – letteralmente, “piegatura” – viene introdotto aggiungendo `\^` alla nota o accordo da piegare; termina automaticamente alla nota o accordo successivi. Sono disponibili i seguenti stili: quello predefinito produce una curva con una punta di freccia rivolta verso l’alto o verso il basso, `'hold` una linea orizzontale tratteggiata, `'pre-bend` una linea verticale con una punta di freccia e `'pre-bend-hold` una linea verticale con una punta di freccia seguita da una linea tratteggiata.

```

bend-styles = {
  <>^"predefinito"
  f'4\^ g'4\^ f'2

  <>^"'hold"
  \grace f'4\^ g'1\bendHold \^ g'1

  <>^"'pre-bend"
  \grace f'4\preBend \^ g'1\bendHold \^ g'1

  <>^"'pre-bend-hold"
  \grace f'4\preBendHold \^ g'1\bendHold \^ g'1\^ f'

  \bar "|."
}

\score {
  \new StaffGroup
  <<
    \new Staff {
      \override TextScript.font-size = -2

```

```

\clef "G_8"
\bind-styles
}
\new TabStaff \bind-styles
>>
\layout {
  \context {
    \Voice
    \omit StringNumber
  }
  \context {
    \TabStaff
    minimumFret = #5
  }
  \context {
    \TabVoice
    \consists "Bend_spanner_engraver"
  }
}
}

```

I bending generalmente non si applicano alle corde vuote. Per piegarle occorre impostare la proprietà 'bend-me su #t. Per evitare che altre note siano piegate impostarla su #f.

```

mus = {
  <>~"predefinito"
  <a b f'>4\^
  <ais b fis'>\^
  <a b f'>2

  <>~"piega le corde vuote"
  <a \tweak bend-me ##t b f'>4\^
  <ais \tweak bend-me ##t bis fis'>\^
  <a b f'>2

  <>~"escludi altre corde"
  <g \tweak bend-me ##f b\3 d'>4\^
  <a e'\2 >\^
  <g \tweak bend-me ##f b\3 d'>2

  \bar "|."
}

\score {

```

```

\new StaffGroup
<<
  \new Staff {
    \override TextScript.font-size = -2
    \clef "G_8"
    \mus
  }
  \new TabStaff \mus
>>
\layout {
  \context {
    \Voice
    \omit StringNumber
  }
  \context {
    \TabVoice
    \consists "Bend_spanner_engraver"
  }
}
}

```

Per i bending consecutivi il bending iniziale potrebbe aver bisogno di un'impostazione appropriata per `details.successive-level`. Per comodità c'è la funzione `bendStartLevel`, che prende un numero intero.

```

\printNext = -\tweak details.target-visibility ##t \etc

mus = {
  c'4\3\^ cis'\3 \^ d'2\3

  \grace bes4\3\preBendHold \bendStartLevel 2 \printNext \^
  d'4\3\bendHold \^ d'2\3\^ des'4\3 \^ c'1\3

  \bar "|"
}

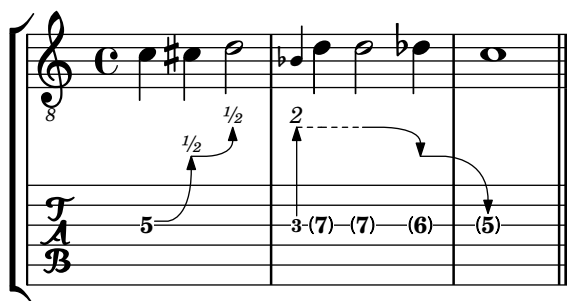
\score {
  \new StaffGroup
  <<
    \new Staff {
      \override TextScript.font-size = -2
      \clef "G_8"
      \mus
    }
  >>
}

```

```

\new TabStaff \mus
>>
\layout {
  \context {
    \Voice
    \omit StringNumber
  }
  \context {
    \TabVoice
    \consists "Bend_spanner_engraver"
  }
}
}

```



Per impostazione predefinita l'oggetto BendSpanner termina alla nota o accordo successivi anche se connesso con legatura di valore alla nota o accordo iniziali. Si può saltare una singola "colonna di note" (NoteColumn) tramite \skipNC. Si può saltare anche un gruppo di oggetti NoteColumn usando \skipNCs all'inizio e \endSkipNCs alla fine.

```

bends-with-ties-and-skips = {
  a'4~\^ \skipNC a'4~ \skipNC a'4 b'4
  a'4~ a'4~\^ \skipNC a'4 b'4
  a'4~ a'4~ a'4~\^ b'4
  c'2~\^ d'~ \bendHold \^ \skipNC d'~ d'\^ c'
  \grace { c'8~\preBendHold \^ }
  \skipNCs d'2~ d'2~ \endSkipNCs d'\^ c'2
  \bar "|."
}

```

```

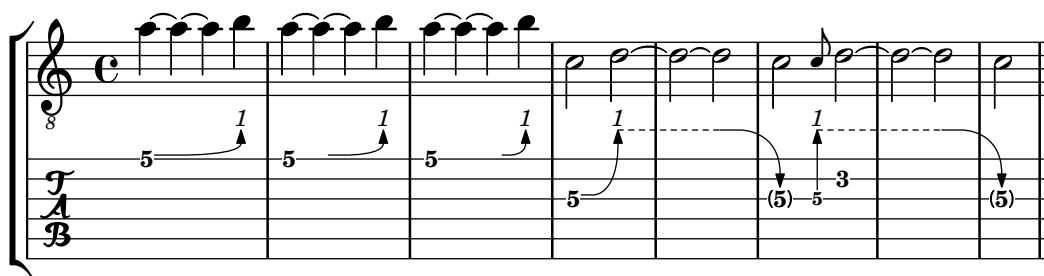
\score {
  \new StaffGroup
  <<
    \new Staff {
      \clef "G_8"
      \bends-with-ties-and-skips
    }
    \new TabVoice \bends-with-ties-and-skips
  >>
  \layout {
    \context {
      \Voice
      \omit StringNumber
    }
    \context {

```

```

\TabStaff
minimumFret = #3
restrainOpenStrings = ##t
}
\context {
\TabVoice
\consists "Bend_spanner_engraver"
}
}
}

```



Comandi predefiniti

\skipNCs, \skipNC, \endSkipNCs.

Frammenti di codice selezionati

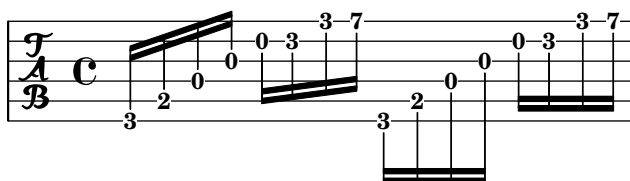
Comportamento di gambi e travature in intavolatura

La direzione dei gambi nell'intavolatura è regolata nello stesso modo della notazione tradizionale. Le travature possono essere rese orizzontali, come illustrato in questo esempio.

```

\new TabStaff {
\relative c {
\tabFullNotation
g16 b d g b d g b
\stemDown
\override Beam.concaveness = 10000
g,,16 b d g b d g b
}
}

```



Polifonia in intavolatura

La polifonia in TabStaff funziona proprio come nel rigo normale.

```

upper = \relative c' {
\time 12/8
\key e \minor
\voiceOne
r4. r8 e, fis g16 b g e e' b c b a g fis e
}

```

```

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  \new StaffGroup = "tab with traditional" <<
    \new Staff = "guitar traditional" <<
      \clef "treble_8"
      \new Voice = "upper" \upper
      \new Voice = "lower" \lower
    >>

    \new TabStaff = "guitar tab" <<
      \new TabVoice = "upper" \upper
      \new TabVoice = "lower" \lower
    >>
  >>
}

```

The image displays a musical score for guitar. The upper staff is a standard musical staff in E minor (one sharp) and 12/8 time. It contains a melody starting with a quarter rest, followed by eighth and sixteenth notes. The lower staff is a guitar tab staff, showing fret numbers (0, 2, 3, 4) and fingerings (0, 1, 2, 3) for the lower voice. The tab is aligned with the melody above it.

Armonici su corde a vuoto in intavolatura

Questo frammento mostra come scrivere armonici su corde a vuoto.

```

openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = 3
  \override TextSpanner.dash-fraction = 0.3
  \override TextSpanner.dash-period = 1

  % first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret 12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan

  % second harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "2nd harm. "
  \harmonicByFret 7 e,\6\startTextSpan
  \harmonicByRatio #1/3 e,\6
  \harmonicByFret 19 e,\6
}

```



```

\harmonicByRatio #2/3 e,\6\stopTextSpan
%\harmonicByFret 19 < e,\6 a,\5 d\4 >
%\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

% third harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "3rd harm. "
\harmonicByFret 5 e,\6\startTextSpan
\harmonicByRatio #1/4 e,\6
\harmonicByFret 24 e,\6
\harmonicByRatio #3/4 e,\6\stopTextSpan
\break

% fourth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "4th harm. "
\harmonicByFret 4 e,\6\startTextSpan
\harmonicByRatio #1/5 e,\6
\harmonicByFret 9 e,\6
\harmonicByRatio #2/5 e,\6
\harmonicByFret 16 e,\6
\harmonicByRatio #3/5 e,\6\stopTextSpan

% fifth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "5th harm. "
\harmonicByFret 3 e,\6\startTextSpan
\harmonicByRatio #1/6 e,\6\stopTextSpan
\break

% sixth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "6th harm. "
\harmonicByFret 2.7 e,\6\startTextSpan
\harmonicByRatio #1/7 e,\6\stopTextSpan

% seventh harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "7th harm. "
\harmonicByFret 2.3 e,\6\startTextSpan
\harmonicByRatio #1/8 e,\6\stopTextSpan

% eighth harmonic
\override TextSpanner.bound-details.left.text =
  \markup\small "8th harm. "
\harmonicByFret 2 e,\6\startTextSpan
\harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff \with { \omit StringNumber } {

```

```

\new Voice {
  \clef "treble_8"
  \openStringHarmonics
}
}
\new TabStaff {
  \new TabVoice {
    \openStringHarmonics
  }
}
>>
}

```

The image displays three systems of musical notation for open string harmonics. Each system consists of a treble staff with a C-clef and an 8-line tablature staff with a T-clef. The first system shows the 1st, 2nd, and 3rd harmonics. The second system shows the 4th and 5th harmonics. The third system shows the 6th, 7th, and 8th harmonics. Each harmonic is represented by a diamond-shaped note on the staff and a corresponding fingering number in parentheses on the tablature staff.

Harmonic	Staff Note (Fingering)	Tablature Fingering
1st	E4 (1)	(12)
2nd	G4 (2)	(7)
3rd	B4 (3)	(5)
4th	D5 (4)	(4)
5th	F#5 (5)	(3)
6th	A5 (6)	(2)
7th	C#6 (7)	(2)
8th	E6 (8)	(2)

Armonici su corde premute in intavolatura

Questo frammento mostra come scrivere su intavolatura armonici su corde premute.

```

pinchedHarmonics = {
  \textSpannerDown
  \override TextSpanner.bound-details.left.text =
    \markup {\halign #-0.5 \teeny "PH" }
  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.dash-period = 0.6
  \override TextSpanner.bound-details.right.attach-dir = 1
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.padding = -0.5
}

```

```

harmonics = {
  % artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4\_markup { \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4\_markup { \teeny "AH 17" }
  <\parenthesize d' d'\harmonic>2\_markup { \teeny "AH 19" }

  % pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d'\harmonic>4
  <e'\harmonic>4\stopTextSpan

  % tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4\_markup { \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4\_markup { \teeny "TH 19" }
  <\parenthesize c'\3 c'\harmonic>2\_markup { \teeny "TH 17" }

  % touch harmonics (TCH)
  a4( <e'\harmonic>2. )\_markup { \teeny "TCH" }
}

frettedStrings = {
  % artificial harmonics (AH)
  \harmonicByFret 4 g4\3
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 g2\3

  % pinched harmonics (PH)
  \harmonicByFret 7 d2\4
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 a4\5

  % tapped harmonics (TH)
  \harmonicByFret 5 d4\4
  \harmonicByFret 7 d4\4
  \harmonicByFret 5 g2\3

  % touch harmonics (TCH)
  a4 \harmonicByFret 9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
  }
  \new TabStaff {

```

```

\new TabVoice {
  \frettedStrings
}
}
>>
}

```

Slides in tablature

Slides can be typeset in both Staff and TabStaff contexts.

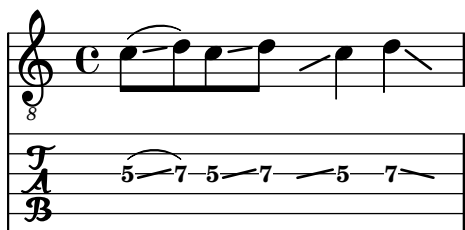
```

slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
    \stemDown \hideNotes
    g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>

  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = 4
      \override Glissando.springs-and-rods =
        #ly:spanner::set-spacing-rods
      \override Glissando.thickness = 2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}

```



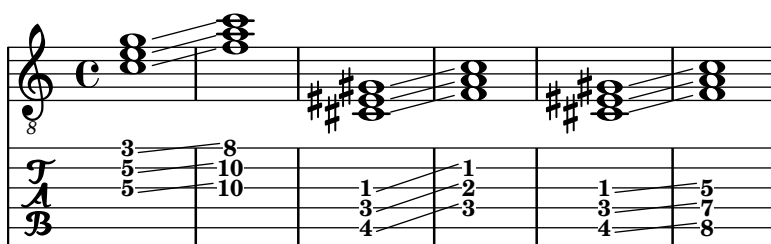
Glissando di accordi in intavolatura

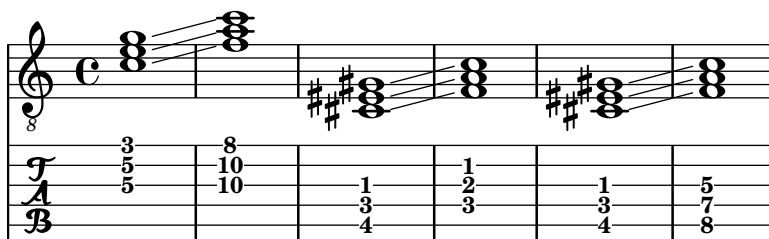
I glissati (o slide) di accordi vengono indicati sia nel rigo (contesto Staff) sia nell'intavolatura (contesto TabStaff). I numeri di corda sono necessari per TabStaff, perché i calcoli automatici della corda sono diversi per gli accordi e per le note singole.

```
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
  <cis, eis gis>1 \glissando <f a c>
  <cis eis gis>1 \glissando <f a c\3>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \myMusic
  >>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \omit StringNumber
      \myMusic
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}
```

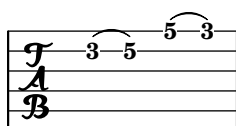




Martellato e strappato

Il martellato (hammer on) e lo strappato (pull off) si possono ottenere con le legature di portamento.

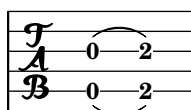
```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Martellato e strappato usando le voci

L'arco del martellato o dello strappato è rivolto in alto nella prima e terza voce, mentre è rivolto in basso nella seconda e quarta voce.

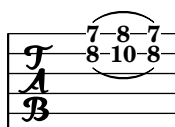
```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Martellato e strappato usando accordi

Quando il martellato o lo strappato si applicano a delle note in un accordo, viene disegnato un solo arco. Ma è possibile avere un “doppio arco” impostando la proprietà `doubleSlurs` su `#t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```



Vedi anche

Guida alla notazione: [Ripetizione di un accordo], pagina 180, [Glissando], pagina 149, [Armonici], pagina 356, [Gambi], pagina 244, [Ripetizioni ricopiate], pagina 171.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

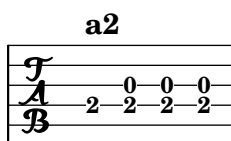
Guida al funzionamento interno: Sezione “TabNoteHead” in *Guida al Funzionamento Interno*, Sezione “TabStaff” in *Guida al Funzionamento Interno*, Sezione “TabVoice” in *Guida al Funzionamento Interno*, Sezione “Beam” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Gli accordi non sono gestiti in un modo specifico, dunque il selettore automatico delle corde potrebbe selezionare la stessa corda per due note di un accordo.

Per poter gestire `\partCombine`, un rigo `TabStaff` deve usare voci create appositamente:

```
melodia = \partCombine { e4 g g g } { e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "one" s1
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



Gli effetti speciali per chitarra sono limitati ad armonici e slide (glissati).

Intavolature personalizzate

L’intavolatura di LilyPond calcola automaticamente il tasto per una nota in base alla corda a cui è assegnata. Per far ciò, deve essere specificata l’accordatura usando la proprietà `stringTunings`.

LilyPond ha accordature predefinite per banjo, mandolino, chitarra, basso, ukulele, violino, viola, violoncello e contrabbasso. LilyPond imposta automaticamente la corretta trasposizione per le accordature predefinite. L’esempio seguente è per basso, che suona un’ottava più bassa di come è scritto.

```
<<
  \new Voice \with {
    \omit StringNumber
  } {
    \clef "bass_8"
    \relative {
      c,4 d e f
    }
  }
  \new TabStaff \with {
    stringTunings = #bass-tuning
  } {
    \relative {
      c,4 d e f
    }
  }
```



L'accordatura predefinita è guitar-tuning, ovvero l'accordatura standard EADGBE. Altre accordature predefinite sono guitar-open-g-tuning, mandolin-tuning e banjo-open-g-tuning. Tutte le accordature predefinite si trovano in `ly/string-tunings-init.ly`.

Si può definire qualsiasi accordatura tramite la funzione `\stringTuning`, che imposta la proprietà `stringTunings` per il contesto corrente.

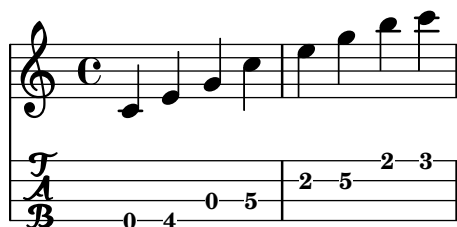
Prende come argomento un accordo che definisce le altezze di ciascuna corda dell'accordatura. Le note dell'accordo sono in modalità di ottava assoluta, vedi [Ottava assoluta], pagina 1. La corda col numero più alto (generalmente la corda più grave) deve apparire per prima nell'accordo. Per esempio, possiamo definire un'accordatura per uno strumento a quattro corde con le altezze a'' , d'' , g' e c' :

```

mynotes = {
  c'4 e' g' c'' |
  e''4 g'' b'' c'''
}

<<
\new Staff {
  \clef treble
  \mynotes
}
\new TabStaff {
  \set Staff.stringTunings = \stringTuning <c' g' d'' a''>
  \mynotes
}
>>

```



La proprietà `stringTunings` viene utilizzata anche da `FretBoards` per calcolare i diagrammi automatici dei tasti.

Le accordature vengono usate per calcolare i diagrammi dei tasti predefiniti (vedi [Diagrammi dei tasti predefiniti], pagina 392).

L'esempio precedente può essere scritto anche così:

```

custom-tuning = \stringTuning <c' g' d'' a''>

mynotes = {

```

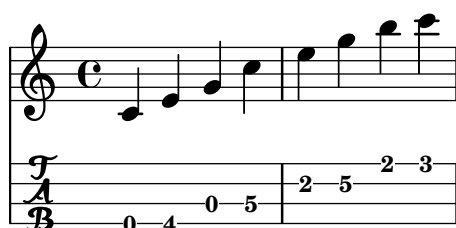


```

c'4 e' g' c'' |
e''4 g'' b'' c'''
}

<<
\new Staff {
  \clef treble
  \mynotes
}
\new TabStaff {
  \set TabStaff.stringTunings = #custom-tuning
  \mynotes
}
>>

```



Internamente un'accordatura è una lista Scheme di altezze, una per ciascuna corda, ordinata da un numero di corda che va da 1 a N, dove la corda 1 è in cima al rigo dell'intavolatura e la corda N è in fondo. Normalmente ciò significa che l'ordine va dall'altezza più alta a quella più bassa, con l'eccezione di alcuni strumenti (come l'ukulele) che non hanno corde ordinate per altezza.

Un'altezza in una lista per accordatura, nella terminologia di LilyPond, è un oggetto altezza, creato dalla funzione Scheme `ly:make-pitch` (vedi Sezione “Scheme functions” in *Guida al Funzionamento Interno*).

`\stringTuning` crea tale oggetto a partire da un accordo.

LilyPond calcola automaticamente il numero di linee del rigo dell'intavolatura (`TabStaff`) e il numero di corde in una tastiera (`FretBoard`) calcolata automaticamente in base al numero di elementi in `stringTunings`.

Per far sì che tutti i contesti `TabStaff` usino la stessa accordatura personalizzata, si può usare

```

\layout {
  \context {
    \TabStaff
    stringTunings = \stringTuning <c' g' d'' a''>
  }
}

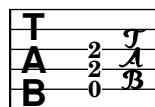
```

Si può usare anche una chiave moderna per l'intavolatura.

```

\new TabStaff {
  \clef moderntab
  <a, e a>1
  \break
  \clef tab
  <a, e a>1
}

```



2



La chiave moderna per l'intavolatura supporta tablature da 4 a 7 corde.

TabStaff supporta i microtoni come i quarti di tono, che possono essere suonati piegando le corde (bending). Occorre impostare `supportNonIntegerFret = ##t` nel contesto `Score`. I microtoni tuttavia non sono supportati nel contesto `FretBoards`.

```
\layout {
  \context {
    \Score
    supportNonIntegerFret = ##t
  }
}
```

```
custom-tuning = \stringTuning <e, a, d ges beh eeh'>
```

```
mus = \relative {
  eeses'4
  eeseh
  ees
  eeh
  e
  eih
  eis
  eisih
  eisis
}
```

```
<<
  \new Staff << \clef "G_8" \mus >>
  \new TabStaff \with { stringTunings = \custom-tuning } \mus
>>
```



Vedi anche

Guida alla notazione: [Ottava assoluta], pagina 1, [Diagrammi dei tasti predefiniti], pagina 392.

File installati: `ly/string-tunings-init.ly`, `scm/tablatore.scm`.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Tab_note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “Scheme functions” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

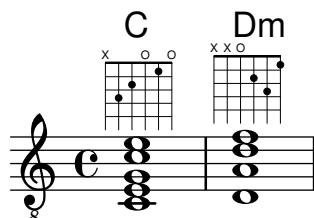
I calcoli automatici dell'intavolatura non funzionano correttamente nella maggior parte dei casi per strumenti (come l'ukulele) in cui le altezze delle corde non variano in modo uniforme col numero di corda.

Diagrammi dei tasti

I diagrammi dei tasti possono essere collegati alla musica come testo (markup) attaccato alla nota desiderata. Il testo contiene l'informazione sul diagramma dei tasti desiderato. Esistono tre diverse interfacce per inserire un diagramma dei tasti (fret-diagram): standard, conciso (terse) e ridondante (verbose). Le tre interfacce producono testi graficamente equivalenti, ma hanno diverse quantità di informazioni nella stringa di testo. I dettagli sulla sintassi delle diverse stringhe di markup usate per definire i diagrammi dei tasti si trovano in Sezione A.12.6 [Instrument-specific markup], pagina 768.

Il comando per il diagramma dei tasti standard, `\fret-diagram`, richiede il numero di corda e il numero di tasto per ogni punto da posizionare sulla corda. Per esempio 5-3 indica che la quinta corda è premuta sul terzo tasto. Si possono indicare anche corde a vuoto o non suonate (mute) attaccando rispettivamente -o e -x al numero di corda.

```
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram "6-x;5-3;4-2;3-o;2-1;1-o;"
    }
    <d a d' f'>1^\markup {
      \fret-diagram "6-x;5-x;4-o;3-2;2-3;1-1;"
    }
  }
}>>
```



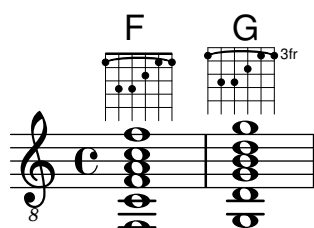
Con fret-diagram si possono aggiungere indicazioni di barré al diagramma inserendo all'inizio della stringa, per esempio, `c:4-1-3`, che indica che il barré va dalla quarta alla prima corda e si trova sul terzo tasto.

```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
```

```

    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram "c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram "c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
    }
  }
}
>>

```

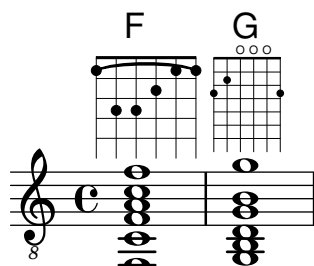


La dimensione del diagramma dei tasti si può variare col modificatore `s:`. Un numero superiore a 1 aumenterà la dimensione predefinita, mentre uno inferiore a 1 la ridurrà. Il numero dei tasti del diagramma può essere cambiato col modificatore `h:` seguito dal numero di tasti desiderato.

```

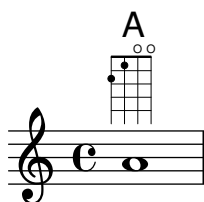
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram "s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram "h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
    }
  }
}
>>

```



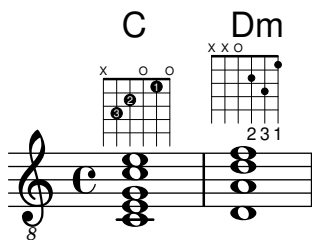
Il numero di corde in un diagramma creato con `\fret-diagram` può essere cambiato per adattarsi a diversi strumenti come banjo e ukulele. Il modificatore è `w:` seguito dal numero di corde desiderato.

```
<<
\new ChordNames {
  \chordmode {
    a1
  }
}
\new Staff {
  % An 'A' chord for ukulele
  a'1^\markup {
    \fret-diagram "w:4;4-2-2;3-1-1;2-o;1-o;"
  }
}
>>
```



Si possono aggiungere indicazioni di diteggiatura appendendo un terzo numero, dopo quelli della corda e del tasto. La posizione delle diteggiature è controllata dal modificatore `f:`, che può prendere due valori: 1, per mostrare i numeri di diteggiatura sul tasto; 2 per mostrarli alla base della rispettiva corda (posizione predefinita).

```
<<
\new ChordNames {
  \chordmode {
    c1 d:m
  }
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram "f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram "f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
  }
}
>>
```



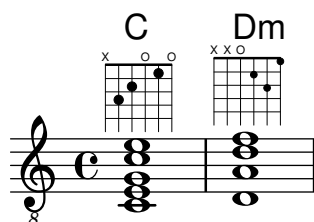
Il comando `fret-diagram` consente anche di modificare il raggio del punto (modificatore `d:`) e la sua posizione rispetto al tasto (modificatore `p:`).

```
<<
```

```

\new ChordNames {
  \chordmode {
    c1 d:m
  }
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {
    \fret-diagram "d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram "p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
  }
}
>>

```



I diagrammi dei tasti possono essere adattati per i mancini

```

\markup
\center-column {
  "C"
  "(left-handed)"
  \override #`(fret-diagram-details . ((handedness . ,LEFT)))
  \fret-diagram "6-x;5-3-3;4-2-2;3-o;2-1;1-o;"
}

```

C
(left-handed)



Il comando markup `\fret-diagram-terse` ha un formato più conciso perché omette i numeri di corda, che sono calcolati implicitamente in base al punto e virgola che separa ciascuna corda. Il primo punto e virgola corrisponde al numero di corda più alto e l'ultimo punto e virgola corrisponde alla prima corda. Si possono indicare anche corde a vuoto, mute e con tasto premuto, come visto prima.

```

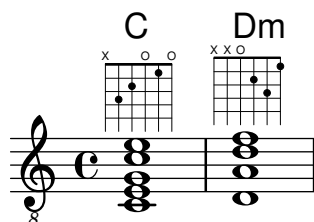
<<
\new ChordNames {
  \chordmode {
    c1 d:m
  }
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1^\markup {

```

```

    \fret-diagram-terse "x;3;2;o;1;o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram-terse "x;x;o;2;3;1;"
  }
}
>>

```

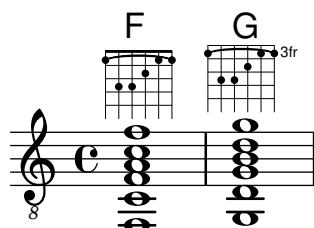


Nella stringa di `\fret-diagram-terse` si possono inserire indicazioni di barré:

```

<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1^\markup {
      \fret-diagram-terse "1-(;3;3;2;1;1-);"
    }
    <g, d g b d' g'>1^\markup {
      \fret-diagram-terse "3-(;5;5;4;3;3-);"
    }
  }
}
>>

```



`\fret-diagram-terse` accetta anche indicazioni di diteggiatura, che si troveranno in seconda posizione, dopo il numero del tasto.

```

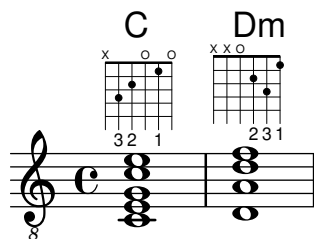
<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
    \clef "treble_8"
    <c e g c' e'>1^\markup {

```

```

    \fret-diagram-terse "x;3-3;2-2;o;1-1;o;"
  }
  <d a d' f'>1^\markup {
    \fret-diagram-terse "x;x;o;2-2;3-3;1-1;"
  }
}
>>

```



Quando si usa `\fret-diagram-terse`, altre proprietà del diagramma dei tasti possono essere regolate usando `\override`.

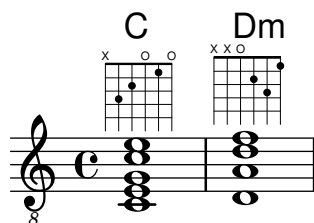
Solo un'indicazione per corda può essere inclusa in un markup `fret-diagram-terse`. Per avere indicazioni multiple per corda, usare una diagramma dei tasti o un markup `fret-diagram-verbose`.

Il comando `\fret-diagram-verbose` prende come argomento una lista Scheme. Ciascun elemento della lista indica un elemento da posizionare sul diagramma.

```

<<
  \new ChordNames {
    \chordmode {
      c1 d:m
    }
  }
  \new Staff {
    \clef "treble_8"
    <c e g c' e'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    }
    <d a d' f'>1^\markup {
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
      )
    }
  }
}
>>

```

Nella lista di `fret-diagram-verbose` si possono inserire indicazioni di diteggiatura e barré. Solo in questa interfaccia è possibile inserire un'indicazione di capotasto per il diagramma. L'indicazione di capotasto è una barra spessa che copre tutte le corde. Il tasto col capotasto sarà il tasto più basso del diagramma.

I punti con le diteggiature possono essere colorati o racchiusi da parentesi; il colore delle parentesi può essere modificato in modo indipendente.

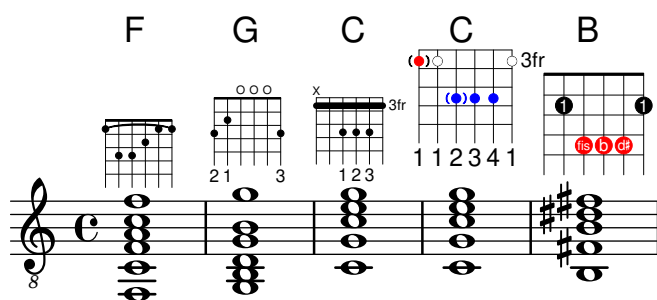
Si può anche inserire del testo di tipo markup dentro i punti.

```
<<
  \new ChordNames {
    \chordmode {
      f1 g c c b
    }
  }
  \new Staff {
    \clef "treble_8"
    \override Voice.TextScript.fret-diagram-details.finger-code = #'below-string
    <f, c f a c' f'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    }
    <g, b, d g b g'>1^\markup {
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
      )
    }
    <c g c' e' g'>1^\markup {
      \fret-diagram-verbose #'(
        (capo 3)
        (mute 6)
        (place-fret 4 5 1)
        (place-fret 3 5 2)
        (place-fret 2 5 3)
      )
    }
  }
```

```

    )
  }
  \override Voice.TextScript.size = 1.4
  <c g c' e' g'>1^\markup {
    \fret-diagram-verbose #'(
      (place-fret 6 3 1 red parenthesized default-paren-color)
      (place-fret 5 3 1 inverted)
      (place-fret 4 5 2 blue parenthesized)
      (place-fret 3 5 3 blue)
      (place-fret 2 5 4 blue)
      (place-fret 1 3 1 inverted)
    )
  }
  \override Voice.TextScript.size = 1.5
  <b, fis b dis' fis'>1^\markup
    \override #'(fret-diagram-details . ((finger-code . in-dot)))
    \fret-diagram-verbose #`(
      (place-fret 5 2 1)
      (place-fret 4 4 "fis" red)
      (place-fret 3 4 "b" red)
      (place-fret
        2 4
        ,#{ \markup
          \concat {
            \vcenter "d"
            \fontsize #-5
            \musicglyph "accidentals.sharp"} #}
        red)
      (place-fret 1 2 1)
    )
  }
  >>

```



Anche in questa interfaccia è possibile modificare tutte le altre proprietà del diagramma dei tasti tramite `\override`.

L'aspetto grafico di un diagramma dei tasti può essere personalizzato secondo le proprie preferenze attraverso le proprietà dell'interfaccia `fret-diagram-interface`. Tutti i dettagli si trovano in Sezione “`fret-diagram-interface`” in *Guida al Funzionamento Interno*. Per un diagramma dei tasti di tipo `markup`, le proprietà dell'interfaccia appartengono a `Voice.TextScript`.

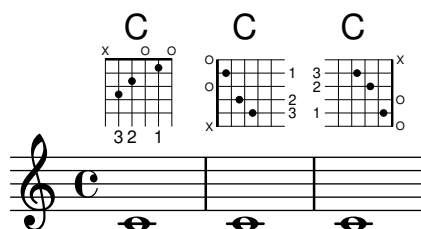
Frammenti di codice selezionati

Cambiare l'orientamento della tastiera

I diagrammi dei tasti possono essere orientati in tre modi.

```
\include "predefined-guitar-fretboards.ly"

<<
\chords {
  c1
  c1
  c1
}
\new FretBoards \chordmode {
  c1
  \override FretBoard.fret-diagram-details.orientation =
    #'landscape
  c1
  \override FretBoard.fret-diagram-details.orientation =
    #'opposing-landscape
  c1
}
\new Voice {
  c'1
  c'1
  c'
}
>>
```



Personalizzare il diagramma dei tasti di tipo markup

Le proprietà del diagramma dei tasti si possono impostare tramite 'fret-diagram-details'. Per diagrammi di tipo markup, gli override possono essere applicati all'oggetto Voice.TextScript o direttamente al markup.

```
<<
\chords { c1 | c | c | d }

\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript.size = 1.2
  \override TextScript.fret-diagram-details.finger-code = #'in-dot
  \override TextScript.fret-diagram-details.dot-color = #'white

  %% C major for guitar, no barre, using defaults
  % terse style
  c'1~\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }
}
```

```

%% C major for guitar, barred on third fret
% verbose style
% size 1.0
% roman fret label, finger labels below string, straight barre
c'1~\markup {
  % standard size
  \override #'(size . 1.0) {
    \override #'(fret-diagram-details . (
      (number-type . roman-lower)
      (finger-code . in-dot)
      (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
        (place-fret 5 3 1)
        (place-fret 4 5 2)
        (place-fret 3 5 3)
        (place-fret 2 5 4)
        (place-fret 1 3 1)
        (barre 5 1 3))
    }
  }
}

```

```

%% C major for guitar, barred on third fret
% verbose style
% landscape orientation, arabic numbers, M for mute string
% no barre, fret label down or left, small mute label font
c'1~\markup {
  \override #'(fret-diagram-details . (
    (finger-code . below-string)
    (number-type . arabic)
    (label-dir . -1)
    (mute-string . "M")
    (orientation . landscape)
    (barre-type . none)
    (xo-font-magnification . 0.4)
    (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
      (place-fret 5 3 1)
      (place-fret 4 5 2)
      (place-fret 3 5 3)
      (place-fret 2 5 4)
      (place-fret 1 3 1)
      (barre 5 1 3))
    }
  }
}

```

```

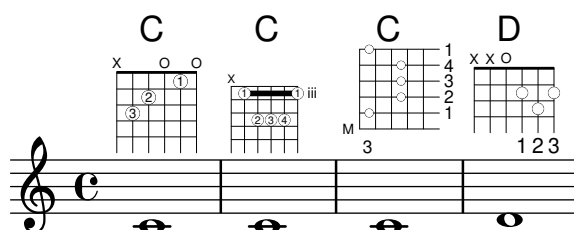
%% simple D chord
% terse style
% larger dots, centered dots, fewer frets
% label below string
d'1~\markup {
  \override #'(fret-diagram-details . (

```

```

(finger-code . below-string)
(dot-radius . 0.35)
(dot-position . 0.5)
(fret-count . 3))) {
\ fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
}
}
}
>>

```



Vedi anche

Guida alla notazione: Sezione A.12.6 [Instrument-specific markup], pagina 768.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “fret-diagram-interface” in *Guida al Funzionamento Interno*.

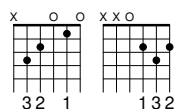
Diagrammi dei tasti predefiniti

I diagrammi dei tasti possono essere visualizzati attraverso il contesto `FretBoards`. Per impostazione predefinita, il contesto `FretBoards` mostrerà i diagrammi dei tasti salvati in una tabella dati:

```

\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode {
    c1 d
  }
}

```



I diagrammi dei tasti predefiniti sono contenuti nel file `predefined-guitar-fretboards.ly`. Si basano sulle altezze di un accordo e sul valore di `stringTunings` in uso. `predefined-guitar-fretboards.ly` contiene i diagrammi dei tasti predefiniti solo per l'accordatura standard per chitarra (`guitar-tuning`). Si possono aggiungere diagrammi predefiniti per altri strumenti o accordature seguendo l'esempio in `predefined-guitar-fretboards.ly`.

I diagrammi dei tasti per ukulele sono contenuti nel file `predefined-ukulele-fretboards.ly`.

```

\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

\new ChordNames {
  \myChords
}

```

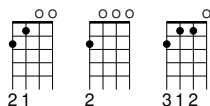
```

}

\new FretBoards {
  \set Staff.stringTunings = #ukulele-tuning
  \myChords
}

```

A Am A+



I diagrammi dei tasti per mandolino sono contenuti nel file `predefined-mandolin-fretboards.ly`.

```

\include "predefined-mandolin-fretboards.ly"

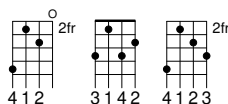
myChords = \chordmode { c1 c:m7.5- c:aug }

\new ChordNames {
  \myChords
}

\new FretBoards {
  \set Staff.stringTunings = #mandolin-tuning
  \myChords
}

```

C C[∅] C+

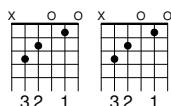


Le altezze degli accordi possono essere inserite sia come musica simultanea sia usando la modalità per accordi (vedi [Panoramica sulla modalità accordo], pagina 432).

```

\include "predefined-guitar-fretboards.ly"
\new FretBoards {
  \chordmode { c1 }
  <c' e' g'>1
}

```



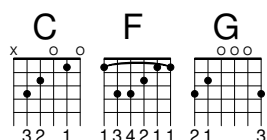
Di solito i nomi degli accordi e i diagrammi dei tasti appaiono insieme. Per farlo si pone un contesto `ChordNames` in parallelo con un contesto `FretBoards` assegnando a entrambi la stessa musica.

```

\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 f g
}

```

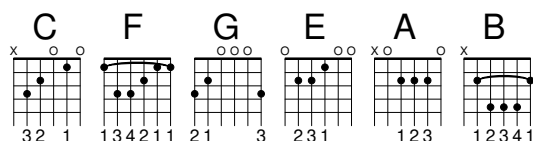
```
<<
\new ChordNames {
  \mychords
}
\new FretBoards {
  \mychords
}
>>
```



I diagrammi dei tasti predefiniti possono essere trasposti, purché un diagramma per l'accordo trasposto sia salvato nella tabella del diagramma.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode {
  c1 f g
}

mychordlist = {
  \mychords
  \transpose c e { \mychords }
}
<<
\new ChordNames {
  \mychordlist
}
\new FretBoards {
  \mychordlist
}
>>
```

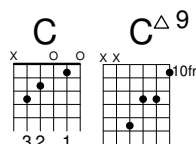


La tabella dei diagrammi dei tasti predefiniti per chitarra contiene otto accordi (maggiore, minore, aumentato, diminuito, settima dominante, settima maggiore, settima minore, nona dominante) per ognuna delle 17 tonalità.

La tabella dei diagrammi dei tasti per ukulele contiene questi accordi più ulteriori tre accordi (sesta maggiore, seconda sospesa e quarta sospesa). Un elenco completo dei diagrammi dei tasti predefiniti si trova in Sezione A.4 [Diagrammi degli accordi predefiniti], pagina 682. Se nella tabella non esiste una voce per un accordo, l'incisore FretBoards calcolerà un diagramma usando la funzionalità automatica descritta in [Diagrammi dei tasti automatici], pagina 403.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 c:maj9
}
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



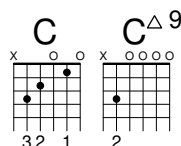
Si possono aggiungere diagrammi dei tasti alla tabella. Basta specificare la tabella del diagramma, l'accordo, l'accordatura e una definizione del diagramma. Di solito la tabella di riferimento sarà *default-fret-table*. La definizione di diagramma può essere una stringa di fret-diagram-terse o un elenco di fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
  \chordmode { c:maj9 }
  #guitar-tuning
  "x;3-2;o;o;o;o;"

mychords = \chordmode {
  c1 c:maj9
}

<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Diagrammi diversi per accordi con lo stesso nome possono essere salvati usando un'altezza con un'ottava diversa. L'ottava diversa deve essere almeno due ottave sopra o sotto l'ottava predefinita, perché le ottave sopra e sotto sono usate per trasporre le tastiere.

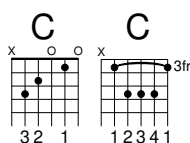
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table
  \chordmode { c'' }
  #guitar-tuning
  #(offset-fret 2 (chord-shape 'bes guitar-tuning))
```



```
mychords = \chordmode {
  c1 c''
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
>>
```



Oltre ai diagrammi dei tasti, LilyPond contiene un elenco interno di forme di accordo. Queste forme sono diagrammi dei tasti che possono essere spostati lungo il manico in diverse posizioni per fornire diversi accordi. Si possono aggiungere nuove forme di accordo all'elenco interno e poi usarle per definire diagrammi dei tasti predefiniti. Dato che possono essere spostate in varie posizioni sul manico, le forme di un accordo normalmente non contengono corde a vuoto. Come per i diagrammi dei tasti, le forme possono essere inserite come stringhe di fret-diagram-terse o come elenchi di fret-diagram-verbose.

```
\include "predefined-guitar-fretboards.ly"
```

```
% Add a new chord shape
```

```
\addChordShape #'powerf #guitar-tuning "1-1;3-3;3-4;x;x;x;"
```

```
% add some new chords based on the power chord shape
```

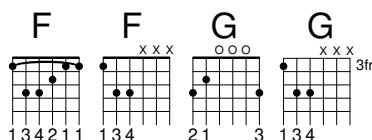
```
\storePredefinedDiagram #default-fret-table
  \chordmode { f'' }
  #guitar-tuning
  #(chord-shape 'powerf guitar-tuning)
```

```
\storePredefinedDiagram #default-fret-table
  \chordmode { g'' }
  #guitar-tuning
  #(offset-fret 2 (chord-shape 'powerf guitar-tuning))
```

```
mychords = \chordmode {
  f1 f'' g g''
}
```

```
<<
  \new ChordNames {
    \mychords
  }
  \new FretBoards {
    \mychords
  }
```

```
}
>>
```



L'aspetto grafico di un diagramma dei tasti può essere personalizzato in base alle proprie preferenze attraverso le proprietà dell'interfaccia `fret-diagram-interface`. I dettagli si trovano in Sezione “`fret-diagram-interface`” in *Guida al Funzionamento Interno*. Per un diagramma dei tasti predefinito, le proprietà dell'interfaccia appartengono a `FretBoards.FretBoard`.

Frammenti di codice selezionati

Personalizzare la tastiera del diagramma dei tasti

Le proprietà del diagramma dei tasti possono essere impostate tramite '`fret-diagram-details`'. Per i diagrammi dell'oggetto `FretBoard`, gli override vengono applicati all'oggetto `FretBoards.FretBoard`. Come `Voice`, `FretBoards` è un contesto di basso livello, dunque può essere omesso negli override delle proprietà.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        "x;1-1-(;3-2;3-3;3-4;1-1-);"

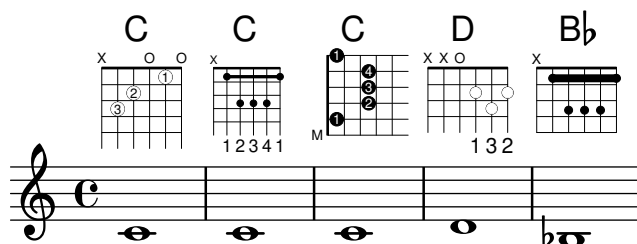
% shorthand
oo = #(define-music-function
      (grob-path value)
      (list? scheme?)
      #{ \once \override $grob-path = #value #})

<<
\new ChordNames {
  \chordmode { c1 | c | c | d | bes }
}
\new FretBoards {
  % Set global properties of fret diagram
  \override FretBoards.FretBoard.size = 1.2
  \override FretBoard.fret-diagram-details.finger-code = #'in-dot
  \override FretBoard.fret-diagram-details.dot-color = #'white
  \chordmode {
    c
    \oo FretBoard.size #1.0
    \oo FretBoard.fret-diagram-details.barre-type #'straight
    \oo FretBoard.fret-diagram-details.dot-color #'black
    \oo FretBoard.fret-diagram-details.finger-code #'below-string
    c'
    \oo FretBoard.fret-diagram-details.barre-type #'none
    \oo FretBoard.fret-diagram-details.number-type #'arabic
    \oo FretBoard.fret-diagram-details.orientation #'landscape
    \oo FretBoard.fret-diagram-details.mute-string "M"
```

```

\oo FretBoard.fret-diagram-details.label-dir #LEFT
\oo FretBoard.fret-diagram-details.dot-color #'black
c'
\oo FretBoard.fret-diagram-details.finger-code #'below-string
\oo FretBoard.fret-diagram-details.dot-radius #0.35
\oo FretBoard.fret-diagram-details.dot-position #0.5
\oo FretBoard.fret-diagram-details.fret-count #3
d
\oo FretBoard.fret-diagram-details.barre-type #'straight
\oo FretBoard.fret-diagram-details.finger-code #'none
\oo FretBoard.fret-diagram-details.dot-radius #0.25
\oo FretBoard.fret-diagram-details.dot-color #'black
\oo FretBoard.fret-diagram-details.string-overhang #0.
\oo FretBoard.fret-diagram-details.barre-thickness #2.
bes
}
}
\new Voice {
  c'1 | c' | c' | d' | bes
}
>>

```



Setting up predefined fretboards for other instruments

Predefined fret diagrams can be added for new instruments in addition to the standard diagrams used for guitar. This file shows how this is done by defining a new string tuning and a few predefined fretboards for the Venezuelan *cuatro*.

This file also shows how fingerings can be included in the chords used as reference points for the chord lookup, and displayed in the fret diagram and the TabStaff, but not the music.

These fretboards are not transposable because they contain string information. This is planned to be corrected in the future.

```

% Add fretboards for the cuatro.
%
% Note: This section could be put into a separate file
%       `predefined-cuatro-fretboards.ly`
%       and be \included into each of your compositions.

```

```

cuatroTuning = #`((ly:make-pitch 0 6 0)
                ,(ly:make-pitch 1 3 SHARP)
                ,(ly:make-pitch 1 1 0)
                ,(ly:make-pitch 0 5 0))

```

```

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }

```

```

aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
    #cuatroTuning
    "o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
    #cuatroTuning
    "o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
    #cuatroTuning
    "o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
    #cuatroTuning
    "o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
    #cuatroTuning
    "2-2;o;1-1;o;"

% End of potential include file `predefined-cuatro-fretboards.ly`.

#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }
  }

  \new FretBoards {
    \set Staff.stringTunings = #cuatroTuning
  }
}

```

```

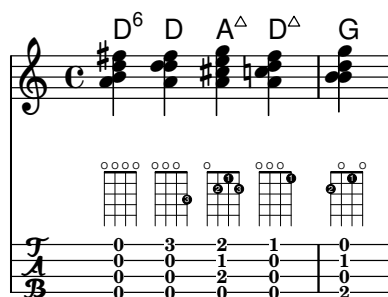
%      \override FretBoard
%      #'(fret-diagram-details string-count) = 4
\override FretBoard.fret-diagram-details.finger-code = #'in-dot
\primeros
}

\new TabStaff \relative c'' {
  \set TabStaff.stringTunings = #cuatroTuning
  \primeros
}

>>

\layout {
  \context {
    \Score
    \override SpacingSpanner.base-shortest-duration =
      \musicLength 16
  }
}
\midi { }
}

```



Chord changes for fretboards

Fretboards can be set to display only when the chord changes, or at the beginning of a new line.

```

\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
\new ChordNames { \myChords }
\new FretBoards { \myChords }
\new Staff { \myChords }
>>

```

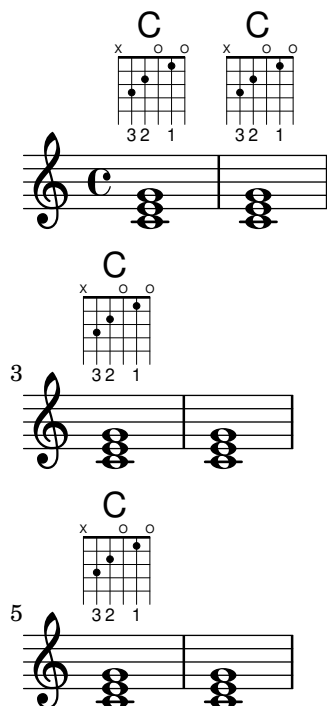


Tabelle alternative per i diagrammi dei tasti

Si possono creare tabelle alternative per i diagrammi dei tasti. Queste possono essere usate per avere diagrammi alternativi per uno stesso accordo.

Per poter usare una tabella alternativa, deve prima essere creata. Quindi si aggiungono i diagrammi alla tabella.

La tabella può essere vuota oppure ricopiata da una tabella esistente.

La tabella da usare nel mostrare i diagrammi predefiniti viene selezionata dalla proprietà `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table.
#(define custom-fretboard-table-one
  (make-fretboard-table))

% Make a new fretboard table as a copy of `default-fret-table`.
#(define custom-fretboard-table-two
  (make-fretboard-table default-fret-table))

% Add a chord to `custom-fretboard-table-one`.
\storePredefinedDiagram #custom-fretboard-table-one
  \chordmode {c}
  #guitar-tuning
  "3-(;3;5;5;5;3-);"

% Add a chord to `custom-fretboard-table-two`.
\storePredefinedDiagram #custom-fretboard-table-two
  \chordmode {c}
  #guitar-tuning
  "x;3;5;5;5;o;"
```

```

<<
\chords {
  c1 | d1 |
  c1 | d1 |
  c1 | d1 |
}
\new FretBoards {
  \chordmode {
    \set predefinedDiagramTable = #default-fret-table
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-one
    c1 | d1 |
    \set predefinedDiagramTable = #custom-fretboard-table-two
    c1 | d1 |
  }
}
\new Staff {
  \clef "treble_8"
  <<
    \chordmode {
      c1 | d1 |
      c1 | d1 |
      c1 | d1 |
    }
    {
      s1_\markup "Default table" | s1 |
      s1_\markup \column { "New table" "from empty" } | s1 |
      s1_\markup \column { "New table" "from default" } | s1 |
    }
  >>
}
>>

```

Diagram illustrating the output of the provided LilyPond code. It shows three measures of music on a staff, each with a corresponding fretboard diagram above it. The first measure is labeled 'Default table' and shows a C major chord. The second measure is labeled 'New table from empty' and shows a D major chord. The third measure is labeled 'New table from default' and shows a C major chord. The fretboard diagrams show the fingerings for each chord: C major (3, 2, 1), D major (1, 3, 2), and C major (3, 2, 1).

Vedi anche

Guida alla notazione: [Intavolature personalizzate], pagina 378, [Diagrammi dei tasti automatici], pagina 403, [Panoramica sulla modalità accordo], pagina 432, Sezione A.4 [Diagrammi degli accordi predefiniti], pagina 682.

File installati: ly/predefined-guitar-fretboards.ly,
 ly/predefined-guitar-ninth-fretboards.ly,
 ly/predefined-ukulele-fretboards.ly,
 ly/predefined-mandolin-fretboards.ly.

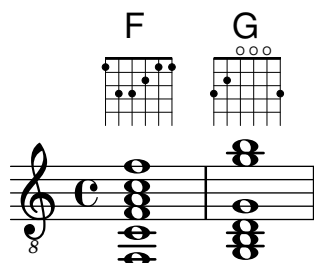
Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “fret-diagram-interface” in *Guida al Funzionamento Interno*.

Diagrammi dei tasti automatici

I diagrammi dei tasti possono essere creati automaticamente dalle note inserite usando il contesto FretBoards. Se non è disponibile un diagramma predefinito per le note inserite nell'accordatura (stringTunings) attiva, questo contesto calcola le corde e i tasti che possono essere usati per suonare le note.

```
<<
  \new ChordNames {
    \chordmode {
      f1 g
    }
  }
  \new FretBoards {
    <f, c f a c' f'>1
    <g,\6 b, d g b g'>1
  }
  \new Staff {
    \clef "treble_8"
    <f, c f a c' f'>1
    <g, b, d g b' g'>1
  }
>>
```



Dato che nessun diagramma predefinito viene caricato senza impostarlo esplicitamente, il calcolo automatico dei diagrammi è il comportamento predefinito. Quando i diagrammi predefiniti vengono caricati, il calcolo automatico può essere abilitato e disabilitato con dei comandi predefiniti:

```
\storePredefinedDiagram #default-fret-table
                                <c e g c' e'>
                                #guitar-tuning
                                "x;3-1-(;5-2;5-3;5-4;3-1-1-);"
```

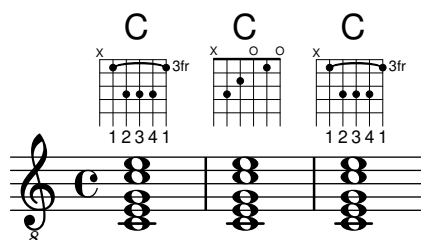
```
<<
  \new ChordNames {
    \chordmode {
      c1 c c
    }
  }
  \new FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>1
```



```

\predefinedFretboardsOn
<c e g c' e'>1
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <c e g c' e'>1
  <c e g c' e'>1
}
>>

```



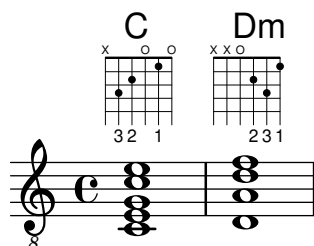
Talvolta il calcolatore dei diagrammi non sarà in grado di trovare un diagramma accettabile. Di solito il rimedio consiste nell'assegnare manualmente una nota a una corda. In molti casi, solo una nota deve essere posta esplicitamente su una corda; tutte le altre note verranno posizionate correttamente dal contesto FretBoards.

Si possono aggiungere le diteggiture ai diagrammi dei tasti dell'oggetto FretBoard.

```

<<
\new ChordNames {
  \chordmode {
    c1 d:m
  }
}
\new FretBoards {
  <c-3 e-2 g c'-1 e'>1
  <d a-2 d'-3 f'-1>1
}
\new Staff {
  \clef "treble_8"
  <c e g c' e'>1
  <d a d' f'>1
}
>>

```



Il tasto minimo da usare per calcolare le corde e i tasti per il contesto FretBoard si può impostare con la proprietà `minimumFret`.

```

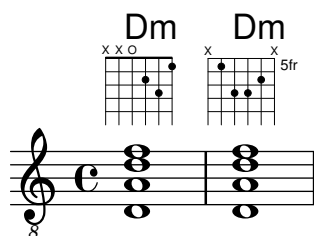
<<
\new ChordNames {

```

```

\chordmode {
  d1:m d:m
}
}
\new FretBoards {
  <d a d' f'>1
  \set FretBoards.minimumFret = #5
  <d a d' f'>1
}
\new Staff {
  \clef "treble_8"
  <d a d' f'>1
  <d a d' f'>1
}
>>

```



Le corde e i tasti per il contesto `FretBoards` dipendono dalla proprietà `stringTunings`, che funziona proprio come nel contesto `TabStaff`. Si rimanda a [Intavolature personalizzate], pagina 378, per informazioni sulla proprietà `stringTunings`.

L'aspetto grafico di un diagramma dei tasti può essere personalizzato secondo le proprie preferenze tramite le proprietà dell'interfaccia `fret-diagram-interface`. I dettagli si trovano in Sezione “fret-diagram-interface” in *Guida al Funzionamento Interno*. Per un diagramma dei tasti del contesto `FretBoards`, le proprietà dell'interfaccia appartengono a `FretBoards.FretBoard`.

Comandi predefiniti

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Vedi anche

Guida alla notazione: [Intavolature personalizzate], pagina 378.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “fret-diagram-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

I calcoli automatici dei tasti non funzionano correttamente per gli strumenti con accordature non monotoniche.

Diteggiature della mano destra

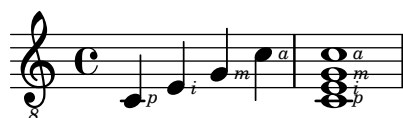
Le diteggiature della mano destra *p-i-m-a* si inseriscono col comando `\rightHandFinger` seguito da un numero.

Nota: Se si inserisce il numero in notazione Scheme, ricordarsi di aggiungere uno spazio prima del `>` di chiusura o simili.

```

\clef "treble_8"
c4\rightHandFinger #1
e\rightHandFinger #2
g\rightHandFinger #3
c'\rightHandFinger #4
<c\rightHandFinger #1 e\rightHandFinger #2
g\rightHandFinger #3 c'\rightHandFinger #4 >1

```



Per comodità, si può abbreviare `\rightHandFinger` in un comando più breve, per esempio RH, aggiungendo la definizione nel livello più alto del file sorgente:

```
RH = \rightHandFinger \etc
```

La maggior parte dei comportamenti delle diteggiature della mano destra (oggetto `StrokeFinger`) si impostano nello stesso modo delle normali diteggiature: vedi [Indicazioni di diteggiatura], pagina 235.

Frammenti di codice selezionati

Posizionamento delle diteggiature della mano destra

È possibile avere un maggior controllo sul posizionamento delle diteggiature della mano destra impostando una specifica proprietà, come illustrato nell'esempio seguente.

```

#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

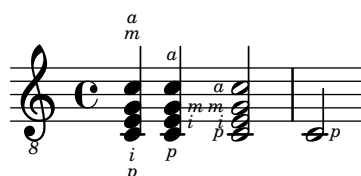
  \set strokeFingerOrientations = #'(up down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >4

  \set strokeFingerOrientations = #'(left)
  <c\RH 1 e\RH 2 g\RH 3 c\RH 4 >2

  \set strokeFingerOrientations = #'(right)
  c\RH 1
}

```

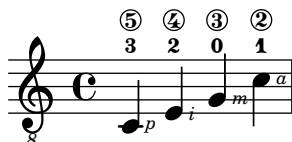


Diteggiature, indicazioni di corda e diteggiature della mano destra

Questo esempio combina la diteggiatura per la mano sinistra, le indicazioni di stringa e la diteggiatura della mano destra.

```
#(define RH rightHandFinger)
```

```
\relative c {
  \clef "treble_8"
  <c-3\5\RH 1 >4
  <e-2\4\RH 2 >4
  <g-0\3\RH 3 >4
  <c-1\2\RH 4 >4
}
```



Vedi anche

Guida alla notazione: [Indicazioni di diteggiatura], pagina 235.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “StrokeFinger” in *Guida al Funzionamento Interno*.

2.4.2 Chitarra

La maggior parte delle questioni relative alla musica per chitarra sono trattate nella sezione generale sugli strumenti a corde con tasti, ma ce ne sono altre che è meglio spiegare qui. Talvolta gli utenti desiderano creare dei canzonieri, ovvero dei documenti che hanno solo il testo e le indicazioni di accordo sopra di esso. Dal momento che LilyPond è un compositore tipografico musicale, non è consigliato per documenti che non contengono notazione musicale. Migliori alternative sono un programma di videoscrittura, un editor di testo oppure, per utenti esperti, un compositore tipografico come GuitarTeX.

Indicazione di posizione e barré

Questo esempio illustra come mostrare la posizione sulla chitarra e le indicazioni di barré.

```
\relative {
  \clef "treble_8"
  b,16 d g b e
  \textSpannerDown
  \override TextSpanner.bound-details.left.text = "XII "
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
  e16 b g d
}
```



Vedi anche

Guida alla notazione: [Estensori del testo], pagina 254.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*, Sezione “Expressive marks” in *Frammenti di codice*.

Indicazione di armonici e note smorzate

Si possono usare teste di nota speciali per indicare note smorzate o armonici. Gli armonici sono solitamente accompagnati da testo esplicativo.

```
\relative {
  \clef "treble_8"
  \override NoteHead.style = #'harmonic-mixed
  d'8^\markup { \italic \fontsize #-2 "harm. 12" } <g b>4
}
```



Le note smorzate (chiamate anche *dead note*) sono supportate sia sul rigo normale che sull'intavolatura:

```
music = \relative {
  < a\3 \deadNote c\2 a'\1 >4
  < b\3 \deadNote d\2 b'\1 >
  < c\3 \deadNote e\2 c'\1 >
  \deadNotesOn
  \tuplet 3/2 { g8 b e }
  \deadNotesOff
  < a,\3 c\2 e\1 >1
}
\new StaffGroup <<
  \new Staff {
    \clef "treble_8"
    \music
  }
  \new TabStaff {
    \music
  }
>>
```

Un'altra tecnica esecutiva (usata specialmente nella chitarra elettrica) è il *palm mute*, in cui la corda viene parzialmente silenziata dal palmo della mano che la colpisce (da qui il nome). LilyPond supporta la notazione delle note in stile palm mute cambiando la testa della nota in una forma triangolare.

```
\new Voice { % Attenzione: la creazione esplicita di una Voice è
              % necessaria per far sì che palmMuteOff funzioni correttamente
```

```

%      quando palmMuteOn si trova all'inizio del brano.
\relative c, {
  \clef "G_8"
  \palmMuteOn
  e8^\markup { \musicglyph "noteheads.s2do" = palm mute }
  < e b' e > e
  \palmMuteOff
  e e \palmMute e e e |
  e8 \palmMute { e e e } e e e e |
  < \palmMute e b' e >8 \palmMute { e e e } < \palmMute e b' e >2
}
}

```



Vedi anche

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

Guida alla notazione: [Teste di nota speciali], pagina 41, Sezione A.9 [Stili delle teste di nota], pagina 707.

Indicazione di power chord

I “power chord” e i loro simboli possono essere inseriti in modalità accordo o come costrutti di accordo. In via eccezionale, nei nomi di questi accordi viene specificato il quinto grado, che è solitamente omesso negli altri accordi (per esempio, le triadi maggiori o minori:

```

ChordsAndSymbols = {
  \chordmode {
    e,,1:5
    a,,5.8
    \set TabStaff.restrainOpenStrings = ##t
    \set minimumFret = #8
    c,:5
    f,:5.8
  }
  \set minimumFret = #2
  \set restrainOpenStrings = ##f
  <a, e> <a cis' e'>
  <g d' g'>
}
\score {
  <<
    \new ChordNames {
      \ChordsAndSymbols
    }
    \new Staff {
      \clef "treble_8"
      \ChordsAndSymbols
    }
    \new TabStaff {

```

```

\ChordsAndSymbols
}
>>
}

```

Vedi anche

Glossario musicale: Sezione “power chord” in *Glossario Musicale*.

Guida alla notazione: [Accordi estesi e alterati], pagina 434, [Stampa dei nomi degli accordi], pagina 437.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

2.4.3 Banjo

Intavolature per banjo

LilyPond ha un supporto basilare per il banjo a cinque corde. Quando si scrivono intavolature per banjo a cinque corde, usare la funzione di formattazione dell’intavolatura per banjo per ottenere i numeri di tasto corretti per la quinta corda:

```

music = {
  g8 d' g'\5 a b g e d' |
  g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
  g4
}

<<
\new Staff \with { \omit StringNumber }
{ \clef "treble_8" \music }
\new TabStaff \with {
  tablatureFormat = #fret-number-tablature-format-banjo
  stringTunings = #banjo-open-g-tuning
}
{ \music }
>>

```

Sono disponibili varie accordature comuni per banjo a cinque corde: *banjo-open-g-tuning* (gDGBD), *banjo-c-tuning* (gCGBD), *banjo-modal-tuning* (gDGCD), *banjo-open-d-tuning*

(aDF#AD), banjo-open-dm-tuning (aDFAD), banjo-double-c-tuning (gCGCD) e banjo-double-d-tuning (aDGDE).

Queste possono essere convertite in accordature a quattro corde usando la funzione `four-string-banjo`:

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Vedi anche

File installati: `ly/string-tunings-init.ly`.

Frammenti: Sezione “Fretted string instruments” in *Frammenti di codice*.

2.4.4 Liuto

Intavolature per liuto

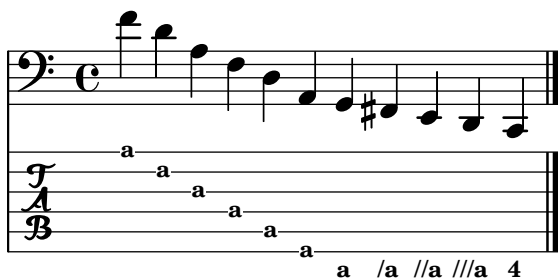
LilyPond supporta l'intavolatura per liuto.

Per aggiungere altre corde basse usare `additionalBassStrings`, dove si impostano le altezze di queste corde. Appariranno sotto la linea più bassa come: `a`, `/a`, `//a`, `///a`, `4`, `5`, etc.

Occorre assegnare `fret-letter-tablature-format` a `tablatureFormat` e probabilmente usare anche `fretLabels` per ulteriori personalizzazioni.

```
m = { f'4 d' a f d a, g, fis, e, d, c, \bar "|" }

\score {
  <<
    \new Staff { \clef bass \cadenzaOn \m }
    \new TabStaff \m
  >>
  \layout {
    \context {
      \Score
      tablatureFormat = #fret-letter-tablature-format
    }
    \context {
      \TabStaff
      stringTunings = \stringTuning <a, d f a d' f'>
      additionalBassStrings = \stringTuning <c, d, e, fis, g,>
      fretLabels = #'("a" "b" "r" "d" "e" "f" "g" "h" "i" "k")
    }
  }
}
```



Problemi noti e avvertimenti

L'uso di `FretBoards` insieme a `additionalBassStrings` non è supportato e produrrà risultati insoddisfacenti.

2.5 Percussioni

2.5.1 Notazione comune per le percussioni

La musica ritmica viene usata principalmente per la notazione di strumenti percussivi, ma può essere usata anche per mostrare i ritmi delle melodie.

Riferimenti per percussioni

- Alcune percussioni possono essere scritte su un rigo ritmico; questo argomento è trattato in [Mostrare i ritmi della melodia], pagina 86, e [Istanziare nuovi righi], pagina 202.
- l'output MIDI è trattato in un'altra sezione, vedi Sezione 3.5 [Creazione dell'output MIDI], pagina 537.

Vedi anche

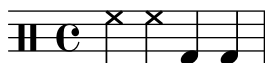
Guida alla notazione: [Mostrare i ritmi della melodia], pagina 86, [Istanziare nuovi righi], pagina 202, Sezione 3.5 [Creazione dell'output MIDI], pagina 537.

Frammenti: Sezione "Percussion" in *Frammenti di codice*.

Notazione di base per percussioni

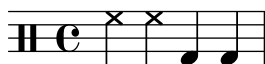
Le note percussive possono essere inserite in modalità `\drummode`, che è simile alla modalità standard di inserimento delle note. Il modo più semplice per inserire note percussive è il comando `\drums`, che crea il contesto e la modalità di inserimento corretti per le percussioni:

```
\drums {
  hihat4 hh bassdrum bd
}
```



Si tratta di una scorciatoia per:

```
\new DrumStaff \drummode {
  hihat4 hh bassdrum bd
}
```



Ogni elemento di uno strumento a percussione ha un nome completo e un nome abbreviato, entrambi utilizzabili nei file di input. L'elenco completo dei nomi delle note percussive si trova in Sezione A.17 [Note percussive], pagina 796.

Si noti che l'uso della normale notazione delle altezze (come `cis4`) in un contesto `DrumStaff` produrrà un messaggio di errore. Le chiavi delle percussioni sono aggiunte automaticamente a un contesto `DrumStaff` ma possono anche essere impostate esplicitamente. È possibile usare anche altre chiavi.

```
\drums {
  \clef percussion
  bd4 4 4 4
  \clef treble
  hh4 4 4 4
}
```



Ci sono alcuni problemi relativi al supporto del MIDI per gli strumenti percussivi; maggiori dettagli in Sezione 3.5 [Creazione dell'output MIDI], pagina 537.

Vedi anche

Guida alla notazione: Sezione 3.5 [Creazione dell'output MIDI], pagina 537, Sezione A.17 [Note percussive], pagina 796.

File installati: `ly/drumpitch-init.ly`.

Frammenti: Sezione "Percussion" in *Frammenti di codice*.

Rulli di tamburo

I rulli di tamburo si indicano con tre slash sul gambo. Le note di un quarto o più lunghe vengono mostrate con tre slash, quelle da un ottavo con due slash (la travatura rappresenta la terza), mentre le note più brevi di un'ottavo hanno uno slash sul gambo per integrare le travature. Ciò si ottiene con la notazione per tremolo, come è descritto in [Ripetizioni con tremolo], pagina 176.

```
\drums {
  \time 2/4
  sn16 8 16 8 8:32 ~
  8 8 4:32 ~
  4 8 16 16
  4 r4
}
```



La scelta di quale bacchetta usare può essere indicata tramite il testo "R" (right, destra) o "L" (left, sinistra) da porre sopra o sotto le note, come spiegato in Sezione 5.4.2 [Direzione e posizionamento], pagina 643. Conviene modificare la proprietà `staff-padding` per ottenere un allineamento soddisfacente.

```
\drums {
  \repeat unfold 2 {
    sn16~"L" 16~"R" 16~"L" 16~"L" 16~"R" 16~"L" 16~"R" 16~"R"
    \stemUp
    sn16_"L" 16_"R" 16_"L" 16_"L" 16_"R" 16_"L" 16_"R" 16_"R"
  }
}
```



Vedi anche

Guida alla notazione: [Ripetizioni con tremolo], pagina 176.

Frammenti: Sezione "Percussion" in *Frammenti di codice*.

Percussioni a suono determinato

Alcuni strumenti percussivi a suono determinato (per esempio xilofono, vibrafono e timpani) si scrivono usando i righi normali, come è spiegato in altre sezioni del manuale.

Vedi anche

Guida alla notazione: Sezione 3.5 [Creazione dell'output MIDI], pagina 537.

Frammenti: Sezione “Percussion” in *Frammenti di codice*.

Righi delle percussioni

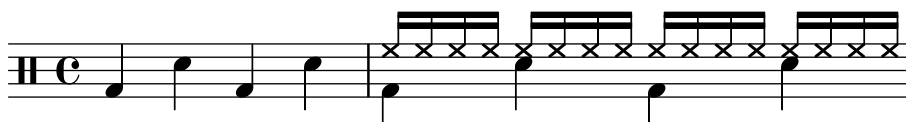
Una parte percussiva per più di uno strumento di norma usa un pentagramma multilinea dove ogni posizione sul rigo si riferisce a una parte delle percussioni. Per scrivere questo tipo di musica, le note devono essere interpretate nei contesti `DrumStaff` e `DrumVoice`.

```
up = \drummode {
  crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat
}
down = \drummode {
  bassdrum4 snare8 bd r bd sn4
}
\new DrumStaff <<
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



L'esempio precedente mostra la notazione polifonica in forma estesa. Ma si può usare anche quella in forma breve, descritta in Sezione “Sento le Voci” in *Manuale di Apprendimento*. Per esempio,

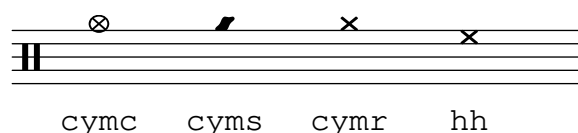
```
\new DrumStaff <<
  \drummode {
    bd4 sn4 bd4 sn4
    << {
      \repeat unfold 16 hh16
    } \ {
      bd4 sn4 bd4 sn4
    } >>
  }
>>
```

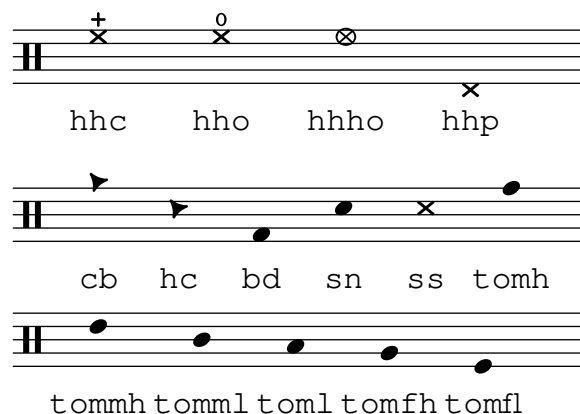


Esistono anche altre opzioni di formattazione, che si possono impostare tramite la proprietà `drumStyleTable` del contesto `DrumVoice`. Sono state definite le seguenti variabili:

`drums-style`

Questo è lo stile predefinito. Crea uno spartito per una tipica batteria su un rigo di cinque linee:

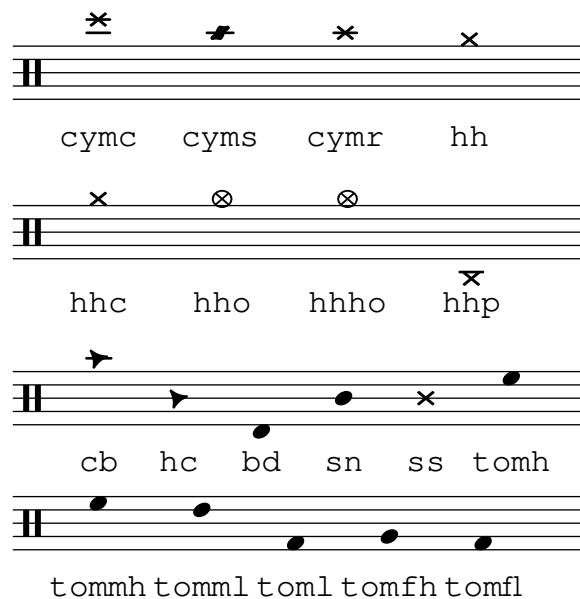




Lo schema percussivo supporta sei diversi tom. Quando si usa un numero inferiore di tom, selezionare i tom che producono il risultato desiderato. Per esempio, per avere i tom sulle tre linee centrali si usa tommh, tomml e tomfh.

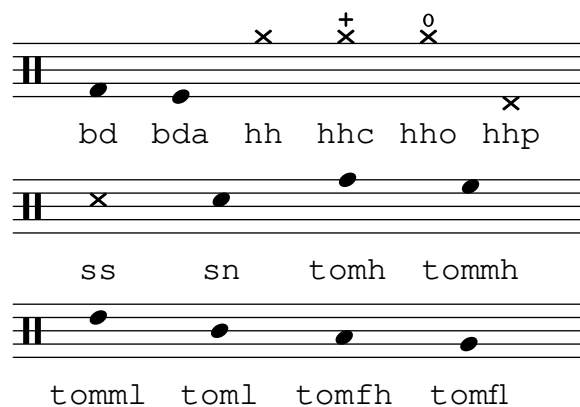
agostini-drums-style

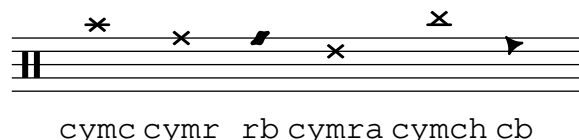
Inventata dal percussionista francese Dante Agostini nel 1965, questa notazione è comunemente utilizzata in Francia ma anche altrove.



weinberg-drums-style

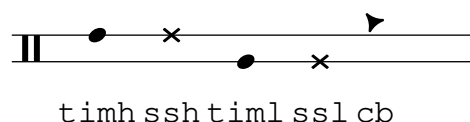
Basato sul lavoro di Norman Weinberg, pubblicato nel suo *Guidelines for Drumset Notation*.





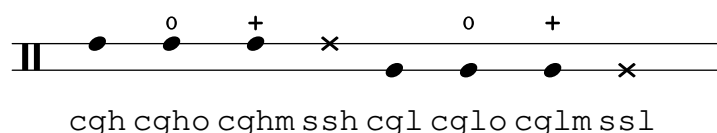
timbales-style

Questo stile rappresenta i timbales su un rigo di due linee:



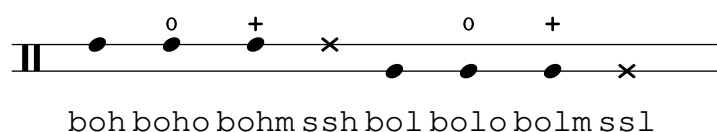
congas-style

Questo stile rappresenta le conga su un rigo di due linee:



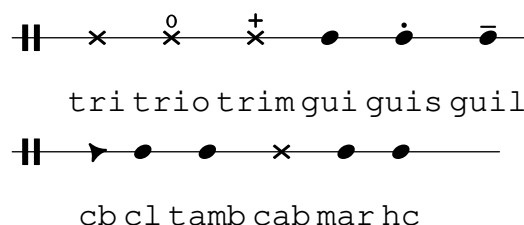
bongos-style

Questo stile rappresenta i bongo su un rigo di due linee:



percussion-style

Per rappresentare qualsiasi tipo di percussione semplice su righe di una linea:



Si possono anche definire stili percussivi personalizzati, come è spiegato in [Righi delle percussioni personalizzati], pagina 416.

Vedi anche

Manuale di apprendimento: Sezione “Sento le Voci” in *Manuale di Apprendimento*.

Guida alla notazione: [Righi delle percussioni personalizzati], pagina 416.

File installati: `ly/drumpitch-init.ly`.

Frammenti: Sezione “Percussion” in *Frammenti di codice*.

Righi delle percussioni personalizzati

È possibile definire stili percussivi personalizzati, su cui poi impostare la proprietà `drumStyleTable`. Le notazioni esistenti possono essere ridefinite come una lista associativa in cui ogni voce deve essere costituita da quattro elementi: un nome, lo stile della testa di nota (o default), un segno di articolazione se necessario (o `#f` se non lo è), e la posizione della testa di nota sul rigo. Questa lista deve poi essere convertita in una tabella hash Scheme, usando la funzione `alist->hash-table`.

```
#(define mydrums '(
```

```

(bassdrum      default  #f          -1)
(snare         default  #f          0)
(hihat         cross    #f          1)
(halfopenhihat cross    halfopen    1)
(pedalhihat     xcircle stopped      2)
(lowtom        diamond  #f          3)))
up = \drummode { hh8 hh hhho hhho hhp4 hhp }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>

```



Si possono aggiungere anche nuovi nomi a queste notazioni personalizzate attraverso la variabile `drumPitchNames`, che può essere ridefinita come una lista associativa (o aumentata appendendo una nuova lista al suo valore esistente, come illustrato sotto), ma anche attraverso le sue voci individuali. Ciò permette anche di definire alias: scorciatoie di input alternativo per alcune notazioni.

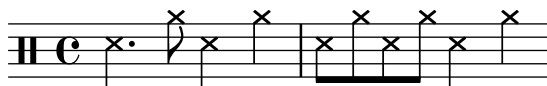
```

drumPitchNames =
  #(append
    '((leftsnap . sidestick)
      (rightsnap . ridecymbal))
    drumPitchNames)

drumPitchNames.ls = #'sidestick
drumPitchNames.rs = #'ridecymbal

\drums {
  leftsnap4. rightsnap8 leftsnap4 rightsnap
  ls8 rs ls rs ls4 rs
}

```



In modo analogo, la proprietà `drumPitchTable` associa una certa altezza (ovvero un diverso suono di strumento, così come è fornito dai soundfont MIDI disponibili) a ciascuna notazione. Questa proprietà deve essere definita come una tabella di hash, che di nuovo viene convertita da una lista associativa (salvata per impostazione predefinita come variabile `midiDrumPitches`). La ridefinizione di queste associazioni si ottiene, come spiegato sopra, o definendo un'intera lista associativa oppure attraverso voci individuali. L'esempio seguente mostra come creare un insieme di notazione completo con la propria sintassi di input, notazioni personalizzate e output MIDI corrispondente.

```

drumPitchNames.dbass = #'dbass
drumPitchNames.dba   = #'dbass % 'db è già in uso
drumPitchNames.dbassmute = #'dbassmute

```

```

drumPitchNames.dbm      = #'dbassmute
drumPitchNames.do       = #'dopen
drumPitchNames.dopenmute = #'dopenmute
drumPitchNames.dom      = #'dopenmute
drumPitchNames.dslap    = #'dslap
drumPitchNames.ds       = #'dslap
drumPitchNames.dslapmute = #'dslapmute
drumPitchNames.dsm      = #'dslapmute

#(define djembe-style
  '((dbass      default #f      -2)
    (dbassmute  default stopped -2)
    (dopen      default #f      0)
    (dopenmute  default stopped 0)
    (dslap      default #f      2)
    (dslapmute  default stopped 2)))

midiDrumPitches.dbass    = g
midiDrumPitches.dbassmute = fis
midiDrumPitches.dopen    = a
midiDrumPitches.dopenmute = gis
midiDrumPitches.dslap    = b
midiDrumPitches.dslapmute = ais

test = \drummode { dba4 do ds dbm dom dsm }

\score {
  \new DrumStaff \with {
    \override StaffSymbol.line-count = #3
    instrumentName = "Djembe "
    drumStyleTable = #(alist->hash-table djembe-style)
    drumPitchTable = #(alist->hash-table midiDrumPitches)
  } {
    \time 3/4
    \test
  }
  \layout {}
  \midi {}
}

```



Vedi anche

File installati: ly/drumpitch-init.ly.

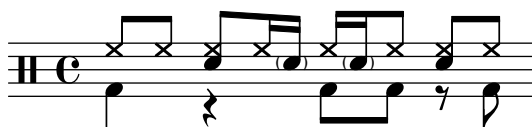
Frammenti: Sezione “Percussion” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “DrumStaff” in *Guida al Funzionamento Interno*,
Sezione “DrumVoice” in *Guida al Funzionamento Interno*.

Ghost note

Le ghost note (note fantasma), note anche come note morte, mute o non suonate, possono essere create col comando `\parenthesize`, spiegato in [Parentesi], pagina 243.

```
\new DrumStaff
<<
  \new DrumVoice = "1" { s1 }
  \new DrumVoice = "2" { s1 }
  \drummode {
    <<
      {
        hh8[ 8] <hh sn> hh16
        \parenthesize sn hh
        \parenthesize sn hh8 <hh sn> hh
      } \\
      {
        bd4 r4 bd8 8 r8 bd
      }
    >>
  }
>>
```



Vedi anche

Guida alla Notazione: [Parentesi], pagina 243.

Frammenti: Sezione “Percussion” in *Frammenti di codice*.

2.6 Strumenti aerofoni

Moderato assai

Questa sezione tratta gli elementi della notazione musicale specifici degli strumenti aerofoni (o strumenti a fiato).

2.6.1 Notazione comune per gli strumenti aerofoni

Questa sezione tratta la notazione comune alla maggior parte degli strumenti aerofoni.

Riferimenti per strumenti aerofoni

Molte delle questioni di notazione degli strumenti aerofoni (o strumenti a fiato) riguardano la respirazione e l'uso della lingua:

- La respirazione può essere indicata con pause o [Respiri], pagina 147.

- Un'esecuzione in legato è indicata con [Legature di portamento], pagina 142.
- Diversi tipi di uso della lingua, dal legato al non legato allo staccato, sono rappresentati solitamente da segni di articolazione, talvolta insieme a legature di portamento, vedi [Articolazioni e abbellimenti], pagina 129, e Sezione A.15 [Elenco delle articolazioni], pagina 793.
- Il frullato si indica solitamente ponendo un segno di tremolo e del testo vicino alla nota. Vedi [Ripetizioni con tremolo], pagina 176.

Altri aspetti della notazione musicale che si possono applicare agli strumenti aerofoni:

- Molti aerofoni sono strumenti traspositori, vedi [Trasporto strumentale], pagina 27.
- I glissati sono caratteristici del trombone, ma altri aerofoni sono in grado di eseguire dei glissati agendo sulle chiavi o sui pistoni. Vedi [Glissando], pagina 149.
- I glissati ottenuti con una serie di armonici, possibili su tutti gli ottoni ma comuni nel corno francese, vengono scritti solitamente come [Abbellimenti], pagina 122.
- Le intonazioni dell'altezza al termine di una nota sono trattate in [Portamenti indeterminati discendenti (cadute) e ascendenti], pagina 148.
- Le chiavi o i pistoni sbattuti si mostrano spesso con lo stile cross (barrato) delle [Teste di nota speciali], pagina 41.
- I legni possono produrre armonici soffiando forte sulle note basse. Gli armonici sono illustrati dall'articolazione *flageolet*. Vedi Sezione A.15 [Elenco delle articolazioni], pagina 793.
- L'uso di sordine per gli ottoni viene indicato solitamente con del testo, ma se ci sono molti cambi rapidi è meglio usare le articolazioni *stopped* e *open*. Vedi [Articolazioni e abbellimenti], pagina 129, e Sezione A.15 [Elenco delle articolazioni], pagina 793.
- I corni "smorzati" si indicano con l'articolazione *stopped*. Vedi [Articolazioni e abbellimenti], pagina 129.

Vedi anche

Guida alla notazione: [Respiri], pagina 147, [Legature di portamento], pagina 142, [Articolazioni e abbellimenti], pagina 129, Sezione A.15 [Elenco delle articolazioni], pagina 793, [Ripetizioni con tremolo], pagina 176, [Trasporto strumentale], pagina 27, [Glissando], pagina 149, [Abbellimenti], pagina 122, [Portamenti indeterminati discendenti (cadute) e ascendenti], pagina 148, [Teste di nota speciali], pagina 41,

Frammenti: Sezione "Wind instruments" in *Frammenti di codice*.

Diteggiature

Tutti gli aerofoni eccetto il trombone richiedono l'uso di varie dita per produrre ciascuna altezza. Alcuni esempi di diteggiatura sono illustrati nei prossimi frammenti.

Si possono creare diagrammi per i legni, descritti in Sezione 2.6.3.1 [Diagrammi per legni], pagina 424.

Frammenti di codice selezionati

Simboli di diteggiatura per strumenti aerofoni

Si possono ottenere simboli speciali combinando glifi esistenti; ciò è utile per gli strumenti aerofoni.

```
lineup =
  \tweak outside-staff-padding #0
  \tweak staff-padding #0
  \tweak padding #0.2
  \tweak parent-alignment-X #CENTER
```

```

\tweak self-alignment-X #CENTER
\etc

\relative c' {
  g\open
  g\lineup ^\markup \combine
    \musicglyph "scripts.open"
    \musicglyph "scripts.tenuto"
  g\lineup ^\markup \combine
    \musicglyph "scripts.open"
    \musicglyph "scripts.stopped"
  g\stopped
}

```



Grafico della diteggiatura per flauto dolce

L'esempio seguente illustra come realizzare grafici delle diteggiature per strumenti aerofoni.

% range chart for paetzold contrabass recorder

```

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

\new Staff \with {
  \remove "Time_signature_engraver"
  \omit Stem
  \omit Flag
  \consists "Horizontal_bracket_engraver"
} {
  \clef bass
  \set Score.timing = ##f

  f,1*1/4 \glissando

  \clef violin
  gis'1*1/4

  a'4^\markup "1)"

  \centermarkup
  \once \override TextScript.padding = 2
  bes'1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
    \fontsize #-5
    \slashed-digit #0 \finger 1 \finger 2
    \finger 3 \finger 4 \finger 5 \finger 6 \finger 7 }
}

```

```

b'1*1/4

c''4^\markup "1)"

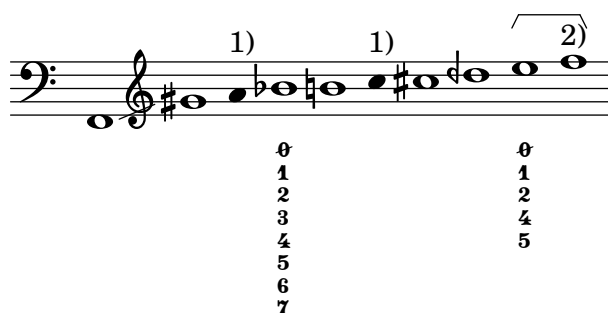
cis''1*1/4

deh''1*1/4

\centermarkup
\once \override TextScript.padding = 2
\once \override Staff.HorizontalBracket.direction = #UP
e''1*1/4_\markup \override #'(baseline-skip . 1.7) \column {
  \fontsize #-5
  \slashed-digit #0 \finger 1 \finger 2
  \finger 4 \finger 5 } \startGroup

f''1*1/4^\markup "2)" \stopGroup
}

```



Vedi anche

Guida alla notazione: Sezione 2.6.3.1 [Diagrammi per legni], pagina 424.

Frammenti: Sezione “Wind instruments” in *Frammenti di codice*.

2.6.2 Cornamusa

Questa sezione tratta la notazione comune per la cornamusa.

Definizioni per cornamusa

LilyPond contiene delle definizioni speciali per la musica per cornamusa scozzese; per usarle, aggiungere

```
\include "bagpipe.ly"
```

in cima al file di input. È così possibile aggiungere con comandi brevi gli speciali abbellimenti tipici della musica per cornamusa. Per esempio, si può scrivere `\taor` invece di

```
\grace { \small G32[ d G e] }
```

`bagpipe.ly` contiene anche le definizioni delle altezze delle note della cornamusa nelle ottave appropriate, in modo da non doversi preoccupare di `\relative` o `\transpose`.

```

\include "bagpipe.ly"
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }

```



In teoria, la musica per cornamusa usa la tonalità di Re maggiore (anche se ciò non è del tutto vero). Tuttavia, essendo l'unica tonalità che si può usare, normalmente non viene scritta esplicitamente. Dunque per nascondere la tonalità si può usare il comando `\hideKeySignature` prima che inizi la musica. Se per qualche ragione si desidera mostrare la tonalità, usare invece il comando `\showKeySignature`.

Alcune musiche moderne usano una diteggiatura barrata su Do e Fa per rendere bemolli queste note. Ciò si può indicare con `c-flat` o `f-flat`. Analogamente, il Sol alto del “piobaireachd” può essere scritto come `g-flat` quando si tratta di musica leggera.

Vedi anche

Frammenti: Sezione “Wind instruments” in *Frammenti di codice*.

Esempio per cornamusa

Ecco come appare la famosa melodia “Amazing Grace” in notazione per cornamusa.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove Bar_number_engraver }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdbl f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
```

```
\bar " | . "
}
```

Amazing Grace

Hymn

Trad. arr.



Vedi anche

Frammenti: Sezione “Wind instruments” in *Frammenti di codice*.

2.6.3 Legni

Questa sezione tratta la notazione specifica per i legni.

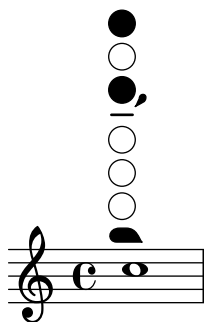
2.6.3.1 Diagrammi per legni

I diagrammi dei legni permettono di indicare la diteggiatura da usare per certe note e sono disponibili per i seguenti strumenti:

- ottavino
- flauto
- oboe
- clarinetto
- clarinetto basso
- sassofono
- fagotto
- controfagotto

I diagrammi dei legni vengono creati come testi di tipo markup:

```
c''1^\markup {
  \woodwind-diagram #'piccolo #'((lh . (gis))
                                (cc . (one three))
                                (rh . (ees)))
}
```



Le chiavi possono essere aperte, semichiusate, ad anello o chiuse completamente:

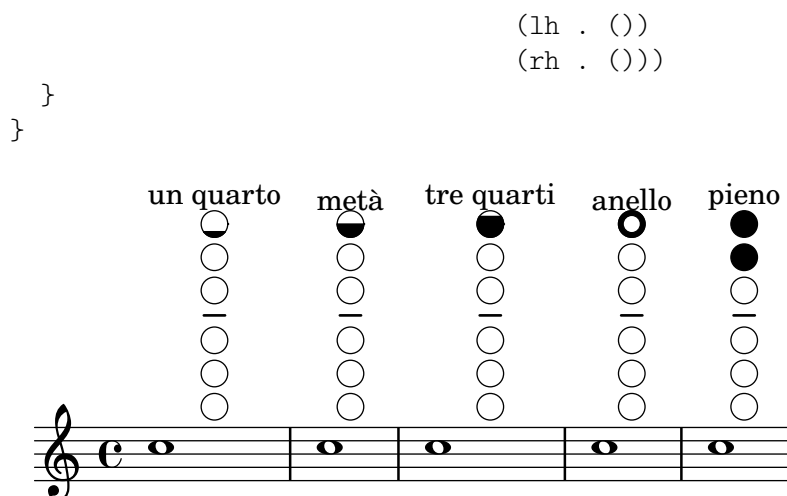
```
\textLengthOn
c''1^\markup {
  \center-column {
    "un quarto"
    \woodwind-diagram #'flute #'((cc . (one1q))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "metà"
    \woodwind-diagram #'flute #'((cc . (one1h))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "tre quarti"
    \woodwind-diagram #'flute #'((cc . (one3q))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "anello"
    \woodwind-diagram #'flute #'((cc . (oneR))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "pieno"
    \woodwind-diagram #'flute #'((cc . (oneF two))
```

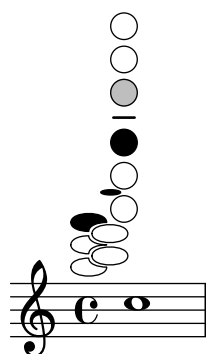


I trilli si indicano con delle chiavi sovrapposte:

```

c''1^\markup {
  \woodwind-diagram #'bass-clarinet
    #'((cc . (threeT four))
      (lh . ())
      (rh . (b fis)))
}

```



Possono essere visualizzati vari tipi di trilli:

```

\textLength0n
c''1^\markup {
  \center-column {
    "one quarter to ring"
    \woodwind-diagram #'flute #'((cc . (one1qTR))
      (lh . ())
      (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "ring to shut"
    \woodwind-diagram #'flute #'((cc . (oneTR))
      (lh . ())
      (rh . ()))
  }
}

```

```

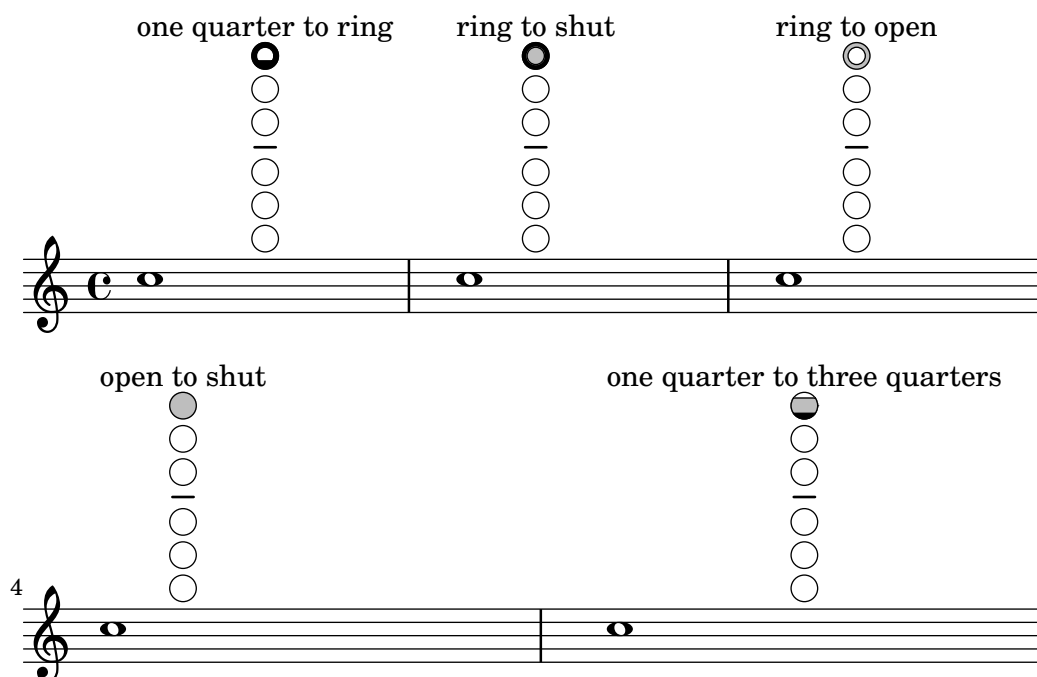
}

c''1^\markup {
  \center-column {
    "ring to open"
    \woodwind-diagram #'flute #'((cc . (oneRT))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "open to shut"
    \woodwind-diagram #'flute #'((cc . (oneT))
                                (lh . ()))
                                (rh . ()))
  }
}

c''1^\markup {
  \center-column {
    "one quarter to three quarters"
    \woodwind-diagram #'flute #'((cc . (one1qT3q))
                                (lh . ()))
                                (rh . ()))
  }
}

```



È possibile visualizzare nella console un elenco di tutte le chiavi e impostazioni possibili usando `$(print-keys-verbose 'flute)` oppure visualizzarlo nel file di log usando `$(print-keys-verbose 'flute (current-error-port))`, ma non appariranno nell'output.

È possibile creare nuovi diagrammi, ma è richiesta una conoscenza di Scheme e dunque tale possibilità non è a disposizione di tutti gli utenti. I modelli dei diagrammi si trovano nei file `scm/define-woodwind-diagrams.scm` e `scm/display-woodwind-diagrams.scm`.

Comandi predefiniti

Frammenti di codice selezionati

Elenco dei diagrammi per i legni

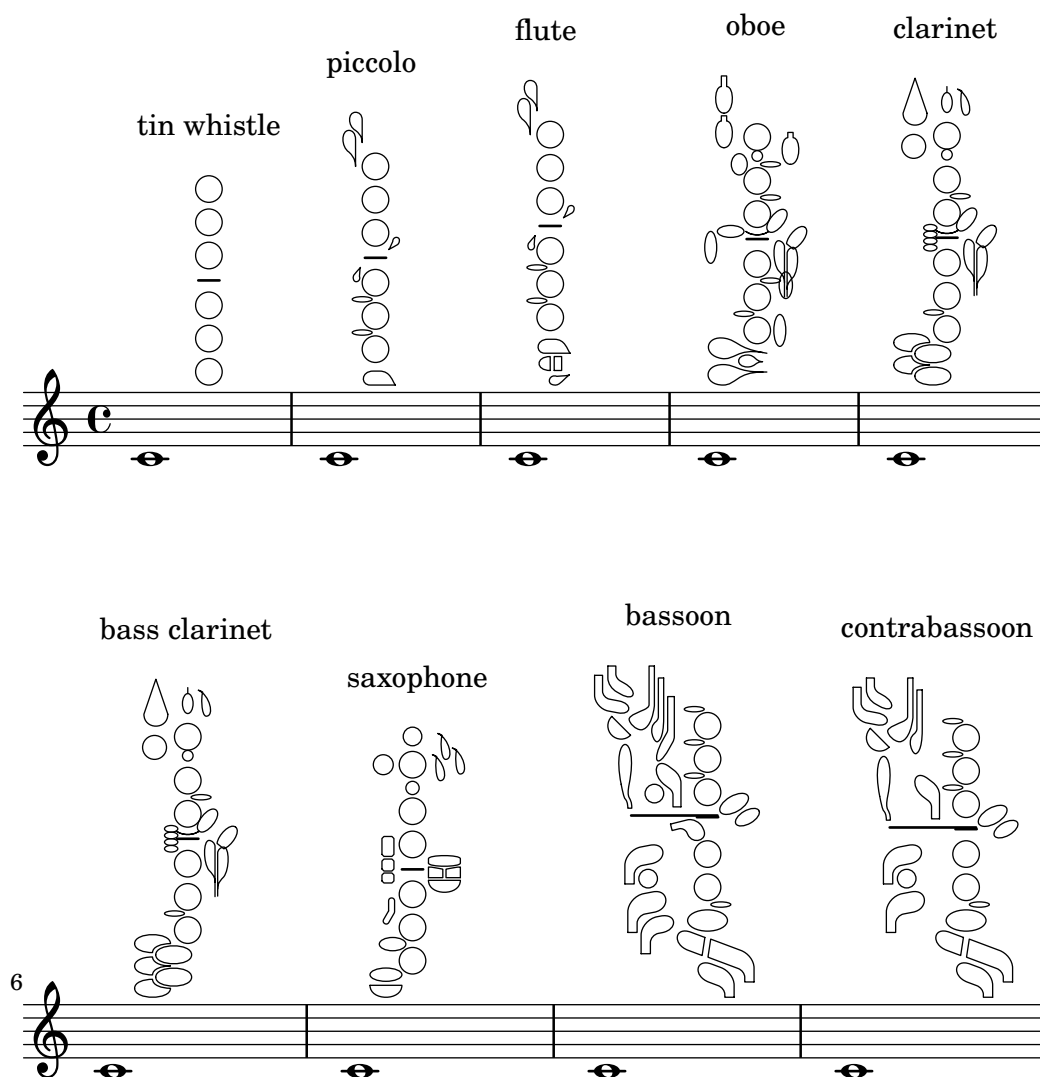
L'esempio seguente mostra tutti i diagrammi per i legni attualmente definiti in LilyPond.

```
\relative c' {
  \textLengthOn
  c1^\markup \center-column { "tin whistle"
    " "
    \woodwind-diagram #'tin-whistle #'() }
  c1^\markup \center-column { "piccolo"
    " "
    \woodwind-diagram #'piccolo #'() }
  c1^\markup \center-column { "flute"
    " "
    \woodwind-diagram #'flute #'() }
  c1^\markup \center-column { "oboe"
    " "
    \woodwind-diagram #'oboe #'() }
  c1^\markup \center-column { "clarinet"
    " "
    \woodwind-diagram #'clarinet #'() }

  \break

  c1^\markup \center-column { "bass clarinet"
    " "
    \woodwind-diagram #'bass-clarinet #'() }
  c1^\markup \center-column { "saxophone"
    " "
    \woodwind-diagram #'saxophone #'() }
  c1^\markup \center-column { "bassoon"
    " "
    \woodwind-diagram #'bassoon #'() }
  c1^\markup \center-column { "contrabassoon"
    " "
    \woodwind-diagram #'contrabassoon #'() }
}

\paper {
  system-system-spacing.padding = 5
}
```

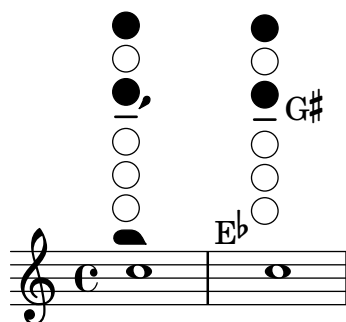


Diagrammi grafici e testuali per i legni

In molti casi, le chiavi diverse da quelle della colonna centrale possono essere visualizzate per nome oltre che in forma grafica.

```
\relative c' {
  \textLengthOn
  c1~\markup
    \woodwind-diagram #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))

  c~\markup
    \override #'(graphical . #f)
    \woodwind-diagram #'piccolo
      #'((cc . (one three))
        (lh . (gis))
        (rh . (ees)))
}
```

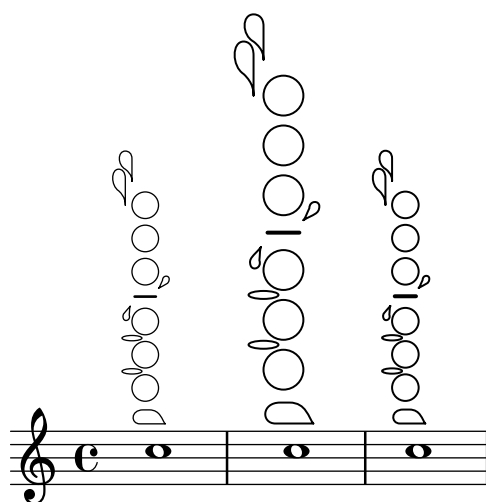


Cambiare la dimensione dei diagrammi per legni

La dimensione e lo spessore dei diagrammi per legni possono essere modificati.

```
\relative c'' {
  \textLength0n
  c1^\markup
    \woodwind-diagram #'piccolo #'()

  c^\markup \override #'(size . 1.5)
    \woodwind-diagram #'piccolo #'()
  c^\markup \override #'(thickness . 0.15)
    \woodwind-diagram #'piccolo #'()
}
```



Elenco delle chiavi dei diagrammi per legni

Il seguente frammento produce un elenco di tutte le possibili chiavi e delle loro impostazioni per i diagrammi per legni, come sono definite in `scm/define-woodwind-diagrams.scm`. L'elenco sarà visualizzato nel file di log, ma non nello spartito. Se si desidera che l'output appaia nella console, omettere dai comandi `(current-error-port)`.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
```

```

#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c'1}

```



Vedi anche

File installati: scm/define-woodwind-diagrams.scm,
scm/display-woodwind-diagrams.scm.

Frammenti: Sezione “Wind instruments” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “TextScript” in *Guida al Funzionamento Interno*,
Sezione “instrument-specific-markup-interface” in *Guida al Funzionamento Interno*.

2.7 Notazione per accordi

F C F F C F
 1. Fair is the sun - shine, Fair - er the moon - light
 2. Fair are the mead - ows, Fair - er the wood - land,
 F B \flat F C⁷ F C
 And all the stars_ in heav'n a - bove;
 Robed in the flow - ers of bloom - ing spring;

Gli accordi possono essere inseriti come normali note oppure nella modalità per accordi, e visualizzati usando varie convenzioni della tradizione europea. Possono essere visualizzati anche i nomi degli accordi e la notazione del basso continuo.

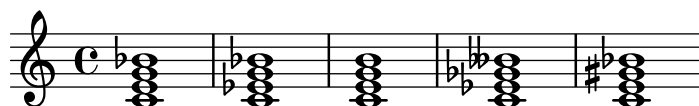
2.7.1 Modalità accordo

La modalità accordo serve a inserire accordi usando un indicatore della struttura dell'accordo invece delle altezze dell'accordo.



Si possono creare accordi di settima:

```
\chordmode { c1:7 c:m7 c:maj7 c:dim7 c:aug7 }
```



La tabella seguente mostra le azioni dei modificatori della qualità su triadi e accordi di settima. Il settimo grado predefinito aggiunto agli accordi è una settima minore, che rende la settima dominante l'accordo di settima fondamentale. Tutte le alterazioni sono relative alla settima dominante. Una tabella più completa dell'uso dei modificatori si trova in Sezione A.2 [Modificatori degli accordi comuni], pagina 678.

Modificatore	Azione	Esempio
Nessuno	L'azione predefinita; produce una triade maggiore.	
m, m7	L'accordo minore. Questo modificatore abbassa il 3° grado.	
dim, dim7	L'accordo diminuito. Questo modificatore abbassa il 3°, il 5° e (se presente) il 7° grado.	
aug	L'accordo aumentato. Questo modificatore alza il 5° grado.	
maj, maj7	L'accordo di settima maggiore. Questo modificatore aggiunge un 7° grado alzato. Il 7 che segue maj è opzionale. Non usare questo modificatore per creare una triade maggiore.	

Vedi anche

Guida alla notazione: Sezione A.2 [Modificatori degli accordi comuni], pagina 678, [Accordi estesi e alterati], pagina 434.

Frammenti: Sezione "Chord notation" in *Frammenti di codice*.

Problemi noti e avvertimenti

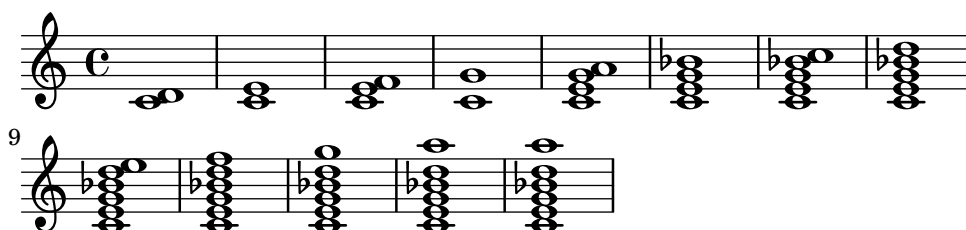
Si può usare un solo modificatore della qualità per accordo, solitamente sul grado più alto presente nell'accordo. Gli accordi con più di un modificatore verranno analizzati senza produrre

errori o avvisi, ma producono risultati imprevedibili. Gli accordi che non si possono ottenere con un singolo modificatore devono essere alterati per altezze individuali, come è spiegato in [Accordi estesi e alterati], pagina 434.

Accordi estesi e alterati

In modalità accordo è possibile creare strutture di accordi di arbitraria complessità. Il modificatore permette di estendere un accordo, aggiungere o rimuovere i gradi di un accordo, alzare o abbassare i gradi, aggiungere una nota di basso o creare un'inversione. Il primo numero che segue il : identifica l'estensione dell'accordo. L'accordo viene costruito aggiungendo in modo sequenziale delle terze alla nota fondamentale finché non si raggiunge il numero specificato. Il settimo grado aggiunto come parte di un accordo esteso sarà la settima minore, non la settima maggiore. Se l'estensione non è una terza (per esempio, 6), vengono aggiunte le terze fino alla terza più alta entro l'estensione, e poi viene aggiunto il grado dell'estensione. Il valore più grande possibile è 13. Qualsiasi valore superiore viene interpretato come 13.

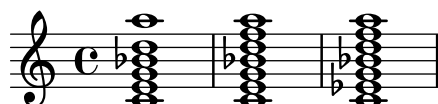
```
\chordmode {
  c1:2 c:3 c:4 c:5
  c1:6 c:7 c:8 c:9
  c1:10 c:11 c:12 c:13
  c1:14
}
```



C'è un'eccezione speciale: c:5 produce un "power chord" che consiste solo della nota fondamentale e della quinta.

Dato che un 11° grado non alterato non suona bene se unito a un 13° grado non alterato, l'11° grado viene rimosso da un accordo :13, a meno che non venga aggiunto esplicitamente.

```
\chordmode {
  c1:13 c:13.11 c:m13
}
```



Si possono aggiungere gradi individuali a un accordo. Le aggiunte seguono l'estensione e sono precedute da un punto (.). Il settimo grado aggiunto a un accordo è una settima minore invece di una settima maggiore.

```
\chordmode {
  c1:3.5.6 c:3.7.8 c:3.6.13
}
```



I gradi aggiunti possono essere alti quanto si vuole.

```
\chordmode {
```

```
c4:3.5.15 c:3.5.20 c:3.5.25 c:3.5.30
}
```



I gradi aggiunti dell'accordo possono essere alterati aggiungendo un suffisso - o + al numero. Per alterare un grado aggiunto automaticamente in quanto parte della struttura base dell'accordo, aggiungerlo come grado alterato.

```
\chordmode {
  c1:7+ c:5+.3- c:3-.5-.7-
}
```



Dopo i gradi da aggiungere, una serie di gradi da togliere viene introdotta dal modificatore con prefisso ^. Per togliere più di un grado, i gradi da rimuovere vengono separati dal . subito dopo il ^ iniziale.

```
\chordmode {
  c1^3 c:7^5 c:9^3 c:9^3.5 c:13.11^3.7
}
```



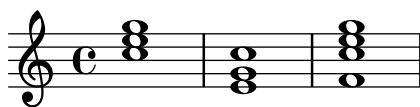
Per creare accordi sospesi si aggiunge alla stringa il modificatore sus, che toglie il terzo grado dall'accordo. Aggiungere 2 o 4 per aggiungere il 2° o il 4° grado dell'accordo. Se sus è seguito dal 2° o dal 4° grado, è equivalente a ^3, altrimenti a sus4, ovvero 5.4.

```
\chordmode {
  c1:sus c:sus2 c:sus4 c:5.4
}
```



Le note di basso aggiunte (ovvero l'inserimento di un'altezza diversa dalla fondamentale come nota più bassa dell'accordo) possono essere specificate aggiungendo /altezza all'accordo.

```
\chordmode {
  c'1 c'/e c'/f
}
```



Se l'altezza aggiunta fa già parte dell'accordo, questa può essere usata per mostrare l'inversione di accordo, nel qual caso l'altezza non viene aggiunta ma semplicemente spostata in fondo all'accordo. È comunque possibile trattarla come una nota aggiunta (facendola quindi apparire due volte) usando la sintassi */+altezza*.

```
\chordmode {
  c'1 c'/g c'/+e
}
```



Le inversioni degli accordi automatiche e le voicing sono spiegate in [Inversioni di accordo e voicing], pagina 436.

I modificatori degli accordi che possono essere usati per produrre i vari accordi standard sono elencati in Sezione A.2 [Modificatori degli accordi comuni], pagina 678.

Vedi anche

Guida alla notazione: [Inversioni di accordo e voicing], pagina 436, Sezione A.2 [Modificatori degli accordi comuni], pagina 678.

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

Problemi noti e avvertimenti

Ogni grado può essere presente in un accordo una volta sola. L'esempio seguente produce un accordo aumentato, perché 5+ viene interpretato per ultimo.

```
\chordmode { c1:3.5.5-.5+ }
```



Inversioni di accordo e voicing

Oltre ai modificatori degli accordi e alle note di basso aggiunte, si possono usare varie funzioni per far sì che gli accordi appaiano automaticamente in una certa inversione o *voicing* – per esempio, la voicing cosiddetta ‘drop 2’ (abbassamento) usata comunemente nella musica jazz.

```
\chordmode {
  \dropNote 2 {
    c2:maj7 d:m7
  }
  \invertChords 1 d1:maj7
}
```



Diversamente dalle note di basso aggiunte mostrate in [Accordi estesi e alterati], pagina 434, questo ha effetto solo sul modo in cui gli accordi appaiono sul rigo, e non sui nomi degli accordi scritti in lettere. Inoltre queste funzioni possono essere usate non solo in modalità accordo ma anche coi costrutti per accordo $\langle \dots \rangle$ spiegati in [Note in un accordo], pagina 178.

Vedi anche

Guida alla notazione: [Accordi estesi e alterati], pagina 434, [Note in un accordo], pagina 178.

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

2.7.2 Visualizzazione accordi

Gli accordi possono essere visualizzati per nome, oltre alla visualizzazione standard come note di un rigo.

Stampa dei nomi degli accordi

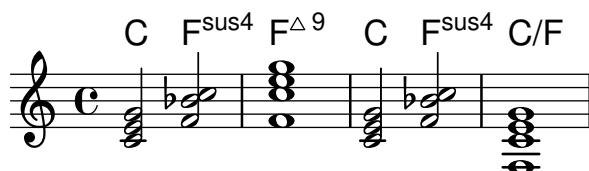
Per visualizzare i nomi degli accordi si usa il contesto ChordNames:

```
\new ChordNames {
  \chordmode {
    c2 f4. g8
  }
}
```

C F G

Gli accordi possono essere inseriti come note simultanee o tramite la modalità accordo. Il nome dell'accordo visualizzato sarà lo stesso, indipendentemente dalla modalità di inserimento, a meno che non ci siano inversioni o note di basso:

```
chordmusic = \relative {
  <c' e g>2 <f bes c>
  <f c' e g>1
  \chordmode {
    c2 f:sus4 c1:/f
  }
}
<<
\new ChordNames {
  \chordmusic
}
{
  \chordmusic
}
>>
```



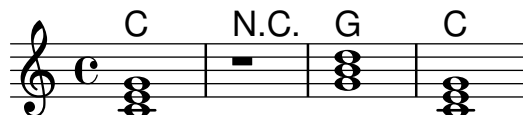
Le pause usate nel contesto ChordNames faranno apparire il testo del simbolo noChordSymbol (nessun accordo).

```
<<
\new ChordNames \chordmode {
  c1
  r1
  g1
  c1
}
```

```

\chordmode {
  c1
  r1
  g1
  c1
}
>>

```



`\chords { ... }` è una scorciatoia per `\new ChordNames \chordmode { ... }`.

```

\chords {
  c2 f4.:m g8:maj7
}

```

C Fm G^Δ

```

\new ChordNames {
  \chordmode {
    c2 f4.:m g8:maj7
  }
}

```

C Fm G^Δ

Frammenti di codice selezionati

Mostrare gli accordi nei cambi

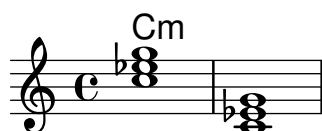
Per impostazione predefinita, ogni accordo inserito viene visualizzato; tale comportamento può essere modificato in modo che i nomi degli accordi siano mostrati solo all'inizio delle linee e quando l'accordo cambia.

```

harmonies = \chordmode {
  c'1:m c:m \break
  c'1:m c:m d
}

<<
\new ChordNames {
  \set chordChanges = ##t
  \harmonies
}
\new Staff {
  \harmonies
}
>>

```

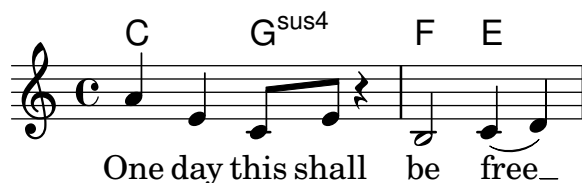




Canzoniere semplice

Mettendo insieme nomi degli accordi, melodia e testo si ottiene un canzoniere (in inglese “lead sheet”):

```
<<
\chords { c2 g:sus4 f e }
\new Staff \relative c'' {
  a4 e c8 e r4
  b2 c4( d)
}
\addlyrics { One day this shall be free __ }
>>
```



Vedi anche

Glossario musicale: Sezione “accordo” in *Glossario Musicale*.

Guida alla notazione: [Scrivere la musica in parallelo], pagina 199.

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “ChordNames” in *Guida al Funzionamento Interno*, Sezione “ChordName” in *Guida al Funzionamento Interno*, Sezione “Chord_name_engraver” in *Guida al Funzionamento Interno*, Sezione “Volta_engraver” in *Guida al Funzionamento Interno*, Sezione “Bar_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Gli accordi che contengono inversioni o note di basso alterate non avranno un nome corretto se inseriti usando la musica simultanea.

Personalizzazione dei nomi degli accordi

Non esiste un unico sistema per il nome degli accordi. Tradizioni musicali diverse usano nomi diversi per lo stesso gruppo di accordi. Esistono anche simboli diversi per un certo nome di accordo. I nomi e i simboli degli accordi sono personalizzabili.

Il formato fondamentale dei nomi degli accordi è un sistema per la musica Jazz, proposto da Klaus Ignatzek (vedi Sezione “Bibliografia” in *Saggio*). (Si possono implementare altri sistemi per i nomi degli accordi attraverso funzioni Scheme, come illustrato nel frammento “Chord names alternative” in Sezione “Chord notation” in *Frammenti di codice*.) Un elenco dei comuni accordi jazz si trova nel Sezione A.1 [Grafico dei nomi degli accordi], pagina 678.

Il sistema predefinito per i nomi degli accordi può essere modificato facilmente in vari modi. Per iniziare, dei comandi predefiniti consentono di usare lingue diverse per l'altezza fondamentale. I comandi predefiniti sono `\germanChords`, `\semiGermanChords`, `\italianChords` e `\frenchChords`:

default	E/D	Cm	B/B	B [#] /B [#]	B ^b /B ^b	B ^{bb} /B ^{bb}
German	E/d	Cm	H/h	H [#] /his	B/b	H ^{bb} /heses
Norwegian	E/D	Cm	H/H	H [#] /Hiss	B/B	B ^b /Bess
Italian	Mi/Re	Do m	Si/Si	Si [#] /Si [#]	Si ^b /Si ^b	Si ^{bb} /Si ^{bb}
French	Mi/Ré	Do m	Si/Si	Si [#] /Si [#]	Si ^b /Si ^b	Si ^{bb} /Si ^{bb}
English	E/D	Cm	B/B	B [#] /B [#]	B ^b /B ^b	B ^{bb} /B ^{bb}



I canzonieri tedeschi talvolta indicano gli accordi minori con lettere minuscole, senza alcun suffisso *m*. Ciò si può ottenere impostando la proprietà `chordNameLowercaseMinor`:

```
\chords {
  \set chordNameLowercaseMinor = ##t
  c2 d:m e:m f
}
```

C d e F

Il nome dell'accordo da visualizzare può essere aggiustato anche tramite le seguenti proprietà.

`chordRootNamer`

Il nome dell'accordo appare solitamente come una lettera per la nota fondamentale seguita da un'alterazione opzionale. La trasformazione dell'altezza in lettera è eseguita da questa funzione. Si possono creare nomi speciali di note (per esempio, il tedesco "H" per l'accordo di Si) salvando una nuova funzione in questa proprietà.

`majorSevenSymbol`

Questa proprietà contiene l'oggetto markup usato per seguire l'output di `chordRootNamer` e identificare un accordo di settima maggiore. Le opzioni predefinite sono `whiteTriangleMarkup` e `blackTriangleMarkup`.

`additionalPitchPrefix`

Se il nome dell'accordo contiene ulteriori altezze, si può aggiungere un prefisso testuale. Il comportamento predefinito è senza prefisso, per evitare troppo ingombro, ma per numeri piccoli di altezze aggiuntive il risultato può essere visualmente efficace.

```
\new ChordNames {
  <c e g d'> % add9
  \set additionalPitchPrefix = "add"
  <c e g d'> % add9
}
```

C^{add9} C^{add9}

`chordNoteNamer`

Se il nome dell'accordo contiene altre altezze oltre all'altezza fondamentale (per esempio, una nota di basso), questa funzione viene usata per mostrare l'altezza ulteriore. Per impostazione predefinita l'altezza viene stampata con la funzione

`chordRootNamer`. La proprietà `chordNoteNamer` può essere impostata su una funzione specializzata per cambiare tale comportamento. Per esempio, la nota di basso può essere stampata in minuscolo.

`chordNameSeparator`

Parti diverse del nome di un accordo sono separate da un piccolo spazio orizzontale. Impostando `chordNameSeparator`, si può usare qualsiasi testo come separatore. Ciò non ha effetti sul separatore tra un accordo e la sua nota di basso; per personalizzarlo si usa una proprietà specifica, `slashChordSeparator`.

```
\chords {
  c4:7.9- c:7.9-/g
  \set chordNameSeparator = \markup { "/" }
  \break
  c4:7.9- c:7.9-/g
}
```

$C^7 \flat^9 \ C^7 \flat^9 / G$

$C^{7/\flat^9} \ C^{7/\flat^9} / G$

`slashChordSeparator`

Gli accordi possono essere suonati utilizzando come nota di basso una nota diversa da quella fondamentale normalmente usata. In questo caso si parla di «inversioni» o «slash chord», perché il modo predefinito di rappresentarli è con una barra obliqua (forward slash) tra l'accordo principale e la nota di basso. Dunque il valore predefinito di `slashChordSeparator` è una barra obliqua, ma è possibile cambiarlo con qualsiasi testo.

```
\chords {
  c4:7.9- c:7.9-/g
  \set slashChordSeparator = \markup { " over " }
  \break
  c4:7.9- c:7.9-/g
}
```

$C^7 \flat^9 \ C^7 \flat^9 / G$

$C^7 \flat^9 \ C^7 \flat^9 \text{ over } G$

`chordNameExceptions`

Questa proprietà è una lista di coppie. Il primo elemento di ciascuna coppia è un insieme di altezze usate per identificare i gradi presenti nell'accordo. Il secondo elemento è un testo markup che seguirà l'output di `chordRootNamer` per creare il nome dell'accordo.

`minorChordModifier`

Gli accordi minori sono spesso indicati con un suffisso “m” a destra della fondamentale dell'accordo. Tuttavia alcuni preferiscono altri suffissi, come il segno meno.

```
\chords {
  c4:min f:min7
  \set minorChordModifier = \markup { "-" }
  \break
  c4:min f:min7
}
```

Cm Fm⁷C- F-⁷`chordPrefixSpacer`

Il modificatore per gli accordi minori determinato da `minorChordModifier` appare solitamente subito dopo la fondamentale dell'accordo. È possibile porre uno spazio tra la fondamentale e il modificatore impostando `chordPrefixSpacer`. Lo spazio non viene usato quando la fondamentale è alterata.

Comandi predefiniti

`\whiteTriangleMarkup`, `\blackTriangleMarkup`, `\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

Frammenti di codice selezionati

Eccezioni dei nomi degli accordi

La proprietà `chordNameExceptions` permette di salvare un elenco di notazioni speciali per accordi specifici.

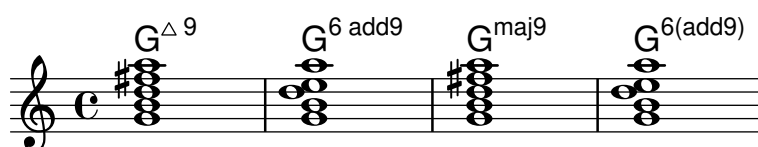
```
% Step 1: Define music with chords and markup for maj9 and 6(add9).
chExceptionMusic = {
  <c e g b d'>-\markup { \super "maj9" }
  <c e g a d'>-\markup { \super "6(add9)" }
}

% Step 2: Create extended exception list.
chExceptions =
#(append (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  % Step 3: Register extended exception list.
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

<<
  \new ChordNames \theMusic
  \new Voice \theMusic
>>

\layout {
  line-width = 10\cm
  ragged-right = ##f
}
```



Nome dell'accordo di settima maggiore

La formattazione dell'accordo di settima maggiore può essere regolata con la proprietà `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```

$C^\Delta C^j7$

Aggiungere stanghette al contesto ChordNames

Per mostrare le stanghette nel contesto `ChordNames`, aggiungere l'incisore `Bar_engraver`.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-1 . 3)
  \consists "Bar_engraver"
}
```

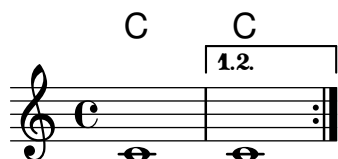
```
\chordmode {
  f1:maj7 f:7 bes:7
}
```

$F^\Delta \mid F^7 \mid B\flat^7 \mid$

Volta sotto gli accordi

Aggiungendo l'incisore `Volta_engraver` al rigo, è possibile inserire le volte sotto gli accordi.

```
\score {
  <<
    \chords { c1 c1 }
    \new Staff \with { \consists "Volta_engraver" }
    {
      \repeat volta 2 { c'1 \alternative { c' } }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Modifica del separatore dell'accordo

L'elemento che separa le diverse parti di un accordo può essere impostato su qualsiasi testo di tipo markup.

```
\chords {
```



```

c:7sus4
\set chordNameSeparator = \markup { \typewriter | }
c:7sus4
}

```

C⁷ sus4 **C**⁷ | sus4

Vedi anche

Guida alla notazione: Sezione A.1 [Grafico dei nomi degli accordi], pagina 678, Sezione A.2 [Modificatori degli accordi comuni], pagina 678.

Saggio sull’incisione musicale automatizzata: Sezione “Bibliografia” in *Saggio*.

File installati: scm/chords-ignatzek-names.scm, scm/chord-entry.scm, ly/chord-modifiers-init.ly.

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

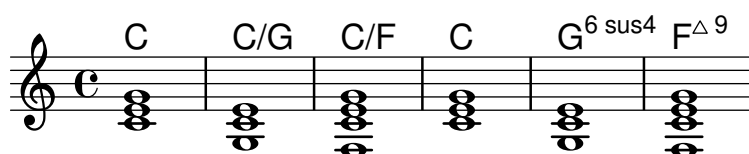
Problemi noti e avvertimenti

I nomi degli accordi sono determinati sia dalle altezze presenti nell’accordo sia dalle informazioni sulla struttura dell’accordo che potrebbero essere state inserite in `\chordmode`. Se si usa il metodo delle altezze simultanee per inserire gli accordi, in caso di inversioni o note di basso si otterranno nomi non voluti.

```

myChords = \relative c' {
  \chordmode { c1 c/g c/f }
  <c e g>1 <g c e> <f c' e g>
}
<<
  \new ChordNames { \myChords }
  \new Staff { \myChords }
>>

```



2.7.3 Basso continuo

Adagio

Violino I.

Violino II.

Violone,
e Cembalo.



La notazione del basso continuo è supportata.

Introduzione al basso continuo

LilyPond supporta il basso continuo, detto anche basso numerato o cifrato:

```
<<
\new Voice { \clef bass dis4 c d ais g fis}
\new FiguredBass {
  \figuremode {
    < 6 >4 < 7\+ >8 < 6+ [_!] >
    < 6 >4 <6 5 [3+] >
    < _ >4 < 6 5/>4
  }
}
>>
```



Il supporto per il basso continuo consiste in due parti: c'è una modalità di input, introdotta dal comando `\figuremode`, che accetta l'inserimento dei numeri di basso; e c'è un contesto chiamato `FiguredBass` che si occupa di mostrare gli oggetti `BassFigure`. Il basso continuo può essere mostrato anche in contesti `Staff`.

`\figures { ... }` è una scorciatoia per `\new FiguredBass \figuremode { ... }`.

Sebbene il supporto per basso continuo possa assomigliare apparentemente al supporto per accordi, è in realtà molto più semplice. La modalità `\figuremode` si limita a contenere i numeri e il contesto `FiguredBass` li stampa così come sono inseriti. Non c'è alcuna conversione in altezze.

Vedi anche

Glossario musicale: Sezione “basso continuo” in *Glossario Musicale*.

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

Inserimento del basso continuo

`\figuremode` permette di passare alla modalità di input per il basso continuo. Maggiori informazioni sulle diverse modalità di input si trovano in Sezione 5.4.1 [Modalità di inserimento], pagina 641.

In modalità basso continuo, un gruppo di numeri di basso è delimitato da `<` e `>`. La durata si inserisce dopo il `>`.

```
\new FiguredBass {
  \figuremode {
    <6 4>2
  }
}
```

6
4

Si possono usare le alterazioni (inclusi i bequadri) per modificare i gradi della scala. Si inseriscono aggiungendo `+` (per i diesis), `-` (per i bemolli) o `!` (per i bequadri) dopo il numero. Per le doppie alterazioni il modificatore va applicato due volte. Per modificare il terzo grado il numero di solito viene omesso e ciò si può fare usando `_` al posto del numero.

```
\figures {
  <7! 6+ 4-> <5++> <3--> < _+ > < 7 _!>
}
```

b7 **x5** **b3** **#** **7**
#6
b4

Si possono indicare gradi aumentati e diminuiti:

```
\figures {
  <6\+ 5/> <7/>
}
```

+6 **7**
5

Si può creare una barra inversa che taglia il numero (usata comunemente per i sesti gradi alzati):

```
\figures {
  <6> <6\\>
}
```

6 **6̣**

Si possono aggiungere parentesi quadre intorno a alterazioni, numeri di basso continuo e gruppi consecutivi di numeri di basso continuo:

```
\figures {
  <9[-] 8 [7-] 5 [4[!] 2+]>
```

```
}
[b]9
8
[b]7
5
[b]4
#2
```

Si può inserire come numero qualsiasi testo di tipo markup:

```
\figures {
  <\markup { \tiny \number 6 \super (1) } 5>
}

6(1)
5
```

Si possono usare linee di continuazione per indicare numeri ripetuti:

```
<<
{
  \clef bass
  e4 d c b,
  e4 d c b,
}
\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 3> <7 3> <7 3>
  \bassFigureExtendersOff
  <6 4>4 <6 3> <7 3> <7 3>
}
>>
```



In questo caso, le linee di estensione sostituiscono i numeri esistenti, a meno che le linee di continuazione non siano state terminate esplicitamente.

```
<<
\figures {
  \bassFigureExtendersOn
  <6 4>4 <6 4> <6\! 4\!> <6 4>
}
{
  \clef bass
  d4 d c c
}
>>
```



La seguente tabella riassume i modificatori del numero disponibili.

ModifierScopo**Esempio**

+, -, ! Alterazioni

 $\flat 7$ $\sharp 5$ $\flat 3$
 $\sharp 6$
 $\flat 4$

\+, / Gradi aumentati e diminuiti

 $\sharp 6$ $\flat 7$
 $\flat 5$

\\ Sesto grado alzato

6

\! Fine della linea di continuazione

**Comandi predefiniti**

\bassFigureExtendersOn, \bassFigureExtendersOff.

Frammenti di codice selezionati***Cambiare la posizione delle alterazioni del basso continuo***

Le alterazioni e i segni più possono apparire prima o dopo i numeri, a seconda delle proprietà `figuredBassAlterationDirection` e `figuredBassPlusDirection`.

#(set-global-staff-size 26)

```
\figures {
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassPlusDirection = #RIGHT
  <5\+> <5+ 4\+> <6 4- 2\+> r
  \set figuredBassAlterationDirection = #LEFT
  <5\+> <5+ 4\+> <6 4- 2\+> r
}
```

$\sharp 5$	$\sharp 5$	6	$\sharp 5$	$\sharp 5$	6	$\sharp 5$	$\sharp 5$	6
$\sharp 4$	$\sharp 4$	$\flat 4$	$\sharp 4$	$\sharp 4$	$\flat 4$	$\sharp 4$	$\sharp 4$	$\flat 4$
		+2			+2			+2

Vedi ancheFrammenti: Sezione “Chord notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “BassFigure” in *Guida al Funzionamento Interno*, Sezione “BassFigureAlignment” in *Guida al Funzionamento Interno*, Sezione “BassFigureLine” in *Guida al Funzionamento Interno*, Sezione “BassFigureBracket” in *Guida al Funzionamento Interno*, Sezione “BassFigureContinuation” in *Guida al Funzionamento Interno*, Sezione “FiguredBass” in *Guida al Funzionamento Interno*.

Visualizzazione del basso continuo

Il basso continuo può essere visualizzato usando il contesto `FiguredBass` o la maggior parte dei contesti del rigo.

Se visualizzato in un contesto `FiguredBass`, la posizione verticale dei numeri è indipendente dalle note del rigo.

```

<<
  \relative {
    c' '4 c'8 r8 c,4 c'
  }
  \new FiguredBass {
    \figuremode {
      <4>4 <10 6>8 s8
      <6 4>4 <6 4>
    }
  }
>>

```



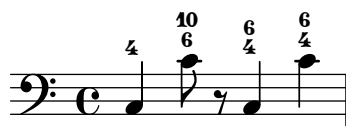
Nell'esempio precedente, il contesto `FiguredBass` deve essere istanziato esplicitamente per evitare la creazione di un secondo rigo (vuoto).

Il basso continuo può anche essere aggiunto direttamente ai contesti `Staff`. In questo caso la posizione verticale dei numeri è regolata automaticamente.

```

<<
  \new Staff = "mioRigo"
  \figuremode {
    <4>4 <10 6>8 s8
    <6 4>4 <6 4>
  }
  %% Inserire le note nello stesso rigo dei numeri di basso
  \context Staff = "mioRigo"
  {
    \clef bass
    c4 c'8 r8 c4 c'
  }
>>

```



Se aggiunto in un contesto `Staff`, il basso continuo può essere visualizzato sopra o sotto il rigo.

```

<<
  \new Staff = "mioRigo"
  \figuremode {
    <4>4 <10 6>8 s8
    \bassFigureStaffAlignmentDown
    <6 4>4 <6 4>
  }
  %% Inserire le note nello stesso rigo dei numeri di basso
  \context Staff = "mioRigo"
  {
    \clef bass

```

```

c4 c'8 r8 c4 c'
}
>>

```



Comandi predefiniti

`\bassFigureStaffAlignmentDown`, `\bassFigureStaffAlignmentUp`,
`\bassFigureStaffAlignmentNeutral`.

Vedi anche

Frammenti: Sezione “Chord notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “BassFigure” in *Guida al Funzionamento Interno*, Sezione “BassFigureAlignment” in *Guida al Funzionamento Interno*, Sezione “BassFigureLine” in *Guida al Funzionamento Interno*, Sezione “BassFigureBracket” in *Guida al Funzionamento Interno*, Sezione “BassFigureContinuation” in *Guida al Funzionamento Interno*, Sezione “FiguredBass” in *Guida al Funzionamento Interno*.

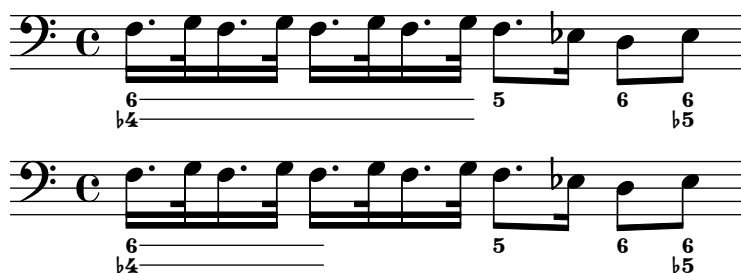
Problemi noti e avvertimenti

Per assicurarsi che le linee di continuazione funzionino correttamente, è più sicuro usare lo stesso ritmo nella linea dei numeri e nella linea dei bassi.

```

<<
{
  \clef bass
  \repeat unfold 4 { f16. g32 } f8. es16 d8 es
}
\figures {
  \bassFigureExtendersOn
  % Qui gli estensori sono corretti, perché hanno lo stesso ritmo del basso
  \repeat unfold 4 { <6 4->16. <6 4->32 }
  <5>8. r16 <6>8 <6\! 5->
}
>>
<<
{
  \clef bass
  \repeat unfold 4 { f16. g32 } f8. es16 d8 es
}
\figures {
  \bassFigureExtendersOn
  % Qui gli estensori non sono corretti, anche se il tempo è lo stesso
  <6 4->4 <6 4->4
  <5>8. r16 <6>8 <6\! 5->
}
>>

```



2.8.3 Notazione grafica

Gli elementi ritmici possono essere continuati da una linea di durata, che è rappresentata dal grob `DurationLine`. Gli stili possibili sono `'beam` (raggio), `'line` (linea), `'dashed-line` (linea tratteggiata), `'dotted-line` (linea punteggiata), `'zigzag`, `'trill` e `'none` (nessuno). La linea di durata può terminare con un gancio (solo nello stile a raggio) o con una freccia.

```
\layout {
  \context {
    \Voice
    \consists "Duration_line_engraver"
    \omit Stem
    \omit Flag
    \omit Beam
    \override NoteHead.duration-log = 2
  }
}

{
  a'1\- s2 r
  \once \override DurationLine.style = #'line
  a'1\- s2 r
  \once \override DurationLine.style = #'dashed-line
  \once \override DurationLine.dash-period = 2
  a'1\- s2 r
  \once \override DurationLine.style = #'dotted-line
  \once \override DurationLine.dash-period = 1
  \once \override DurationLine.bound-details.right.padding = 1
  a'1\- s2 r
  \once \override DurationLine.thickness = 2
  \once \override DurationLine.style = #'zigzag
  a'1\- s2 r
  \once \override DurationLine.style = #'trill
  a'1\- s2 r
  \once \override DurationLine.style = #'none
  a'1\- s2 r
  \once \override DurationLine.bound-details.right.end-style = #'arrow
  a'1\- s2 r
  \override DurationLine.bound-details.right.end-style = #'hook
  a'1\- s2 r
  \override DurationLine.details.hook-direction = #DOWN
  a'1\- s2 r
  \bar "|."
}
```



Problemi noti e avvertimenti

Se un grob `DurationLine` arriva alla fine di una partitura, per motivi tecnici gli elementi al termine della linea non appaiono. Un modo per aggirare il problema potrebbe essere:

```
\layout {
  \context {
    \Voice
    \consists "Duration_line_engraver"
    \omit Stem
    \omit Flag
    \omit Beam
    \override NoteHead.duration-log = 2
  }
}

lastEndStyle =
#(define-music-function (end-style)(symbol?)
#(
  \override DurationLine.stencil =
    #(lambda (grob)
      (let* ((orig (ly:grob-original grob))
             (siblings (if (ly:grob? orig)
                           (ly:spanner-broken-into orig) '())
                           (last-grob (if (pair? siblings) (last siblings) #f)))
             (if last-grob
                 (ly:grob-set-nested-property!
                  last-grob
                  '(bound-details right-broken end-style) end-style))
             duration-line::print))
      #}))

{
  \once \override DurationLine.bound-details.right.end-style = #'arrow
  \lastEndStyle #'arrow
  a'1\~
  \bar " | ."
}
```



2.8.4 Tecniche di arrangiamento contemporaneo

2.8.5 Nuove tecniche strumentali

2.8.6 Letture consigliate e partiture rilevanti

Questa sezione consiglia libri, esempi musicali e altre risorse utili nello studio della notazione musicale contemporanea.

Libri e articoli sulla notazione musicale contemporanea

- *Music Notation in the Twentieth Century: A Practical Guidebook* di Kurt Stone [W. W. Norton, 1980]

- *Music Notation: A Manual of Modern Practice* di Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* di Alfred Blatter [Schirmer, 2nd ed. 1997]

Partiture e esempi musicali

2.9 Notazione antica

Sál- ve, Re- gí- na, máter mi-se- ri cór- di- ae : Ad te cla- má mus, éx-
 su- les, fí li- i Hévae. Ad te suspi- rá- mus, ge- mén tes et flén- tes in
 hac la cri- márum vál- le. E- ia er go, Advo- cá- ta nóstra, illos tú-
 os mi se ri- cór- des ó- cu- los ad nos con- vér- te. Et Jé- sum, be- ne-
 díc- tum frúctum vén tris tú- i, nó- bis post hoc ex sí- li- um os- tén-
 de. O clémens : O pí- a : O dúl- cis Vírgo Ma- rí- a.

Il supporto per la notazione antica comprende funzionalità per la notazione mensurale, la notazione del canto gregoriano e la notazione quadrata di Kiev. Tali funzionalità sono disponibili sia modificando le proprietà stilistiche di alcuni oggetti grafici, come teste di nota e pause, sia usando uno dei contesti predefiniti per questi stili.

Molti oggetti grafici, come teste di nota e code, alterazioni, indicazioni di tempo e pause, forniscono una proprietà `style`, che può essere cambiata per emulare vari stili diversi di notazione antica. Vedere:

- [Teste di nota mensurali], pagina 460,
- [Alterazioni e armature di chiave mensurali], pagina 463,
- [Pause mensurali], pagina 462,
- [Chiavi mensurali], pagina 459,
- [Chiavi gregoriane], pagina 466,
- [Code mensurali], pagina 461,
- [Indicazioni di tempo mensurali], pagina 460.

Alcuni concetti sono presentati in modo specifico per la notazione antica:

- [Custodes], pagina 457,
- [Divisiones], pagina 467,
- [Legature], pagina 456.

Vedi anche

Glossario musicale: Sezione “custos” in *Glossario Musicale*, Sezione “legatura” in *Glossario Musicale*, Sezione “notazione mensurale” in *Glossario Musicale*.

Guida alla notazione: [Teste di nota mensurali], pagina 460, [Alterazioni e armature di chiave mensurali], pagina 463, [Pause mensurali], pagina 462, [Chiavi gregoriane], pagina 466, [Code mensurali], pagina 461, [Indicazioni di tempo mensurali], pagina 460, [Custodes], pagina 457, [Divisiones], pagina 467, [Legature], pagina 456.

2.9.1 Panoramica degli stili supportati

Per scrivere il canto gregoriano sono disponibili tre stili:

- *Editio Vaticana* è uno stile completo per il canto gregoriano, che riprende l’aspetto delle edizioni Solesmes, i libri ufficiali di canti del Vaticano dal 1904. LilyPond supporta tutti i segni della notazione usati in questo stile, incluse legature, *custodes* e segni speciali come il quilisma e l’oriscus.
- Lo stile *Editio Medicaea* offre alcune funzionalità utilizzate nelle edizioni Medicaea (o Ratisbona), che venivano usate prima delle edizioni Solesmes. Le differenze più rilevanti rispetto allo stile *Vaticana* sono le chiavi, che hanno dei tratti inclinati verso il basso, e le teste di nota, che sono quadrate e regolari.
- Lo stile *Hufnagel* (“chiodo del ferro di cavallo”) o *Gotico* replica lo stile di scrittura dei manoscritti tedeschi e mitteleuropei del Medioevo. Il nome deriva dalla forma essenziale della nota (la *virga*), che assomiglia a un piccolo chiodo.

Tre stili emulano l’aspetto dei manoscritti e delle stampe di musica mensurale del tardo Medioevo e del Rinascimento:

- Lo stile *Mensurale* assomiglia molto allo stile di scrittura usato nei manoscritti del tardo Medioevo e del primo Rinascimento, con le sue teste di nota a diamante, piccole e strette, e le sue pause che ricordano uno stile a mano libera.
- Lo stile *Neomensurale* è una versione rinnovata e stilizzata del precedente: le teste di nota sono più larghe e le pause sono fatte di linee dritte. Questo stile è particolarmente adatto agli incipit dei pezzi trascritti di musica mensurale.
- Lo stile *Petrucchi* prende il nome da Ottaviano Petrucci (1466-1539), il primo tipografo a usare i caratteri mobili per la musica (nel suo *Harmonice musices odhecaton*, 1501). Questo stile usa delle teste di nota più grandi rispetto agli altri stili mensurali.

Gli stili *Barocco* e *Classico* non sono stili completi ma differiscono dallo stile predefinito solo in alcuni dettagli: alcune teste di nota (Barocco) e la pausa di un quarto (Classico).

Solo lo stile mensurale ha alternative per tutti gli aspetti della notazione. Negli stili gregoriani non ci sono pause né code, perché questi segni non sono usati nella notazione per canto piano, mentre lo stile Petrucci non ha code o alterazioni specifiche.

Ogni elemento della notazione può essere modificato indipendentemente dagli altri elementi, dunque è possibile usare code mensurali, teste di nota Petrucci, pause classiche e chiavi vaticane nello stesso brano.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*, Sezione “coda” in *Glossario Musicale*.

2.9.2 Notazione antica – funzionalità comuni

Contesti predefiniti

Per la notazione mensurale e del canto gregoriano, sono disponibili i contesti predefiniti della voce e del rigo, che impostano tutti i vari segni della notazione su valori adatti per questi stili. Se i valori predefiniti sono soddisfacenti, si può procedere direttamente all’inserimento delle note senza preoccuparsi di come personalizzare un contesto. Consultare uno dei contesti predefiniti: *VaticanaVoice*, *VaticanaStaff*, *MensuralVoice* e *MensuralStaff*. Vedere anche:

- [Contesti del canto gregoriano], pagina 465,
- [Contesti mensurali], pagina 458.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*.

Guida alla notazione: [Contesti del canto gregoriano], pagina 465, [Contesti mensurali], pagina 458.

Legature

Una legatura è un simbolo grafico che rappresenta almeno due diverse note. Le legature sono apparse in origine nei manoscritti dei canti gregoriani per riferirsi alle sequenze ascendenti e discendenti di note sulla stessa sillaba. Sono utilizzate anche nella notazione mensurale.

Le legature si inseriscono *racchiudendole* tra `\[` e `\]`. Alcuni stili di legatura possono richiedere ulteriore sintassi di input specifica per un particolare tipo di legatura. Nell’impostazione predefinita l’incisore *LigatureBracket* pone una parentesi quadrata sopra la legatura.

```
\relative {
  \[ g' c, a' f d' \]
  a g f
  \[ e f a g \]
}
```



Sono disponibili altri due stili di legatura: lo stile vaticano per il canto gregoriano e lo stile mensurale per la musica mensurale (sono supportate solo le legature mensurali bianche e con alcune limitazioni). Per usare uno di questi stili, occorre sostituire nel contesto *Voice* l’incisore predefinito *Ligature_bracket_engraver* con uno degli incisori specializzati, come è spiegato in [Legature mensurali bianche], pagina 464, e [Legature di neumi quadrati gregoriani], pagina 469.

Vedi anche

Glossario musicale: Sezione “legatura” in *Glossario Musicale*.

Guida alla notazione: [Legature mensurali bianche], pagina 464, [Legature di neumi quadrati gregoriani], pagina 469.

Problemi noti e avvertimenti

La spaziatura richiesta per le legature non è ancora implementata e di conseguenza potrebbe risultare troppo spazio tra di esse. Anche l'interruzione di linea potrebbe essere insoddisfacente.

Il testo vocale potrebbe non allinearsi come desiderato quando si usano le legature.

Le alterazioni non devono apparire all'interno di una legatura, ma devono invece essere raccolte e visualizzate davanti ad essa.

La sintassi utilizza ancora lo stile deprecato “infix” `\[espressione musicale \]`. Per ragioni di coerenza, dovrà essere cambiato in stile “postfix” `nota\[... nota\]`.

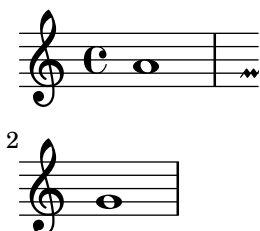
Custodes

Un *custos* (plurale: *custodes*; parola latina per “custode”) è un simbolo che appare alla fine di un rigo. Anticipa l'altezza della prima nota della linea seguente, aiutando quindi l'esecutore a gestire le interruzioni di linea durante l'esecuzione.

I custodes erano usati frequentemente nella notazione musicale fino al diciassettesimo secolo. Attualmente vengono usati solo in poche forme particolari di notazione musicale, per esempio nelle edizioni contemporanee dei canti gregoriani, come l'*Editio Vaticana*. Ci sono glifi diversi per il custos a seconda dei diversi stili di notazione.

Per inserire i custodes, porre l'incisore `Custos_engraver` nel contesto `Staff` del blocco `\layout` e modificare lo stile del custos con un `\override`, se necessario, come è illustrato nell'esempio seguente:

```
\score {
  \relative {
    a'1
    \break
    g
  }
  \layout {
    \context {
      \Staff
      \consists Custos_engraver
      \override Custos.style = #'mensural
    }
  }
}
```



Il glifo del custos viene selezionato tramite la proprietà `style`. Gli stili supportati sono `vaticana`, `medicaea`, `hufnagel` e `mensural`.

```
\new Lyrics \lyricmode {
  \markup { \column {
    \typewriter "vaticana "
    \line { " " \musicglyph "custodes.vaticana.u0" }
  } }
  \markup { \column {
```

```

\typewriter "medicaea "
\line { " " \musicglyph "custodes.medicaea.u0" }
}}
\markup { \column {
\typewriter "hufnagel "
\line { " " \musicglyph "custodes.hufnagel.u0" }
}}
\markup { \column {
\typewriter "mensural "
\line { " " \musicglyph "custodes.mensural.u0" }
}}
}

vaticana medicaea hufnagel mensural

```

Vedi anche

Glossario musicale: Sezione “custos” in *Glossario Musicale*.

Frammenti: Sezione “Ancient notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Custos” in *Guida al Funzionamento Interno*.

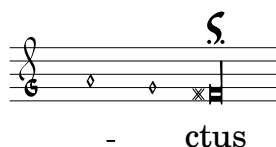
2.9.3 Scrivere la musica mensurale

Contesti mensurali

I contesti predefiniti `MensuralVoice` e `MensuralStaff` permettono di comporre un brano in stile mensurale. Questi contesti inizializzano tutte le proprietà di contesto e dei grob rilevanti su valori adeguati, in modo da poter procedere direttamente con l'inserimento del canto, come dimostra il seguente estratto:

```
\score {
  <<
    \new MensuralVoice = "discantus" \relative {
      \hide Score.BarNumber {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c\breve d\melismaEnd \]
        c\longa
        c\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```





Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*.

Chiavi mensurali

Le chiavi mensurali sono supportate tramite il comando `\clef`. Alcune chiavi usano lo stesso glifo, ma si differenziano soltanto per la linea su cui sono stampate. In tali casi, un numero al termine del nome permette di numerare queste chiavi, partendo dalla linea più bassa e salendo verso quella più alta.

```
\override NoteHead.style = #'vaticana.punctum
\clef "vaticana-do1"
c'1
```



```
\override NoteHead.style = #'medicaea.punctum
\clef "medicaea-do3"
c'1
```



```
\override NoteHead.style = #'hufnagel.punctum
\clef "hufnagel-fa2"
c'1
```



```
\override NoteHead.style = #'neomensural
\clef "neomensural-c4"
c'1
```



È possibile forzare manualmente il posizionamento del glifo di una chiave su una linea arbitraria, come è spiegato in [Chiave], pagina 17. Tutte le chiavi possibili sono descritte in Sezione A.11 [Stili della chiave], pagina 709.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*, Sezione “chiave” in *Glossario Musicale*.

Guida alla notazione: [Chiavi gregoriane], pagina 466, [Chiave], pagina 17.

File installati: `scm/parser-clef.scm`.

Frammenti: Sezione “Pitches” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Clef_engraver” in *Guida al Funzionamento Interno*, Sezione “Clef” in *Guida al Funzionamento Interno*, Sezione “ClefModifier” in *Guida al Funzionamento Interno*, Sezione “clef-interface” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

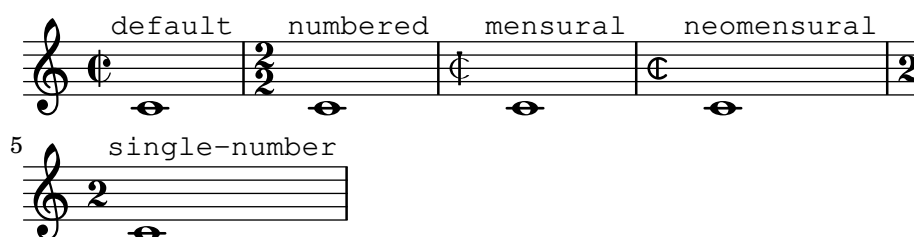
La chiave mensurale di Sol è associata alla chiave Petrucci di Sol.

Indicazioni di tempo mensurali

È disponibile un limitato supporto per i segni della musica mensurale (che sono simili ma non del tutto uguali alle indicazioni di tempo). I glifi sono collegati direttamente a specifiche frazioni temporali. In altre parole, per ottenere un certo segno mensurale col comando `\time n/m`, n e m devono essere scelti in base alla seguente tabella

<code>\time 4/4</code>	<code>\time 2/2</code>	<code>\time 6/4</code>	<code>\time 6/8</code>
C	C	C	C
<code>\time 3/2</code>	<code>\time 3/4</code>	<code>\time 9/4</code>	<code>\time 9/8</code>
O	⓪	⓪	⓪
<code>\time 4/8</code>	<code>\time 2/4</code>		
⓪	⓪		

Per selezionare le indicazioni di tempo antiche si usa la proprietà `style` del `grob TimeSignature`. Gli stili supportati sono `neomensural` e `mensural`. La tabella precedente utilizza lo stile `neomensural`. Gli esempi successivi mostrano le differenze tra gli stili:



[Indicazione di tempo], pagina 71, fornisce un'introduzione generale all'uso delle indicazioni di tempo.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*.

Guida alla notazione: [Indicazione di tempo], pagina 71.

Problemi noti e avvertimenti

I rapporti tra le durate delle note non possono cambiare con l'indicazione di tempo, perché non sono costanti. Per esempio, il rapporto di 1 breve = 3 semibrevis (*tempus perfectum*) può essere creato manualmente impostando

```
breveTP = #(ly:make-duration -1 0 3/2)
...
{ c\breveTP f1 }
```

`breveTP` viene impostato su $3/2$ moltiplicato $2 = 3$ volte una nota intera.

I simboli `mensural68alt` e `neomensural68alt` (simboli alternativi per $6/8$) non sono gestibili con `\time`. Usare invece `\markup {\musicglyph "timesig.mensural68alt" }`.

Teste di nota mensurali

Per la notazione antica, è possibile assegnare alle teste di nota uno stile diverso dal predefinito `default`. Per farlo si imposta la proprietà `style` dell'oggetto `NoteHead` su `baroque`, `neomensural`, `mensural`, `petrucci`, `blackpetrucci` o `semipetrucci`.

Differenze dello stile `baroque` rispetto allo stile `default`:

- Fornisce una testa di nota *maxima*.

- Utilizza una forma quadrata per le teste di nota `\breve`.

Differenze degli stili `neomensural`, `mensural` e `petrucci` rispetto allo stile baroque:

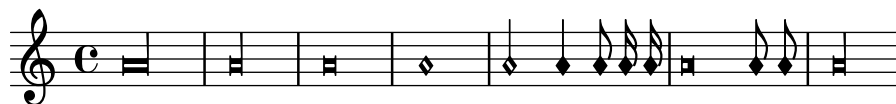
- Usano teste romboidali per le semibreve e tutte le durate più piccole.
- Centrano i gambi sulle teste di nota.

Lo stile `blackpetrucci` produce teste di nota usabili nella notazione mensurale nera o nelle sezioni “coloratio” della notazione mensurale bianca. Dato che lo stile delle teste di nota non influenza il numero delle code, in questo stile una semiminima deve essere scritta come `a8*2` e non come `a4`, altrimenti avrà lo stesso aspetto di una minima. Il moltiplicatore può essere diverso se si usa la “coloratio”, ad esempio per scrivere le terzine.

Usare lo stile `semipetrucci` per ottenere teste di nota semicolorate (breve, longa e maxima).

L'esempio seguente illustra lo stile `petrucci`:

```
\compressEmptyMeasures
\autoBeamOff
\override NoteHead.style = #'petrucci
a'\maxima a'\longa a'\breve a'1 a'2 a'4 a'8 a'16 a'
\override NoteHead.style = #'semipetrucci
a'\breve*5/6
\override NoteHead.style = #'blackpetrucci
a'8*4/3 a'
\override NoteHead.style = #'petrucci
a'\longa
```



Sezione A.9 [Stili delle teste di nota], pagina 707, fornisce una panoramica di tutti gli stili di teste di nota disponibili.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*, Sezione “teste di nota” in *Glossario Musicale*.

Guida alla notazione: Sezione A.9 [Stili delle teste di nota], pagina 707.

Code mensurali

Usare la proprietà `flag-style` del grob `Stem` per selezionare code antiche. Oltre allo stile predefinito default, per le code è supportato solo lo stile `mensural`.

```
\relative c' {
  \override Flag.style = #'mensural
  \override Stem.thickness = #1.0
  \override NoteHead.style = #'mensural
  \autoBeamOff
  c8 d e f c16 d e f c32 d e f s8
  c'8 d e f c16 d e f c32 d e f
}
```





Si noti che l'estremità più interna di ciascuna coda mensurale è allineata verticalmente con una linea del rigo.

Non esiste uno stile di coda specifico per la notazione neomensurale o Petrucci. Non esistono code nella notazione del canto gregoriano.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*, Sezione “coda” in *Glossario Musicale*.

Problemi noti e avvertimenti

L'allineamento verticale di ciascuna coda con una linea del rigo presuppone che i gambi terminino sempre esattamente su una linea o esattamente a metà tra due linee. Ciò potrebbe non essere sempre vero se si usano funzionalità di formattazione avanzata della notazione classica (che sono tuttavia non pertinenti per la notazione mensurale).

Pause mensurali

Usare la proprietà `style` del grob `Rest` per selezionare pause antiche. Gli stili antichi supportati sono `neomensural` e `mensural`.

L'esempio seguente illustra questi stili:

```
\compressEmptyMeasures
\override Rest.style = #'mensural
r\longa^"mensural" r\breve r1 r2 r4 r8 r16 s \break
\override Rest.style = #'neomensural
r\longa^"neomensural" r\breve r1 r2 r4 r8 r16
```



Non esistono pause di un trentaduesimo e di un sessantaquattresimo specifiche per gli stili mensurali o neomensurali; vengono quindi usate le pause dello stile predefinito.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*.

Guida alla notazione: [Pause], pagina 63.

Frammenti: Sezione “Ancient notation” in *Frammenti di codice*.

Problemi noti e avvertimenti

Il glifo della pausa di massima nello stile mensurale è in realtà una pausa di lunga perfetta; usare due (o tre) pause di lunga per ottenere una pausa di massima. Le pause di lunga non sono raggruppate automaticamente, dunque bisogna farlo manualmente usando le pause con altezza.

Alterazioni e armature di chiave mensurali

Lo stile mensural fornisce uno stile dei segni diesis e bemolle diverso da quello predefinito. La notazione mensurale usava raramente il segno di bequadro: al suo posto viene usato il segno diesis o bemolle appropriato. Per esempio, un Si bequadro nella tonalità di Fa maggiore viene indicato con un diesis. Tuttavia, se richiesto esplicitamente, il segno bequadro viene preso dallo stile vaticana.

mensural

♭ ✖

Il modo di usare questo stile è trattato in [Glifi di alterazione alternativi], pagina 36. È lo stile predefinito del contesto MensuralStaff.

Vedi anche

Glossario musicale: Sezione “notazione mensurale” in *Glossario Musicale*, Sezione “nomi delle altezze” in *Glossario Musicale*, Sezione “alterazione” in *Glossario Musicale*, Sezione “armatura di chiave” in *Glossario Musicale*.

Guida alla notazione: Sezione 1.1 [Altezze], pagina 1, [Alterazioni], pagina 6, [Alterazioni automatiche], pagina 29, [Glifi di alterazione alternativi], pagina 36, Sezione A.10 [Gruppi di glifi di alterazione], pagina 708, [Armatura di chiave], pagina 22.

Guida al funzionamento interno: Sezione “KeySignature” in *Guida al Funzionamento Interno*.

Alterazioni suggerite (*musica ficta*)

Nella musica europea antecedente il 1600, ai cantanti era richiesto di alterare cromaticamente le note di loro iniziativa in base a certe regole. Questa prassi è nota come *musica ficta*. Nelle trascrizioni moderne queste alterazioni vengono poste solitamente sopra la nota.

Tali alterazioni suggerite sono supportate e possono essere attivate impostando `suggestAccidentals` su vero.

```
\relative {
  fis' gis
  \set suggestAccidentals = ##t
  ais bis
}
```



In questo modo *ogni* successiva alterazione sarà trattata come *musica ficta* finché la proprietà non viene reimpostata con `\set suggestAccidentals = ##f`. È quindi più pratico usare `\once \set suggestAccidentals = ##t`, che può essere anche definito come una comoda scorciatoia:

```
ficta = { \once \set suggestAccidentals = ##t }
\score { \relative
  \new MensuralVoice {
    \once \set suggestAccidentals = ##t
    bes'4 a2 g2 \ficta fis8 \ficta e! fis2 g1
  }
}
```



Vedi anche

Guida al funzionamento interno: Sezione “Accidental_engraver” in *Guida al Funzionamento Interno*, Sezione “AccidentalSuggestion” in *Guida al Funzionamento Interno*.

Legature mensurali bianche

Il supporto per le legature mensurali bianche è presente ma limitato.

Per scriverle, sostituire nel blocco layout l’incisore `Ligature_bracket_engraver` con l’incisore `Mensural_ligature_engraver` all’interno del contesto `Voice`:

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```

Non esiste un ulteriore linguaggio di input per descrivere la forma di una legatura mensurale bianca. La forma è determinata unicamente dall’altezza e dalla durata delle note racchiuse. Questo approccio, sebbene possa richiedere a un nuovo utente un po’ di tempo per prenderci confidenza, ha il grande vantaggio che tutte le informazioni musicali della legatura sono note internamente. Non solo ciò è richiesto per avere un output MIDI corretto, ma permette anche la trascrizione automatica delle legature.

In alcuni punti due note consecutive possono essere rappresentate sia come due quadrati sia come un parallelogramma obliquo (forma *flexa*). In tali casi vengono usati automaticamente i due quadrati, ma si può scegliere una *flexa* impostando la proprietà `ligature-flexa` della seconda testa di nota. La lunghezza di una *flexa* si imposta tramite la proprietà della testa di nota `flexa-width`.

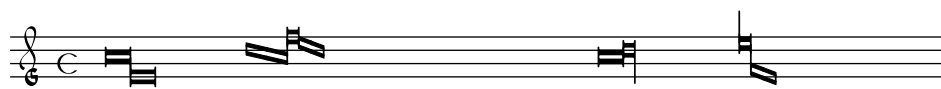
Per esempio,

```
\score {
  \relative {
    \set Score.timing = ##f
    \set Score.measureBarType = ""
    \override NoteHead.style = #'petrucci
    \override Staff.TimeSignature.style = #'mensural
    \clef "petrucci-g"
    \[ c'\maxima g \]
    \[ d'\longa
      \override NoteHead.ligature-flexa = ##t
      \once \override NoteHead.flexa-width = #3.2
      c\breve f e d \]
    \[ c\maxima d\longa \]
    \[ e1 a, g\breve \]
  }
  \layout {
    \context {
      \Voice
      \remove Ligature_bracket_engraver
```

```

\consists Mensural_ligature_engraver
}
}
}

```



Senza sostituire l'incisore `Ligature_bracket_engraver` con `Mensural_ligature_engraver`, la stessa musica appare così:



Vedi anche

Glossario musicale: Sezione “legatura” in *Glossario Musicale*.

Guida alla notazione: [Legature di neumi quadrati gregoriani], pagina 469, [Legature], pagina 456.

Problemi noti e avvertimenti

La spaziatura orizzontale delle legature può essere insoddisfacente. Le alterazioni potrebbero entrare in collisione con le note precedenti.

2.9.4 Scrivere il canto gregoriano

Quando si scrive un brano nella notazione del canto gregoriano, l'incisore `Vaticana_ligature_engraver` seleziona automaticamente le teste di nota adeguate, dunque non c'è bisogno di impostare esplicitamente lo stile della testa di nota. È tuttavia possibile modificare lo stile della testa di nota, per esempio su `vaticana.punctum` per produrre i neumi *punctum*. Analogamente, l'incisore `Mensural_ligature_engraver` raggruppa automaticamente le legature mensurali.

Vedi anche

Glossario musicale: Sezione “legatura” in *Glossario Musicale*.

Guida alla notazione: [Legature mensurali bianche], pagina 464, [Legature], pagina 456.

Contesti del canto gregoriano

Si possono usare i contesti predefiniti `VaticanaVoice` e `VaticanaStaff` per scrivere un brano di canto gregoriano nello stile Editio Vaticana. Questi contesti inizializzano tutte le proprietà dei grob e di contesto rilevanti impostandole su valori adatti, in modo da non dover far altro che inserire il canto, come dimostra il passo seguente:

```

\new VaticanaScore {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
  }
}

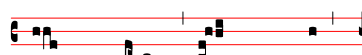
```

```

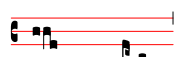
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}

\layout {
  indent = 0
  ragged-last = ##t
}

```



San- ctus, San- ctus,



San- ctus

Chiavi gregoriane

La tabella seguente mostra tutte le chiavi gregoriane supportate dal comando `\clef`. Alcune chiavi usano lo stesso glifo, ma si differenziano rispetto alla linea in cui sono collocate. In tali casi, si usa un numero in fondo al nome per numerare queste chiavi, dalla linea più bassa a quella più alta. È tuttavia possibile forzare manualmente la posizione del glifo della chiave su una linea arbitraria, come è spiegato in [Chiave], pagina 17. La nota che appare a destra di ogni chiave nella colonna degli esempi indica il c' (Do) relativo a quella chiave.

Descrizione	Chiavi supportate	Esempio
stile Editio Vaticana chiave di Do	vaticana-do1, vaticana-do2, vaticana-do3	
stile Editio Vaticana chiave di Fa	vaticana-fa1, vaticana-fa2	
stile Editio Medicaea chiave di Do	medicaea-do1, medicaea-do2, medicaea-do3	
stile Editio Medicaea chiave di Fa	medicaea-fa1, medicaea-fa2	
stile hufnagel chiave di Do	hufnagel-do1, hufnagel-do2, hufnagel-do3	
stile hufnagel chiave di Fa	hufnagel-fa1, hufnagel-fa2	
stile hufnagel chiave combinata Do/Fa	hufnagel-do-fa	

Vedi anche

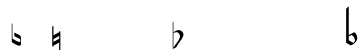
Glossario musicale: Sezione “chiave” in *Glossario Musicale*.

Guida alla notazione: [Chiave], pagina 17.

Alterazioni e armature di chiave gregoriane

Sono disponibili alterazioni per i tre diversi stili gregoriani:

vaticana medicaea hufnagel



Come si vede, non tutte le alterazioni sono supportate da ciascun stile. Quando si tenterà di accedere a un'alterazione non supportata, LilyPond passerà a uno stile diverso.

Come passare da uno stile all'altro è spiegato in [Glifi di alterazione alternativi], pagina 36.

Vedi anche

Glossario musicale: Sezione “alterazione” in *Glossario Musicale*, Sezione “armatura di chiave” in *Glossario Musicale*.

Guida alla notazione: Sezione 1.1 [Altezze], pagina 1, [Alterazioni], pagina 6, [Alterazioni automatiche], pagina 29, [Glifi di alterazione alternativi], pagina 36, [Armatura di chiave], pagina 22.

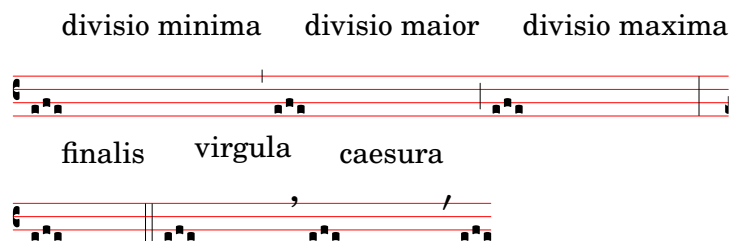
Guida al funzionamento interno: Sezione “KeySignature” in *Guida al Funzionamento Interno*.

Divisiones

Nella notazione del canto gregoriano non esistono pause; al loro posto si usano le [Divisiones], pagina 467.

Una *divisio* (plurale: *divisiones*; parola latina per “divisione”) è un simbolo di contesto usato per indicare la frase e la struttura delle sezioni della musica gregoriana. Il significato musicale di *divisio minima*, *divisio maior* e *divisio maxima* può essere descritto come pausa breve, media e lunga, un po' come per i segni di respiro (vedi [Respiri], pagina 147). Il segno *finalis* non segna soltanto la fine di un canto, ma viene anche usato frequentemente all'interno di un singolo canto antifonale/di risposta per contrassegnare la fine di ciascuna sezione.

Per usare le divisiones, includere il file `gregorian.ly`. Contiene le definizioni che si possono applicare inserendo semplicemente `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` e `\finalis` nei punti giusti dell'input. Alcune edizioni usano *virgula* o *caesura* al posto della *divisio minima*. Dunque `gregorian.ly` contiene anche una definizione per `\virgula` e `\caesura`.



Comandi predefiniti

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Vedi anche

Glossario musicale: Sezione “caesura” in *Glossario Musicale*, Sezione “divisio” in *Glossario Musicale*.

Guida alla notazione: [Respiri], pagina 147.

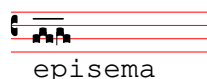
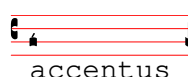
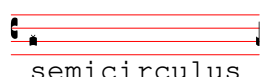
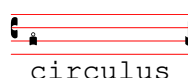
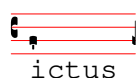
File installati: `ly/gregorian.ly`.

Segni di articolazione gregoriani

Oltre ai segni di articolazione standard descritti nella sezione [Articolazioni e abbellimenti], pagina 129, sono disponibili dei segni di articolazione progettati specificamente per la notazione in stile *Editio Vaticana*.

```
\new VaticanaScore {
  \new VaticanaVoice {
    \override TextScript.font-family = #'typewriter
    \override TextScript.font-shape = #'upright
    \override Script.padding = #-0.1
    a\ictus_"ictus " \bar "" \break
    a\circulus_"circulus " \bar "" \break
    a\semicirculus_"semicirculus " \bar "" \break
    a\accentus_"accentus " \bar "" \break
    \[ a_"episema" \epistemInitium \pes b \flexa a b \epistemFinis \flexa a \]
  }
}
```

```
\layout {
  indent = 0
  ragged-last = ##t
}
```



Vedi anche

Guida alla notazione: [Articolazioni e abbellimenti], pagina 129.

Frammenti: Sezione “Ancient notation” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Episema” in *Guida al Funzionamento Interno*, Sezione “EpisemaEvent” in *Guida al Funzionamento Interno*, Sezione “Episema-engraver” in *Guida al Funzionamento Interno*, Sezione “Script” in *Guida al Funzionamento Interno*, Sezione “ScriptEvent” in *Guida al Funzionamento Interno*, Sezione “Script-engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Alcune articolazioni sono posizionate troppo vicine verticalmente alle teste di nota corrispondenti.

Punti di aumentazione (morae)

I punti di aumentazione, chiamati anche *morae*, si aggiungono con la funzione musicale `\augmentum`. Si noti che `\augmentum` è implementata come funzione musicale unaria invece che come prefisso. Ciò significa che si applica soltanto all'espressione musicale immediatamente seguente. Per esempio, `\augmentum \virga c` non avrà effetti visibili. Scrivere invece `\virga \augmentum c` o `\augmentum {\virga c}`. È possibile scrivere anche `\augmentum {a g}` come scorciatoia per `\augmentum a \augmentum g`.

```
\new VaticanaScore {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



Vedi anche

Guida alla notazione: [Respiri], pagina 147.

Guida al funzionamento interno: Sezione “BreathingSign” in *Guida al Funzionamento Interno*.

Frammenti: Sezione “Ancient notation” in *Frammenti di codice*.

Legature di neumi quadrati gregoriani

LilyPond fornisce un supporto limitato alla notazione quadrata del canto gregoriano (secondo lo stile Editio Vaticana). Possono essere già scritte le legature fondamentali, ma ci sono ancora importanti limiti per una composizione tipografica professionale: tra gli altri, l'allineamento orizzontale di legature multiple, l'allineamento del testo vocale e una corretta gestione delle alterazioni.

Il supporto per i neumi gregoriani si abilita includendo il file `gregorian.ly` all'inizio del file. Così si hanno a disposizione vari comandi aggiuntivi per la produzione di simboli di neumi usati nella notazione del canto gregoriano.

Le teste di nota possono essere *modificate* e/o *unite*.

- La forma della testa di nota può essere modificata *anteponendo* al nome della nota uno dei seguenti comandi: `\virga`, `\strophæ`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.
- Le legature, ovvero le note unite insieme, si producono mettendo uno dei comandi di unione, `\pes` o `\flexa`, rispettivamente per il movimento verso l'alto e verso il basso, *tra* le note da unire.

Un nome di nota senza alcun qualificativo produrrà un *punctum*. Tutti gli altri neumi, inclusi quelli a nota singola con una forma diversa come la *virga*, sono considerati come legature e devono quindi essere posti tra `\[...]`.

Neumi a nota singola

- Il *punctum* è la forma principale della nota (nello stile *Vaticana*: un quadrato con una leggera curvatura per dare eleganza tipografica). Oltre al normale *punctum*, esiste anche la forma obliqua, chiamata *punctum inclinatum*, prodotta col prefisso `\inclinatum`. Il normale *punctum* può essere modificato con `\cavum`, che produce una nota vuota, e con `\linea`, che disegna linee verticali su entrambi i lati della nota.
- La *virga* ha un gambo discendente sul lato destro. È prodotta dal modificatore `\virga`.

Legature

Diversamente dalla maggior parte degli altri sistemi di notazione dei neumi, l'aspetto tipografico delle legature non è determinato direttamente dai comandi di input, ma segue certe convenzioni a seconda del significato musicale. Per esempio, una legatura di tre note con la forma musicale basso-alto-basso, come `\[a \pes b \flexa g \]`, produce un Torculus che è composto da tre teste Punctum, mentre la forma alto-basso-alto, come `\[a \flexa g \pes b \]`, produce un Porrectus con una forma flexa curvata e soltanto una testa Punctum. Non esiste un comando che permetta di gestire esplicitamente la forma flexa curvata; la decisione di quando utilizzarla viene presa automaticamente da LilyPond in base all'input musicale. L'idea di fondo di questo approccio è separare gli aspetti musicali dell'input dallo stile tipografico dell'output. In questo modo lo stesso input può essere riutilizzato per comporre la stessa musica in uno stile diverso di canto gregoriano.

Neumi liquescenti

Un'altra categoria primaria di note nel canto gregoriano è composta dai cosiddetti neumi liquescenti. Vengono usati in certe circostanze alla fine di una sillaba che termina con una lettera 'liquida', ovvero le consonanti che hanno un tono (le nasali, l, r, v, j e i loro dittonghi equivalenti). Dunque i neumi liquescenti non sono mai usati da soli (sebbene alcuni di questi possano essere prodotti così) e si trovano sempre al termine di una legatura.

I neumi liquescenti sono rappresentati graficamente in due modi diversi, più o meno intercambiabili: con una nota più piccola o "ruotando" la nota principale in su o in giù. Il primo si ottiene con una normale pes o flexa e modificando la forma della seconda nota: `\[a \pes \deminutum b \]`; il secondo si ottiene modificando la forma di un neuma a nota singola con `\auctum` e uno dei segni di direzione `\descendens` o `\ascendens`, per esempio: `\[\auctum \descendens a \]`.

Segni speciali

Una terza categoria di segni è composta da un piccolo gruppo di segni aventi un significato speciale (e solitamente poco conosciuto): il *quilisma*, l'*oriscus* e lo *strophicus*. Si ottengono antepoendo alla nota il modificatore corrispondente: `\quilisma`, `\oriscus` o `\strophica`.

In pratica, all'interno dei delimitatori della legatura `\[e \]`, è possibile ammassare qualsiasi numero di teste per formare una legatura singola; prefissi come `\pes`, `\flexa`, `\virga`, `\inclinatum`, etc., possono essere combinati a piacere. L'uso dell'insieme di regole che sta alla base della costruzione delle legature nella tabella precedente è estrapolato di conseguenza. È così possibile creare moltissime legature diverse.

L'uso di questi segni nella musica stessa segue alcune regole, che non controllate sono da LilyPond. Per esempio, il *quilisma* è sempre la nota centrale di una legatura ascendente, ma è assolutamente possibile, anche se sbagliato, produrre un *quilisma* di una nota singola.

Oltre ai segni delle note, `gregorian.ly` definisce anche i comandi `\versus`, `\responsum`, `\ij`, `\iij`, `\IJ` e `\IIJ`, che produrranno i caratteri corrispondenti, per esempio per essere usati nel testo vocale, come segni di sezione, etc. Questi comandi utilizzano speciali caratteri Unicode e funzionano soltanto se si usa un font che li supporti.

La tabella seguente illustra un insieme limitato ma rappresentativo di legature gregoriane, insieme ai frammenti di codice che le producono. La tabella si basa sulla tabella dei neumi del secondo volume dell'Antiphonale Romanum (*Liber Hymnarius*), pubblicato nel 1983 dai monaci di Solesmes. La prima colonna dà il nome della legatura, con la forma principale in grassetto e quella liquescente in corsivo. La terza colonna mostra il frammento di codice che produce la legatura, usando g, a e b come altezze di esempio.















Neumi a nota singola

Forme principali e <i>liquescenti</i>	Output	Codice LilyPond
Punctum	▪	<code>\[b \]</code>
	◻	<code>\[\cavum b \]</code>
	▣	<code>\[\linea b \]</code>
<i>Punctum Auctum Ascendens</i>	⤴	<code>\[\auctum \ascendens b \]</code>
<i>Punctum Auctum Descendens</i>	⤵	<code>\[\auctum \descendens b \]</code>
Punctum inclinatum	◆	<code>\[\inclinatum b \]</code>
<i>Punctum Inclinatum Auctum</i>	⤴◆	<code>\[\inclinatum \auctum b \]</code>
<i>Punctum Inclinatum Parvum</i>	◆•	<code>\[\inclinatum \deminutum b \]</code>
Virga	┘	<code>\[\virga b' \]</code>

Legature a due note

Clivis vel Flexa		<code>\[b \flexa g \]</code>
<i>Clivis Aucta Descendens</i>		<code>\[b \flexa \auctum \descendens g \]</code>
<i>Clivis Aucta Ascendens</i>		<code>\[b \flexa \auctum \ascendens g \]</code>
<i>Cephalicus</i>		<code>\[b \flexa \deminutum g \]</code>
Podatus/Pes		<code>\[g \pes b \]</code>
<i>Pes Auctus Descendens</i>		<code>\[g \pes \auctum \descendens b \]</code>
<i>Pes Auctus Ascendens</i>		<code>\[g \pes \auctum \ascendens b \]</code>
<i>Epiphonus</i>		<code>\[g \pes \deminutum b \]</code>
<i>Pes Initio Debilis</i>		<code>\[\deminutum g \pes b \]</code>
<i>Pes Auctus Descendens Initio Debilis</i>		<code>\[\deminutum g \pes \auctum \descendens b \]</code>

Legature a note multiple

Torculus		<code>\[a \pes b \flexa g \]</code>
<i>Torculus Auctus Descendens</i>		<code>\[a \pes b \flexa \auctum \descendens g \]</code>
<i>Torculus Deminutus</i>		<code>\[a \pes b \flexa \deminutum g \]</code>
<i>Torculus Initio Debilis</i>		<code>\[\deminutum a \pes b \flexa g \]</code>
<i>Torculus Auctus Descendens Initio Debilis</i>		<code>\[\deminutum a \pes b \flexa \auctum \descendens g \]</code>
<i>Torculus Deminutus Initio Debilis</i>		<code>\[\deminutum a \pes b \flexa \deminutum g \]</code>
Porrectus		<code>\[a \flexa g \pes b \]</code>
<i>Porrectus Auctus Descendens</i>		<code>\[a \flexa g \pes \auctum \descendens b \]</code>
<i>Porrectus Deminutus</i>		<code>\[a \flexa g \pes \deminutum b \]</code>
Climacus		<code>\[\virga b \inclinatum a \inclinatum g \]</code>
<i>Climacus Auctus</i>		<code>\[\virga b \inclinatum a \inclinatum \auctum g \]</code>
<i>Climacus Deminutus</i>		<code>\[\virga b \inclinatum a \inclinatum \deminutum g \]</code>
Scandicus		<code>\[g \pes a \virga b \]</code>
<i>Scandicus Auctus Descendens</i>		<code>\[g \pes a \pes \auctum \descendens b \]</code>

Scandicus Deminutus

\[g \pes a \pes \deminutum b \]

Segni speciali**Quilisma**

\[g \pes \quilisma a \pes b \]

Quilisma Pes Auctus Descendens\[\quilisma g \pes \auctum
\descendens b \]**Oriscus**

\[\oriscus b \]

Pes Quassus

\[\oriscus g \pes \virga b \]

Pes Quassus Auctus Descendens\[\oriscus g \pes \auctum
\descendens b \]**Salicus**

\[g \oriscus a \pes \virga b \]

Salicus Auctus Descendens\[g \oriscus a \pes \auctum
\descendens b \]**(Apo)stropha**

\[\stropha b \]

Stropha Aucta

\[\stropha \auctum b \]

Bistropha

\[\stropha b \stropha b \]

Tristropha\[\stropha b \stropha b
\stropha b \]*Trigonus*\[\stropha b \stropha b
\stropha a \]**Comandi predefiniti**

Sono supportati i seguenti prefissi: \virga, \stropha, \inclinatum, \auctum, \descendens, \ascendens, \oriscus, \quilisma, \deminutum, \cavum, \linea.

Problemi noti e avvertimenti

LilyPond supporta la notazione di Kiev dello stile sinodale, come è stato usato nella raccolta di canti stampata dal Santo Sinodo russo nel secondo decennio del Novecento e recentemente ristampato dal Moscow Patriarchate Publishing House. LilyPond non supporta le forme più antiche (meno comuni), utilizzate in Galizia per scrivere il canto liturgico ruteno.

Chiavi della notazione di Kiev

La notazione di Kievan utilizza una sola chiave, la chiave Tse-fa-ut, usata per indicare la posizione del Do (c):

```
\clef "kievan-do"
\kievanOn
c'
```



Vedi anche

Glossario musicale: Sezione “notazione di Kiev” in *Glossario Musicale*, Sezione “chiave” in *Glossario Musicale*.

Guida alla notazione: [Chiave], pagina 17.

Note della notazione di Kiev

Nella notazione quadrata di Kiev, occorre scegliere lo stile appropriato per le teste di nota e disabilitare i gambi e le code. Ciò si ottiene chiamando la funzione `\kievanOn`, che imposta le proprietà di teste di nota, gambi e code. Quando non sono più necessarie le teste di nota di Kiev, tali proprietà possono essere ripristinate tramite la funzione `\kievanOff`.

La nota finale nello stile di Kiev, solitamente al termine di un brano, può essere selezionata impostando la durata su `\longa`. Il segno recitativo di Kiev, usato per indicare il canto di varie sillabe su una nota, si seleziona impostando la durata su `\breve`. L'esempio seguente illustra le varie teste di nota di Kiev:

```
\autoBeamOff
\cadenzaOn
\kievanOn
b'1 b'2 b'4 b'8 b'\breve b'\longa
\kievanOff
b'2
```



Vedi anche

Glossario musicale: Sezione “notazione di Kiev” in *Glossario Musicale*, Sezione “testa di nota” in *Glossario Musicale*.

Guida alla notazione: Sezione A.9 [Stili delle teste di nota], pagina 707.

Problemi noti e avvertimenti

LilyPond determina automaticamente se una nota debba avere il gambo in su o in giù. Tuttavia quando si scrive un canto nella notazione quadrata è consuetudine dirigere i gambi nella stessa direzione all'interno di un singolo melisma. Ciò si può fare manualmente impostando la proprietà `direction` dell'oggetto `Stem`.

Alterazioni della notazione di Kiev

Lo stile kievian fornisce un segno di diesis e di bemolle diversi dallo stile predefinito. Nella notazione di Kiev non esiste il segno di bequadro. Il segno di diesis non viene usato nella musica sinodale ma può essere presente nei manoscritti precedenti. È stato incluso soprattutto per garantire la compatibilità.

```
\clef "kievan-do"
\set Staff.alterationGlyphs =
  #alteration-kievan-glyph-name-alist
bes' dis'
```



Vedi anche

Glossario musicale: Sezione “notazione di Kiev” in *Glossario Musicale*, Sezione “alterazione” in *Glossario Musicale*.

Guida alla notazione: [Alterazioni], pagina 6, [Alterazioni automatiche], pagina 29, [Glifi di alterazione alternativi], pagina 36, Sezione A.8 [Il font Emmentaler], pagina 693.

Stanghetta della notazione di Kiev

Alla fine di un brano in notazione di Kiev viene solitamente posta un’immagine decorativa, che si può chiamare la stanghetta finale nello stile di Kiev. Può essere richiamata con `\bar "k"`.

```
\kievanOn
\clef "kievan-do"
c' \bar "k"
```



Vedi anche

Guida alla notazione: Sezione 1.2.5 [Battute], pagina 105, Sezione A.8 [Il font Emmentaler], pagina 693.

Melismi della notazione di Kiev

Le note all’interno di un melisma di Kiev vengono solitamente posizionate una vicina all’altra mentre il melisma è separato da uno spazio bianco. Ciò permette al cantante di identificare velocemente le strutture melodiche del canto Znamenny. In LilyPond, i melismi sono trattati come legature e la spaziatura è implementata dall’incisore `Kievan_ligature_engraver`.

Quando si usano i contesti `KievanVoice` e `KievanStaff`, l’incisore `Kievan_ligature_engraver` viene abilitato automaticamente. In altri contesti, può essere invocato sostituendo l’incisore `Ligature_bracket_engraver` con `Kievan_ligature_engraver` del blocco layout:

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Kievan_ligature_engraver
  }
}
```

La spaziatura tra le note di una legatura di Kiev può essere controllata attraverso la proprietà `padding` di `KievanLigature`.

L'esempio seguente illustra l'uso delle legature di Kiev:

```
% Impostazioni dei font cirillici
\paper {
  property-defaults.fonts.serif = "Linux Libertine O,serif"
}

\score {
  <<
    \new KievanVoice = "melody" \relative c' {
      \cadenzaOn
      e2 \[ e4( d4 ) \] \[ c4( d e d ) \] e1 \bar "k"
    }
    \new Lyrics \lyricsto "melody" {
      Га -- ври -- и -- лу
    }
  >>
}
```



Vedi anche

Glossario musicale: Sezione “legatura” in *Glossario Musicale*.

Guida alla notazione: [Legature mensurali bianche], pagina 464, [Legature di neumi quadrati gregoriani], pagina 469, [Legature], pagina 456.

Problemi noti e avvertimenti

La spaziatura orizzontale delle legature è insoddisfacente.

2.9.6 Lavorare con la musica antica – scenari e soluzioni

Lavorare con la musica antica richiede spesso compiti particolari che differiscono notevolmente dalla notazione moderna per cui LilyPond è progettato. Nel resto di questa sezione vengono delineati vari scenari tipici e le loro possibili soluzioni:

- come creare gli incipit (ovvero materiale introduttivo per indicare come appariva l'originale) a trascrizioni moderne della musica mensurale;
- come ottenere la formattazione *Mensurstriche* usata frequentemente per le trascrizioni moderne della musica polifonica;
- come trascrivere il canto gregoriano nella notazione moderna;
- come generare la notazione sia antica che moderna dallo stesso file sorgente.

Incipit

Nella trascrizione di musica mensurale in notazione moderna è consuetudine inserire un'indicazione di come apparivano le pause e la nota o le note iniziali della versione originale (incluse le chiavi originali). Viene chiamata *incipit*. Il comando `\incipit` usa l'indent del rigo principale per impostare la larghezza occupata dall'incipit, e `incipit-width` per impostare la larghezza del rigo dell'incipit.

```
\score {
```

```

\new Staff <<
  \new Voice = Tenor {
    \set Staff.instrumentName = "Tenor"
    \override Staff.InstrumentName.self-alignment-X = #RIGHT
    \incipit { \clef "mensural-c4" \key f \major r\breve r1 c'1 }
    \clef "treble_8"
    \key f \major
    R1 r2 c'2 |
    a4. c'8
  }
  \new Lyrics \lyricsto Tenor { Cyn -- thia your }
>>
\layout
{
  indent = 5\cm
  incipit-width = 3\cm
}
}

```



Problemi noti e avvertimenti

Attenzione: `instrumentName` deve essere impostato perché l'incipit venga prodotto. Se non è richiesto il nome di uno strumento, usare `\set Staff.instrumentName = ""`.

Formattazione mensurstriche

Mensurstriche ('linee di misurazione') è il termine accettato per le stanghette che collegano i righi di un sistema senza attraversare i righi stessi. È un modo per preservare l'aspetto ritmico dell'originale, ovvero non dover interrompere note sincopate che si trovano al cambio di battuta, continuando a fornire la funzione di orientamento che le stanghette hanno.

La formattazione mensurale, in cui le stanghette non appaiono sui righi ma nello spazio tra i righi, si può ottenere usando `StaffGroup` al posto di `ChoirStaff`. La stanghetta sui righi viene nascosta con `\hide`.

```

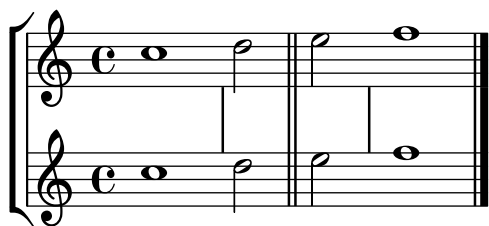
\layout {
  \context {
    \Staff
    measureBarType = "-span|"
  }
}

music = \fixed c'' {
  c1
  d2 \section e2
  f1 \fine
}

\new StaffGroup <<
  \new Staff \music
  \new Staff \music

```

>>



Trascrivere il canto gregoriano

Il canto gregoriano può essere trascritto in notazione moderna con alcune semplici modifiche.

Gambi. I gambi possono essere tolti del tutto rimuovendo col comando `\remove` l'incisore `Stem_engraver` dal contesto `Voice`:

```
\layout {
  ...
  \context {
    \Voice
    \remove Stem_engraver
  }
}
```

Tempo. Per i canti senza metro esistono varie alternative.

Si può togliere l'incisore `Time_signature_engraver` dal contesto `Staff` senza alcun effetto collaterale negativo. Se invece lo si rende trasparente, rimarrà uno spazio bianco nella partitura, perché l'indicazione di tempo prenderà comunque spazio.

In molti casi, usare `\set Score.timing = ##f` darà buoni risultati. Un'altra possibilità è l'uso di `\cadenzaOn` e `\cadenzaOff`.

Per togliere le stanghette, l'approccio radicale è togliere l'incisore `Bar_engraver` dal contesto `Staff`. Ma si può anche usare `\hide BarLine`, se si desidera una stanghetta saltuariamente.

Un tipo comune di trascrizione è il canto recitativo, dove le note ripetute sono indicate con una singola breve. Il testo per il tono della recitazione può essere trattato in due modi diversi: o impostato come una singola sillaba allineata a sinistra:

```
chant = \relative {
  \clef "G_8"
  c'\breve c4 b4 a c2 c4 \divisioMaior
  c\breve c4 c f, f \finalis
}

verba = \lyricmode {
  \once \override LyricText.self-alignment-X = #-1
  "Noctem quietam et" fi -- nem per -- fec -- tum
  \once \override LyricText.self-alignment-X = #-1
  "concedat nobis Dominus" om -- ni -- po -- tens.
}

\score {
  \new Staff <<
  \new Voice = "melody" \chant
  \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
```

Noctem quietam et finem perfectum concedat nobis Dominus omnipo-

In alcuni stili di trascrizione, i gambi vengono usati occasionalmente, per esempio per indicare la transizione da un recitativo di un singolo tono a un atto melodico fisso. In questi casi, si può usare `\hide Stem` o `\override Stem.length = #0` invece di togliere l'incisore `Stem_engraver` e ripristinare il gambo quando serve col corrispondente `\undo \hide Stem`.

```

verba = \lyricmode {
  No -- ctem qui -- e -- tam et fi -- nem per -- fec -- tum
  con -- ce -- dat no -- bis Do -- mi -- nus om -- ni -- po -- tens.
}

```

```
\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics \lyricsto "melody" \verba
  >>
  \layout {
    \context {
      \Staff
```

Per evitare che le sillabe di larghezza diversa (come “-ri” e “-rum”) si estendano sui gruppi di note delle sillabe in modo non uniforme, la proprietà 'X-extent dell’oggetto LyricText può essere impostata su un valore fisso. Un altro modo, più scomodo, consiste nell’aggiungere le sillabe come elementi \markup. Se sono necessari ulteriori aggiustamenti, si può fare facilmente con le pause spaziatrici s.

```

\spiritus = \relative {
  \time 1/4
  \override Lyrics.LyricText.X-extent = #'(0 . 3)
  d'4 \tuplet 3/2 { f8 a g } g a a4 g f8 e
  d4 f8 g g8 d f g a g f4 g8 a a4 s
  \tuplet 3/2 { g8 f d } e f g a g4
}

\spirLyr = \lyricmode {
  Spi -- ri -- _ _ tus _ Do -- mi -- ni _ re -- ple -- _ vit _
  or -- _ bem _ ter -- ra -- _ rum, al -- _ _ le -- _ lu
  -- _ ia.
}

\score {
  \new Staff <<
    \new Voice = "chant" \spiritus
    \new Lyrics = "one" \lyricsto "chant" \spirLyr
  >>
  \layout {
    \context {
      \Staff
      \remove Time_signature_engraver
      \override BarLine.X-extent = #'(-1 . 1)
      \hide Stem
      \hide Beam
      \hide BarLine
      \hide TupletNumber
    }
  }
}

```



Antico e moderno da un unico sorgente

Usare le etichette per produrre musica mensurale e moderna dallo stesso sorgente

Usando le etichette (tag), è possibile usare la stessa musica per produrre sia la musica mensurale che quella moderna. In questo frammento, viene introdotta la funzione `menrest`, che permette alle pause mensurali di essere posizionate precisamente sul rigo come nell'originale, ma con le pause moderne nella posizione standard. Le etichette vengono usate per produrre diversi tipi di stanghetta alla fine della musica, ma possono essere usate anche quando sono necessarie altre differenze: per esempio se si vogliono usare “pause d'intero” (`R1`, `R\breve`, etc.) nella musica moderna, ma pause normali (`r1`, `r\breve`, etc.) nella versione mensurale. La conversione di musica mensurale nel suo equivalente moderno viene solitamente chiamata trascrizione.

```
\layout {
  line-width = 150\mm
}

menrest = #(define-music-function (note) (ly:music?)
  #{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
  #})

Be = \tag #'mod
  #(begin
    (ly:expect-warning (G_ "stem does not fit in beam"))
    (ly:expect-warning (G_ "beam was started here"))
    (make-span-event 'BeamEvent START))

Am = \tag #'mod ]

MenStyle = {
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
  \omit Slur
  \omit Beam
}

finalis = \section

Music = \relative c'' {
```



```

\key f \major
g1 d'2 \menrest bes4 bes a2 \menrest r4 g4 fis4. fis8 fis4 fis \break
g e f4.([ g8] a4[ g8 f] g2.\Be fis8 e\Am fis2) g\breve \finalis
}

MenLyr = \lyricmode {
  So farre, deere life, deare life,
  from thy bright beames ab- en- ted,
}
ModLyr = \lyricmode {
  So far, dear life, dear life,
  from your bright beams ab -- sen -- ted, --
}

\score {
  \keepWithTag #'mens {
    <<
    \new PetrucciStaff {
      \new PetrucciVoice = "Cantus" {
        \clef "petrucci-c1" \time 4/4 \MenStyle \Music
      }
    }
    \new Lyrics \lyricsto "Cantus" \MenLyr
  } >>
}
\layout {
  \context {
    \PetrucciVoice
    % No longer necessary starting with version 2.25.23.
    \override Flag.style = #'mensural
  }
}

\markup\vspace #1

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
    \new Staff {
      \new Voice = "Sop" \with {
        \remove "Note_heads_engraver"
        \consists "Completion_heads_engraver"
        \remove "Rest_engraver"
        \consists "Completion_rest_engraver"
      } \shiftDurations 1 0 { \time 2/4 \autoBeamOff \Music }
    }
    \new Lyrics \lyricsto "Sop" \ModLyr
  } >>
}

```

The image displays four musical staves, each with a different notation system for the same lyrics: "So farre, deere life, deare life, from thy bright beames ab-fen-ted,".

- Staff 1:** Uses a C-clef and a common time signature (C). Notes are represented by diamond shapes. Some notes have a cross (x) through them, and there are some unusual symbols like a double cross (X) and a double dot (..).
- Staff 2:** Uses a C-clef and a common time signature (C). Notes are represented by diamond shapes. Some notes have a cross (x) through them, and there are some unusual symbols like a double cross (X) and a double dot (..).
- Staff 3:** Uses a G-clef and a 2/4 time signature. Notes are represented by standard musical notation (quarter notes, eighth notes, etc.).
- Staff 4:** Uses a G-clef and a 2/4 time signature. Notes are represented by standard musical notation (quarter notes, eighth notes, etc.).

2.10 World music

Questo capitolo tratta la notazione delle musiche tradizionali diverse da quelle occidentali.

2.10.1 Notazione comune per la musica non occidentale

Questa sezione spiega come scrivere e stampare partiture musicali che non appartengono alla tradizione classica occidentale, chiamata anche *Periodo di pratica comune*.

Estensione dei sistemi di notazione e di accordatura

La notazione classica standard (nota anche come notazione del *Periodo di pratica comune*) viene usata in tutti i generi musicali, non solo nella musica 'classica' occidentale. Questa notazione è trattata in Sezione 1.1.1 [Inserimento delle altezze], pagina 1, mentre i vari nomi di nota che possono essere usati sono spiegati in [Nomi delle note in altre lingue], pagina 8.

Tuttavia, molti generi musicali non occidentali (e alcuni tipi di musica folk occidentale e tradizionale) utilizzano sistemi di accordatura alternativi o estesi che non rientrano facilmente nella notazione classica standard.

In alcuni casi viene usata comunque la notazione standard, mentre le differenze di altezza restano implicite. Per esempio, la *musica araba* viene rappresentata con le alterazioni standard di un semitono e di un quarto di tono, e le esatte alterazioni di altezza sono determinate in base al contesto. Vengono usati solitamente i nomi italiani delle note, mentre il file di inizializzazione `arabic.ly` fornisce un comodo insieme di macro e definizioni che estendono la notazione standard. Maggiori dettagli in Sezione 2.10.2 [Musica araba], pagina 486.

Altri generi musicali richiedono notazioni estese o uniche. La *musica classica turca* o musica ottomana, per esempio, usa forme melodiche note come *makamlar*, i cui intervalli sono basati su divisioni di 1/9 del tono intero. Usa comunque le note sul rigo standard occidentale, ma con alterazioni speciali presenti esclusivamente nella musica turca, definite nel file `makam.ly`. Maggiori informazioni sulla musica classica turca e sui *makamlar* in Sezione 2.10.3 [Musica classica turca], pagina 491.

Per trovare i file di inizializzazione come `arabic.ly` o `makam.ly` nel proprio sistema, leggere Sezione "Altre fonti di informazione" in *Manuale di Apprendimento*.

Frammenti di codice selezionati

Esempio di makam

Makam è un tipo di melodia proveniente dalla Turchia che usa alterazioni microtonali di 1/9. Consultare il file di inizializzazione ‘ly/makam.ly’ per vedere come sono definiti i nomi delle altezze e le alterazioni.

```
\include "makam.ly"

\relative c' {
  \set Staff.keyAlterations = #`((6 . , (- KOMA)) (3 . , BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Vedi anche

Glossario musicale: Sezione “Periodo di pratica comune” in *Glossario Musicale*, Sezione “makamlar” in *Glossario Musicale*.

Manuale di apprendimento: Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.1.1 [Inserimento delle altezze], pagina 1, [Nomi delle note in altre lingue], pagina 8, Sezione 2.10.2 [Musica araba], pagina 486, Sezione 2.10.3 [Musica classica turca], pagina 491.

2.10.2 Musica araba

Questa sezione evidenzia le questioni relative alla notazione della musica araba.

Referenze per la musica araba

Finora la musica araba è stata soprattutto una tradizione orale. La musica, se trascritta, viene solitamente schematizzata in una sorta di abbozzo, sul quale gli esecutori hanno molta libertà di improvvisazione. Ma, per poter comunicare e preservare la musica araba, l’uso della notazione accidentale, con alcune variazioni, è sempre più diffuso.

Alcuni elementi della notazione musicale occidentale, come la trascrizione di accordi e parti indipendenti, non sono necessari per scrivere i brani arabi più tradizionali. Ci sono tuttavia alcune questioni differenti, come il bisogno di indicare gli intervalli intermedi che si trovano tra un semitono e un tono, oltre agli intervalli minori e maggiori usati nella musica occidentale. C’è anche il bisogno di raggruppare e indicare un gran numero di diversi maqam (modi) che fanno parte della musica araba.

In generale, la notazione della musica araba non tenta di indicare precisamente gli elementi microtonali presenti nella pratica musicale.

Varie questioni rilevanti per la musica araba sono trattate in altre sezioni della documentazione:

- I nomi delle note e le alterazioni (inclusi i quarti di tono) possono essere modificati come è spiegato in Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.
- Ulteriori armature di chiave possono essere adattate come descritto in [Armatura di chiave], pagina 22.

- Indicazioni di tempo complesse potrebbero obbligare a raggruppare le note manualmente, come descritto in [Travature manuali], pagina 101.
- I *takasim*, improvvisazioni in tempo libero, possono essere scritti omettendo le stanghette, come è spiegato in [Musica in tempo libero], pagina 79.

Vedi anche

Guida alla notazione: Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485, [Armatura di chiave], pagina 22, [Travature manuali], pagina 101.

Frammenti: Sezione “World music” in *Frammenti di codice*.

Nomi delle note in arabo

I nomi più tradizionali delle note in arabo possono essere piuttosto lunghi e non sono adatti alla scrittura musicale, quindi non vengono suati. I nomi delle note in inglese non sono molto conosciute nell'istruzione musicale araba, quindi al loro posto sono usati i nomi delle note in italiano, ovvero i nomi usati nel solfeggio (do, re, mi, fa, sol, la, si); si possono usare anche le alterazioni. I nomi delle note in italiano e le alterazioni sono spiegati in [Nomi delle note in altre lingue], pagina 8; l'uso della notazione occidentale standard per scrivere musica non occidentale è trattato in Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

Ecco un esempio di come può essere scritta la scala araba *rast*:

```
\include "arabic.ly"
\relative {
  do' re misb fa sol la sisb do sisb la sol fa misb re do
}
```



Il simbolo di semibemolle non corrisponde al simbolo usato nella notazione araba. Per ottenere il simbolo arabo di semibemolle, si può usare il simbolo `\dwn`, definito in `arabic.ly`, prima di un simbolo di bemolle. L'aspetto del simbolo di semibemolle nell'armatura di chiave non può essere modificato con questo metodo.

```
\include "arabic.ly"
\relative {
  \set Staff.extraNatural = ##f
  dod' dob dosd \dwn dob dobsb dodsd do do
}
```



Vedi anche

Guida alla notazione: [Nomi delle note in altre lingue], pagina 8, Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

Frammenti: Sezione “World music” in *Frammenti di codice*.

Armature di chiave arabe

Oltre alle armature di chiave minori e maggiori, sono definite le seguenti armature in `arabic.ly`: *bayati*, *rast*, *sikah*, *iraq* e *kurd*. Queste armature definiscono un piccolo numero di gruppi di *maqam* invece del gran numero di *maqam* di uso comune.

In generale, un maqam usa l'armatura di chiave del suo gruppo o di un gruppo vicino, e le variazioni di alterazioni sono contrassegnate nel corso della musica.

Per esempio, per indicare l'armatura di chiave di un brano maqam muhayer:

```
\key re \bayati
```

dove *re* è l'altezza predefinita del maqam muhayer, e *bayati* è il nome del maqam di base nel gruppo.

Sebbene l'armatura di chiave indichi il gruppo, di solito si precisa nel titolo il maqam più specifico, dunque in questo esempio il nome di maqam muhayer dovrebbe apparire nel titolo.

Altri maqam nello stesso gruppo bayati, come mostrato nella tabella in basso (bayati, hussaini, saba, and ushaq), possono essere indicati nello stesso modo. Sono tutte variazioni del maqam di base e più comune nel gruppo, ovvero bayati. Solitamente differiscono dal maqam di base nei tetracordi superiori, o in certi dettagli di disposizione che non cambiano fondamentalmente la loro natura.

L'altro maqam nello stesso gruppo (Nawa) è correlato con bayati per la modulazione, indicata in tabella tra parentesi per quei maqam che sono modulazioni del loro maqam fondamentale. I maqam arabi permettono soltanto modulazioni limitate, a causa della natura degli strumenti musicali arabi. Nawa può essere indicato nel modo seguente:

```
\key sol \bayati
```

Nella musica araba, lo stesso termine, come bayati, usato per indicare un gruppo maqam, è anche un maqam che è solitamente il più importante nel gruppo, e può anche essere considerato come il maqam di base.

Ecco un raggruppamento consigliato che mostra le armature di chiave dei più comuni maqam:

gruppo maqam	tonalità	finalis	Altri maqam nel gruppo (finalis)
ajam	major	sib	jaharka (fa)
bayati	bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz	kurd	re	shahnaz, shad arban (sol), hijazkar (do)
iraq	iraq	sisb	-
kurd	kurd	re	hijazkar kurd (do)
nahawand	minor	do	busalik (re), farah faza (sol)
nakriz	minor	do	nawa athar, hisar (re)
rast	rast	do	mahur, yakah (sol)
sikah	sikah	misb	huzam

Frammenti di codice selezionati

Armature di chiave non tradizionali

Il comando `\key` comunemente usato imposta la proprietà `keyAlterations` del contesto `Staff`. Per creare armature di chiave non standard, tale proprietà va impostata esplicitamente.

Il formato di questo comando è una lista:

```
\set Staff.keyAlterations = #`(((ottava . grado) . alterazione) ((ottava .  
grado) . alterazione) ...)
```

dove, per ogni elemento della lista, *ottava* indica l'ottava (0 è l'ottava dal Do centrale al Si precedente), *grado* indica la nota all'interno dell'ottava (0 significa Do e 6 significa Si) e *alterazione* può essere `,SHARP`, `,FLAT`, `,DOUBLE-SHARP` etc.

Altrimenti, usando il formato breve per ogni elemento della lista, `(grado . alterazione)`, ciò indica che la stessa alterazione deve essere presente in tutte le ottave. Per le scale microtonalidove

un “diesis” non è 100 centesimi, alterazione si riferisce alla proporzione di un duecentesimo di tono intero.

```
\include "arabic.ly"

\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                (1 . ,SEMI-FLAT)
                                (2 . ,FLAT)
                                (5 . ,FLAT)
                                (6 . ,SEMI-FLAT))

  % \set Staff.extraNatural = ##f
  re reb \down reb resd
  dod dob dosd \down dob |
  dobsb dodsd do do |
}
```



Vedi anche

Glossario musicale: Sezione “maqam” in *Glossario Musicale*, Sezione “bayati” in *Glossario Musicale*, Sezione “rast” in *Glossario Musicale*, Sezione “sikah” in *Glossario Musicale*, Sezione “iraq” in *Glossario Musicale*, Sezione “kurd” in *Glossario Musicale*.

Guida alla notazione: [Armatura di chiave], pagina 22.

Manuale di apprendimento: Sezione “Altezze e armature di chiave” in *Manuale di Apprendimento*.

Guida al funzionamento interno: Sezione “KeySignature” in *Guida al Funzionamento Interno*.

Frammenti: Sezione “World music” in *Frammenti di codice*, Sezione “Pitches” in *Frammenti di codice*.

Indicazioni di tempo arabe

Alcune forme di musica classica araba e turca, come *Semai*, usano indicazioni di tempo insolite, come 10/8. Ciò può determinare un raggruppamento automatico delle note piuttosto diverso dalle partiture esistenti, dove le note possono non essere raggruppate sul tempo, ma in un modo che è difficile da cogliere aggiustando la disposizione automatica delle travature. L’alternativa consiste nel disattivare la disposizione automatica delle travature e disporre le travature manualmente. Anche se non è richiesto di riprodurre lo stile di una partitura esistente, può essere comunque opportuno regolare il comportamento della disposizione automatica delle travature e/o usare indicazioni di tempo composto.

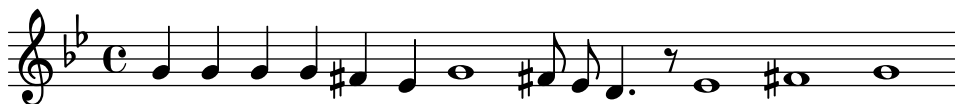
Frammenti di codice selezionati

Improvvisazione araba

Per improvvisazioni o *taqasim* a tempo libero, si può omettere l’indicazione di tempo e usare `\cadenzaOn`. Può essere necessaria la modifica dello stile delle alterazioni, perché l’assenza delle stanghette farà sì che l’alterazione sia contrassegnata una volta sola. Ecco un esempio di quello che potrebbe essere l’inizio di un’improvvisazione *hijaz*:

```
\include "arabic.ly"
```

```
\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Vedi anche

Glossario musicale: Sezione “semai” in *Glossario Musicale*, Sezione “taqasim” in *Glossario Musicale*.

Guida alla notazione: [Travature manuali], pagina 101, [Travature automatiche], pagina 88, [Musica in tempo libero], pagina 79, [Alterazioni automatiche], pagina 29, [Impostare il comportamento delle travature automatiche], pagina 91, [Indicazione di tempo], pagina 71.

Frammenti: Sezione “World music” in *Frammenti di codice*.

Esempio di musica araba

Ecco un modello che usa l’inizio di un *Semai* turco, conosciuto nell’educazione musicale araba, per illustrare le peculiarità della notazione musicale araba, come gli intervalli intermedi e i modi inusuali discussi in questa sezione.

```
\include "arabic.ly"
\score {
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
  \relative {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re'4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
  }
}
```



Vedi anche

Frammenti: Sezione “World music” in *Frammenti di codice*.

Lecture complementari sulla musica araba

1. *The music of the Arabs* di Habib Hassan Touma [Amadeus Press, 1996], contiene uno studio dei maqam e il loro metodo di classificazione.

Ci sono anche vari siti web che spiegano i maqam e alcuni offrono degli esempi audio, come:

- <https://www.maqamworld.com/>
- <https://www.turath.org/>

Nonostante ci sia un generale consenso sui criteri di classificazione dei maqam che sono collegati tra loro a causa di comuni tetracordi inferiori o della modulazione, i metodi di classificazione possono variare in alcuni dettagli.

2. Non c'è una coerenza perfetta, talvolta perfino nello stesso testo, su come specificare le armature di chiave per certi maqam. È tuttavia pratica comune l'utilizzo di una armatura di chiave per gruppo, invece di un'armatura diversa per ogni maqam.

I metodi dei seguenti autori per l'Oud, il liuto arabo, contengono esempi di composizioni in gran parte turche e arabe.

- Charbel Rouhana
- George Farah
- Ibrahim Ali Darwish Al-masri

2.10.3 Musica classica turca

Questa sezione evidenzia le questioni rilevanti per la notazione di musica classica turca.

Riferimenti per la musica classica turca

La musica classica turca si è sviluppata nell'impero ottomano in un periodo più o meno contemporaneo a quello della musica classica in Europa, e ha proseguito nei secoli ventesimo e ventunesimo come una tradizione viva e distinta, con le sue forme compositive, la sua teoria e i propri stili di esecuzione. Tra le sue principali peculiarità c'è l'uso degli intervalli microtonali basati sui ‘commi’ di $1/9$ di un tono, dal quale vengono costruite le forme melodiche note come *makam* (plurale *makamlar*).

Alcune questioni rilevanti per la musica classica turca sono trattate in altre sezioni della documentazione:

- I nomi per le note speciali e le alterazioni sono spiegati in Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

Nomi delle note in turco

Nella musica classica turca le altezze tradizionalmente hanno nomi unici, e a causa della divisione del tono in nove parti i makamlar usano un insieme di intervalli completamente diverso dalle scale e dai modi occidentali: *koma* ($1/9$ di un tono), *eksik bakiye* ($3/9$), *bakiye* ($4/9$), *küçük mücenneb* ($5/9$), *büyük mücenneb* ($8/9$), *tanîni* (un tono intero) e *artık ikili* ($12/9$ o $13/9$ di un tono).

Dal punto di vista della notazione moderna conviene usare le note standard occidentali sul rigo (do, re, mi, . . .) con delle alterazioni speciali che alzano o abbassano le note di intervalli di $1/9$, $4/9$, $5/9$ e $8/9$ di tono. Queste alterazioni sono definite nel file `makam.ly`.

La tabella seguente elenca:

- il nome di queste alterazioni speciali,

- il suffisso di alterazione da aggiungere alle note,
- e l'alterazione di altezza rappresentata come una frazione dell'intero tono.

Nome alterazione	suffisso	alterazione
------------------	----------	-------------

büyük mücenneb (diesis)	-bm	+8/9
küçük mücenneb (diesis)	-k	+5/9
bakiye (sharp)	-b	+4/9
koma (sharp)	-c	+1/9

koma (flat)	-fc	-1/9
bakiye (flat)	-fb	-4/9
küçük mücenneb (bemolle)	-fk	-5/9
büyük mücenneb (bemolle)	-fbm	-8/9

Per una spiegazione più generale della notazione musicale non occidentale, leggere Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

Vedi anche

Glossario musicale: Sezione “makam” in *Glossario Musicale*, Sezione “makamlar” in *Glossario Musicale*.

Guida alla notazione: Sezione 2.10.1 [Notazione comune per la musica non occidentale], pagina 485.

3 Input e output

Questa sezione tratta le questioni generali relative all'input e all'output di LilyPond, non specifiche di un certo tipo di notazione.

3.1 Struttura dell'input

Il principale formato di input di LilyPond sono i file di testo. Per convenzione, questi file hanno estensione `.ly`.

3.1.1 Struttura di una partitura

Un blocco `\score` deve contenere una singola espressione musicale delimitata da parentesi graffe:

```
\score {
  ...
}
```

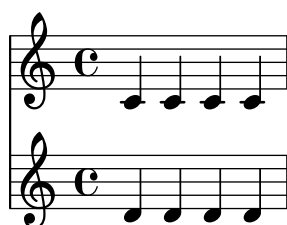
Nota: Ci deve essere **solo una** espressione musicale più esterna in un blocco `\score` e **deve** essere racchiusa tra parentesi graffe.

Questa espressione musicale singola può essere di qualsiasi dimensione e contenere altre espressioni musicali di qualsiasi complessità. Tutti gli esempi seguenti sono espressioni musicali:

```
{ c'4 c' c' c' }
{
  { c'4 c' c' c' }
  { d'4 d' d' d' }
}
```



```
<<
  \new Staff { c'4 c' c' c' }
  \new Staff { d'4 d' d' d' }
>>
```



```
{
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { \flauto }
      \new Staff { \oboe }
    >>
    \new StaffGroup <<
      \new Staff { \violinoI }
      \new Staff { \violinoII }
    >>
  >>
}
```

```

    >>
  >>
}

```

I commenti sono un’eccezione a questa regola generale (altre eccezioni sono spiegate in Sezione 3.1.5 [Struttura del file], pagina 497). Sia i commenti su una singola riga che quelli multirigo delimitati da `%{ ... %}` possono essere inseriti ovunque nel file di input: dentro o fuori un blocco `\score` e dentro o fuori la singola espressione musicale di un blocco `\score`.

È bene ricordare che anche se un file contiene soltanto un blocco `\score`, questo è implicitamente racchiuso in un blocco `\book`. Un blocco `\book` in un file sorgente produce almeno un file di output e il nome predefinito del file di output deriva dal nome del file di input, quindi `fandangoperelefanti.ly` genererà `fandangoperelefanti.pdf`.

Maggiori dettagli sui blocchi `\book` si trovano in Sezione 3.1.2 [Molteplici partiture in un libro], pagina 494, Sezione 3.1.3 [Molteplici file di output da un unico file di input], pagina 495, e Sezione 3.1.5 [Struttura del file], pagina 497.

Vedi anche

Manuale di apprendimento: Sezione “Lavorare sui file di input” in *Manuale di Apprendimento*, Sezione “Espressioni musicali” in *Manuale di Apprendimento*, Sezione “La partitura è una (singola) espressione musicale composta” in *Manuale di Apprendimento*.

3.1.2 Molteplici partiture in un libro

Un documento può contenere più brani di musica e testo, come, per esempio, uno studio o una parte orchestrale con vari movimenti. Ogni movimento si inserisce con un blocco `\score`,

```

\score {
  ...musica...
}

```

e il testo si inserisce con un blocco `\markup`,

```

\markup {
  ...testo...
}

```

Tutti i movimenti e i testi che appaiono nello stesso file `.ly` normalmente vengono elaborati in un singolo file di output.

```

\score {
  ...
}
\markup {
  ...
}
\score {
  ...
}

```

Un’importante eccezione è costituita dai documenti da elaborare con `lilypond-book`, dove occorre aggiungere esplicitamente un blocco `\book`, altrimenti apparirà nell’output solo il primo blocco `\score` o `\markup`.

L’istestazione di ogni brano musicale può essere inserita nel blocco `\score`. Il nome definito nel campo `piece` (brano) dell’istestazione apparirà all’inizio di ogni movimento. Il titolo dell’intero libro può trovarsi all’interno del blocco `\book` oppure, se questo non è presente, nel blocco `\header` all’inizio del file.

```

\header {

```

```

    title = "Otto miniature"
    composer = "Igor Stravinsky"
}
\score {
  \header { piece = "Romanza" }
  ...
}
\markup {
  ...testo della seconda strofa...
}
\markup {
  ...testo della terza strofa...
}
\score {
  \header { piece = "Minuetto" }
  ...
}

```

I brani musicali possono essere raggruppati in parti di libro tramite i blocchi `\bookpart`. Le parti di libro sono separate da un'interruzione di pagina e possono iniziare con un titolo, come il libro stesso, specificandolo in un blocco `\header`.

```

\bookpart {
  \header {
    title = "Titolo del libro"
    subtitle = "Prima parte"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Seconda parte"
  }
  \score { ... }
  ...
}

```

3.1.3 Molteplici file di output da un unico file di input

Per generare molteplici file di output dallo stesso file `.ly`, basta aggiungere molteplici blocchi `\book`, ognuno dei quali produrrà un file di output separato. Se non è specificato alcun blocco `\book` nel file di input, LilyPond tratterà implicitamente l'intero file come un singolo blocco `\book`, come è spiegato in Sezione 3.1.5 [Struttura del file], pagina 497.

Nel generare molteplici file da un singolo file sorgente, LilyPond controlla che nessuno dei file di output di alcun blocco `\book` sovrascriva il file di output prodotto da un blocco `\book` precedente dello stesso file di input.

Per farlo, aggiunge un suffisso al nome del file di output di ogni blocco `\book`, derivato dal nome del file di input (se viene lasciata l'impostazione predefinita).

Il comportamento predefinito consiste quindi nell'appendere un suffisso numerico a ogni nome che potrebbe entrare in conflitto, quindi

```

\book {
  \score { ... }
}

```

```

    \paper { ... }
  }
  \book {
    \score { ... }
    \paper { ... }
  }
  \book {
    \score { ... }
    \paper { ... }
  }

```

nel file sorgente `ottominiature.ly` genererà

- `ottominiature.pdf`,
- `ottominiature-1.pdf` e
- `ottominiature-2.pdf`.

3.1.4 Nomi dei file di output

LilyPond permette di decidere quali nomi di file debbano essere usati dai vari backend quando questi generano i file di output.

Nella sezione precedente abbiamo visto come LilyPond prevenga i conflitti di nome quando genera molti file di output da un singolo file sorgente. È possibile anche specificare un proprio suffisso per ogni blocco `\book`. Quindi si possono, per esempio, generare file chiamati `ottominiature-Romanza.pdf`, `ottominiature-Minuetto.pdf` e `ottominiature-Notturmo.pdf` aggiungendo una dichiarazione `\bookOutputSuffix` all'interno di ogni blocco `\book`.

```

\book {
  \bookOutputSuffix "Romanza"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Minuetto"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputSuffix "Notturmo"
  \score { ... }
  \paper { ... }
}

```

È possibile anche specificare un diverso nome del file di output per ciascun blocco `book`, tramite le dichiarazioni `\bookOutputName`

```

\book {
  \bookOutputName "Romanza"
  \score { ... }
  \paper { ... }
}
\book {
  \bookOutputName "Minuetto"
  \score { ... }
  \paper { ... }
}

```

```

}
\book {
  \bookOutputName "Notturmo"
  \score { ... }
  \paper { ... }
}

```

Questo file produrrà i seguenti file di output:

- Romanza.pdf,
- Minuetto.pdf e
- Notturmo.pdf.

3.1.5 Struttura del file

Un file .ly può contenere un qualsiasi numero di espressioni di livello superiore (in inglese, *toplevel expressions*). Per espressione di livello superiore si intende una delle seguenti:

- Una definizione di output, come `\paper`, `\midi` e `\layout`. Tale definizione, se posta nel livello superiore, cambia le impostazioni predefinite al livello del libro. Se più di una di queste definizioni viene inserita nel livello superiore, le definizioni vengono combinate, ma in caso di conflitto hanno precedenza le definizioni più recenti. Per sapere con precisione come ciò influisca sul blocco `\layout`, leggere Sezione 4.2.1 [Il blocco `\layout`], pagina 563.
- Un'espressione Scheme diretta, come `#{set-default-paper-size "a7" 'landscape}` o `#{ly:set-option 'point-and-click #f}`.
- Un blocco `\header`. Se all'inizio del file, imposta il blocco dell'intestazione globale. Questo è il blocco che contiene le impostazioni predefinite dei campi dei titoli come compositore, titolo, etc. per tutti i libri del file (vedi [Come funzionano i titoli], pagina 499).
- Un blocco `\score`. Questa partitura e altre eventuali partiture di livello superiore saranno combinate insieme in un singolo blocco `\book`. Tale comportamento può essere modificato impostando la variabile `toplevel-score-handler` nel livello superiore. Il gestore (in inglese *handler*) predefinito è definito nel file `../scm/lily-library.scm` e viene impostato nel file `../ly/declarations-init.ly`.
- Un blocco `\book` combina logicamente molteplici movimenti (ovvero molteplici blocchi `\score`) in un documento. Se ci sono vari blocchi `\score`, verrà creato un file di output per ogni blocco `\book`, in cui saranno concatenati tutti i movimenti corrispondenti. Ha senso specificare esplicitamente i blocchi `\book` in un file .ly solo se si desidera creare vari file di output da un solo file di input. Un'eccezione è data dai documenti `lilypond-book`, dove bisogna aggiungere esplicitamente un blocco `\book` se si vuole più di un singolo blocco `\score` o `\markup` nello stesso esempio. Tale comportamento può essere modificato impostando la variabile `toplevel-book-handler` nel livello superiore. Il gestore predefinito è definito nel file di inizializzazione `../scm/lily.scm`.
- Un blocco `\bookpart`. Un libro può essere suddiviso in varie parti, tramite blocchi `\bookpart`, per semplificare le interruzioni di pagina o per usare impostazioni `\paper` diverse nelle varie parti.
- Un'espressione musicale composta, come

```
{ c'4 d' e'2 }
```

pone il brano in un blocco `\score` e lo formatta in un unico libro insieme a tutti gli altri blocchi `\score` e espressioni musicali di livello superiore. In altre parole, un file che contiene soltanto l'espressione musicale precedente verrà trasformato in

```

\book {
  \score {
    \new Staff {

```

```

        \new Voice {
            { c'4 d' e'2 }
        }
    }
    \layout { }
}
\paper { }
\header { }
}

```

Tale comportamento può essere modificato impostando la variabile `toplevel-music-handler` nel livello superiore. Il gestore predefinito è definito nel file di inizializzazione `../scm/lily.scm`.

- Un testo, per esempio una strofa

```

\markup {
    2. La prima riga della seconda strofa.
}

```

I testi possono trovarsi sopra, sotto o in mezzo alle partiture o espressioni musicali, ovunque esse appaiano.

- Una variabile, come

```
foo = { c4 d e d }
```

può essere utilizzata in un punto successivo del file scrivendo `\foo`. Il nome di una variabile deve avere solo caratteri alfabetici; nessun numero, trattino o trattino basso.

L'esempio seguente mostra tre elementi che possono essere inseriti nel livello superiore

```

\layout {
    % Non giustificare l'output
    ragged-right = ##t
}

\header {
    title = "Do-re-mi"
}

{ c'4 d' e2 }

```

Ciascuna delle seguenti istruzioni lessicali può essere inserita in qualsiasi punto di un file:

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Un commento su riga singola, introdotto da un segno `%`.
- Un commento multirigo delimitato da `%{ ... %}`.

Lo spazio bianco tra gli elementi dell'input viene generalmente ignorato e può essere liberamente omesso o aumentato per migliorare la leggibilità. Tuttavia esistono dei casi in cui lo spazio bianco deve essere sempre usato per non incorrere in errori:

- Intorno ad ogni parentesi graffa di apertura e di chiusura.
- Dopo ogni comando o variabile, ovvero qualsiasi elemento che inizi con un segno `\`.
- Dopo ogni elemento che debba essere interpretato come un'espressione Scheme, ovvero ogni elemento che inizi con un segno `#`.

- Per separare tutti gli elementi di un'espressione Scheme.
- Nella modalità `lyricmode` prima e dopo i comandi `\set` e `\override`.

Vedi anche

Manuale di apprendimento: Sezione “Come funzionano i file di input di LilyPond” in *Manuale di Apprendimento*.

Guida alla notazione: [Come funzionano i titoli], pagina 499, Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

3.2 Titoli e intestazioni

Quasi tutte le partiture musicali hanno il titolo e il nome del compositore e alcuni brani comprendono molte altre informazioni.

3.2.1 Creazione di titoli intestazioni e piè di pagina

Come funzionano i titoli

Ogni blocco `\book` in un singolo file di input produce un diverso file di output, vedi Sezione 3.1.5 [Struttura del file], pagina 497. In ciascun file di output sono disponibili tre tipi di titolazioni: *titoli del libro* all'inizio di ogni libro (*book*), *titoli della parte* all'inizio di ciascuna parte (*bookpart*) e *titoli del brano* all'inizio di ciascun brano (*score*).

I valori dei campi dei titoli come `title` (titolo) e `composer` (compositore) sono definiti nei blocchi `\header` (la sintassi dei blocchi `\header` e un elenco completo dei campi disponibili si trovano in [Formattazione predefinita dei titoli delle parti e dei brani], pagina 502). I titoli del libro, delle parti e dei brani possono avere tutti gli stessi campi, sebbene per impostazione predefinita i campi dei titoli del brano siano limitati a *piece* e *opus*.

I blocchi `\header` possono essere inseriti in quattro diversi punti formando una gerarchia discendente di blocchi `\header`:

- All'inizio del file di input, prima di tutti i blocchi `\book`, `\bookpart` e `\score`.
- All'interno di un blocco `\book` ma fuori da tutti i blocchi `\bookpart` e `\score` compresi in quel libro.
- All'interno di un blocco `\bookpart` ma fuori da tutti i blocchi `\score` compresi in quella parte.
- In un blocco `\score`.

I valori dei campi vengono filtrati attraverso questa gerarchia, con i valori più in alto nella gerarchia che persistono finché un valore più in basso nella gerarchia non ha la precedenza. In sintesi:

- Il titolo di un libro deriva dai campi impostati all'inizio del file di input, modificati dai campi definiti nel blocco `\book`. I valori risultanti vengono usati per stampare il titolo di quel libro, purché ci sia altro materiale che generi una pagina all'inizio del libro, prima della prima parte. Una singola interruzione di pagina (`\pageBreak`) è sufficiente.
- Il titolo di una parte deriva dai campi impostati all'inizio del file di input, modificati dai campi definiti nel blocco `\book` e poi da quelli definiti nel blocco `\bookpart`. I valori risultanti vengono usati per stampare il titolo di quella parte del libro.
- Il titolo di un brano deriva dai campi all'inizio del file di input, modificati dai campi definiti nel blocco `\book` e poi da quelli definiti nel blocco `\bookpart` e infine da quelli definiti nel blocco `\score`. I valori risultanti vengono usati per stampare il titolo di quel brano. Nota bene: per impostazione predefinita nei titoli del brano appaiono soltanto i campi *piece* e *opus* a meno che la variabile `print-all-headers` di `\paper` non sia impostata su `#t`.

Non è necessario inserire blocchi `\header` in tutti e quattro i luoghi: alcuni o perfino tutti possono essere omessi. Analogamente, in semplici file di input si possono omettere i blocchi `\book` e `\bookpart`, lasciando che questi siano creati implicitamente.

Se il libro ha un solo brano, il blocco `\header` viene di solito messo all'inizio del file, in modo che sia prodotto soltanto il titolo della parte e lasciando disponibili tutti i campi di titolazione.

Se il libro ha vari brani, è possibile usare diverse disposizioni dei blocchi `\header`, corrispondenti ai diversi tipi di pubblicazione musicale. Per esempio, se la pubblicazione contiene vari brani dello stesso compositore, la soluzione più adatta prevederebbe un blocco `\header` all'inizio del file che specifichi il titolo del libro e il nome del compositore, e dei blocchi `\header` in ciascun blocco `\score` che specifichino il titolo del brano (*piece*) e dell'opera (*opus*), come in questo esempio:

```
\header {
  title = "SUITE I."
  composer = "J. S. Bach."
}

\score {
  \header {
    piece = "Prélude."
  }
  \new Staff \relative {
    \clef bass
    \key g \major
    \repeat unfold 2 { g,16( d' b') a b d, b' d, } |
    \repeat unfold 2 { g,16( e' c') b c e, c' e, } |
  }
}

\score {
  \header {
    piece = "Allemande."
  }
  \new Staff \relative {
    \clef bass
    \key g \major
    \partial 16 b16 |
    <g, d' b'~>4 b'16 a( g fis) g( d e fis) g( a b c) |
    d16( b g fis) g( e d c) b(c d e) fis( g a b) |
  }
}
```

SUITE I.

J. S. Bach.

Prélude.



Allemande.



Sono possibili disposizioni più complesse. Per esempio, i campi testuali del blocco `\header` di un libro possono essere stampati nei titoli di tutti i brani, magari sovrascrivendo alcuni campi e sopprimendone altri:

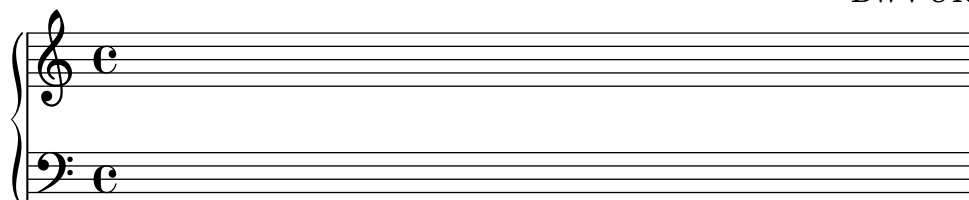
```
\book {
  \paper {
    print-all-headers = ##t
  }
  \header {
    title = "DAS WOHLTEMPERIRTE CLAVIER"
    subtitle = "TEIL I"
    % Non mostrare il piè di pagina predefinito nell'ultima pagina di questo libro
    tagline = ##f
  }
  \markup { \vspace #1 }
  \score {
    \header {
      title = "PRAELUDIUM I"
      opus = "BWV 846"
      % Non mostrare il sottotitolo in questo brano
      subtitle = ##f
    }
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
  }
  \score {
    \header {
      title = "FUGA I"
      subsubtitle = "A 4 VOCI"
      opus = "BWV 846"
      % Non mostrare il sottotitolo in questo brano
      subtitle = ##f
    }
    \new PianoStaff <<
      \new Staff { s1 }
      \new Staff { \clef "bass" s1 }
    >>
  }
}
```

DAS WOHLTEMPERIRTE CLAVIER

TEIL I

PRAELUDIUM I

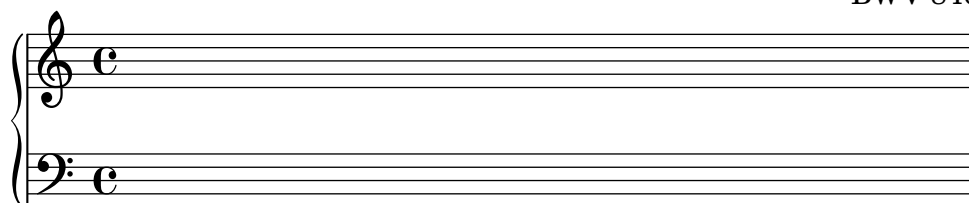
BWV 846



FUGA I

A 4 VOCI

BWV 846



Vedi anche

Guida alla notazione: Sezione 3.1.5 [Struttura del file], pagina 497, [Formattazione predefinita dei titoli delle parti e dei brani], pagina 502, [Formattazione personalizzata dei titoli], pagina 507.

Formattazione predefinita dei titoli delle parti e dei brani

Questo esempio illustra visivamente tutte le variabili del blocco `\header`:

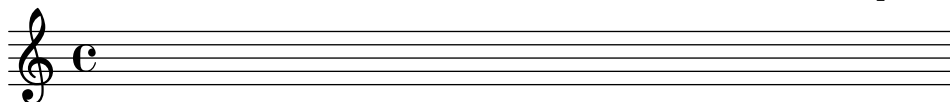
```
\book {
  \header {
    % I seguenti campi sono centrati
    dedication = "Dedica"
    title = "Titolo"
    subtitle = "Sottotitolo"
    subsubtitle = "Sottosottotitolo"
    % I seguenti campi sono distribuiti uniformemente su una riga
    % il campo "instrument" appare anche nelle pagine seguenti
    instrument = \markup \with-color #green "Strumento"
    poet = "Poeta"
    composer = "Compositore"
    % I campi seguenti sono posti agli estremi opposti della stessa riga
    meter = "Tempo"
    arranger = "Arrangiatore"
    % I campi seguenti sono centrati in fondo
    tagline = "Lo slogan va in fondo all'ultima pagina"
```

```

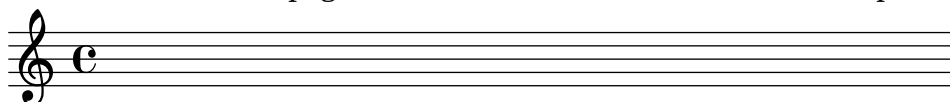
    copyright = "Il copyright va in fondo alla prima pagina"
}
\score {
  \header {
    % I campi seguenti sono posti agli estremi opposti della stessa riga
    piece = "Brano 1"
    opus = "Opera 1"
  }
  { s1 }
}
\score {
  \header {
    % I campi seguenti sono posti agli estremi opposti della stessa riga
    piece = "Brano 2 sulla stessa pagina"
    opus = "Opera 2"
  }
  { s1 }
}
\pageBreak
\score {
  \header {
    % I campi seguenti sono posti agli estremi opposti della stessa riga
    piece = "Brano 3 su una nuova pagina"
    opus = "Opera 3"
  }
  { s1 }
}
}

```

	Dedica	
	Titolo	
	Sottotitolo	
	Sottosottotitolo	
Poeta	Strumento	Compositore
Tempo		Arrangiatore
Brano 1		Opera 1



Brano 2 sulla stessa pagina	Opera 2
-----------------------------	---------



Il copyright va in fondo alla prima pagina

2	Strumento	
	Brano 3 su una nuova pagina	Opera 3
		

Lo slogan va in fondo all'ultima pagina

Notare che:

- Il nome dello strumento sarà ripetuto su ogni pagina.
- Appaiono soltanto i campi `piece` e `opus` di un blocco `\score` quando la variabile `print-all-headers` del foglio è impostata su `##f` (valore predefinito).
- I campi testuali non specificati in un blocco `\header` sono sostituiti con `\null` in modo da non sprecare spazio.
- Le impostazioni predefinite per `scoreTitleMarkup` posizionano i campi `piece` e `opus` alle estremità opposte della stessa riga.

Per cambiare la formattazione predefinita leggere [Formattazione personalizzata dei titoli], pagina 507.

Se un blocco `\book` inizia subito con un blocco `\bookpart`, non verrà stampato alcun titolo per il libro, dato che non esiste una pagina in cui farlo apparire. Se il titolo del libro deve comunque apparire, conviene inserire all'inizio di `\book` del testo inserito con `\markup` oppure un comando `\pageBreak`.

Usare la variabile `breakbefore` all'interno di un blocco `\header` racchiuso in un blocco `\score` per far sì che i titoli del blocco `\header` di più alto livello appaiano da soli nella prima pagina, mentre la musica (definita nel blocco `\score`) inizia nella pagina successiva.

```
\book {
  \header {
    title = "Titolo del libro"
    subtitle = "Sottotitolo del libro"
    copyright = "Fine della prima pagina"
  }
  \score {
    \header {
      piece = "Titolo del brano"
      breakbefore = ##t
    }
  }
}
```

```
\repeat unfold 4 { e'' e'' e'' e'' }
}
```

Titolo del libro

Sottotitolo del libro

Fine della prima pagina

2

Titolo del brano



LilyPond v2.25.35

Vedi anche

Manuale di apprendimento: Sezione “Come funzionano i file di input di LilyPond” in *Manuale di Apprendimento*,

Guida alla notazione: [Formattazione personalizzata dei titoli], pagina 507, Sezione 3.1.5 [Struttura del file], pagina 497.

File installati: `ly/titling-init.ly`.

Formattazione predefinita delle intestazioni e dei piè di pagina

Le *intestazioni* e i *piè di pagina* sono linee di testo che appaiono in cima e in fondo alle pagine, distinte dal testo principale di un libro. Possono essere definite nelle seguenti variabili del blocco `\paper`:

- `oddHeaderMarkup`
- `evenHeaderMarkup`
- `oddFooterMarkup`
- `evenFooterMarkup`

Queste variabili di markup possono soltanto accedere ai campi testuali definiti nei blocchi `\header` del livello superiore (che vengono applicati a tutti i brani del libro) e sono definiti in `ly/titling-init.ly`. Impostazioni predefinite:

- i numeri di pagina sono collocati automaticamente in alto a sinistra (se la pagina è pari) o in alto a destra (se la pagina è dispari), a partire dalla seconda pagina.
- il campo `instrument` viene ripetuto al centro di ogni pagina, a partire dalla seconda pagina.
- il testo del copyright è centrato in fondo alla prima pagina.
- lo “slogan” (o firma) – `tagline` – è centrato in fondo all’ultima pagina o sotto il campo del copyright se c’è una sola pagina.

Il testo del piè di pagina predefinito per l’ultima pagina può essere modificato aggiungendo il campo `tagline` al blocco `\header` del livello superiore.

```
\book {
  \header {
    tagline = "... notazione musicale per tutti"
  }
  \score {
    \relative {
      c'4 d e f
    }
  }
}
```



... notazione musicale per tutti

Per toglierlo, impostare `tagline` su `##f`.

3.2.2 Titoli intestazioni e piè di pagina personalizzati

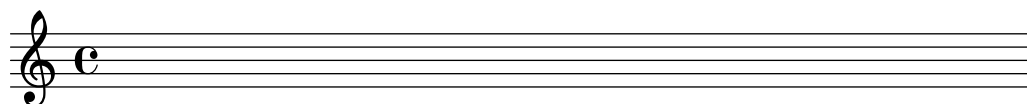
Titoli personalizzati

Si possono usare i normali comandi `\markup` per personalizzare qualsiasi intestazione, piè di pagina e titolo di un blocco `\header`.

```
\score {
  \header {
    piece = \markup { \fontsize #4 \bold "PRAELUDIUM I" }
    opus = \markup { \italic "BWV 846" }
  }
  { s1 }
}
```

PRAELUDIUM I

BWV 846



Vedi anche

Guida alla notazione: Sezione 1.8.2 [Formattazione del testo], pagina 260.

Formattazione personalizzata dei titoli

I comandi `\markup` nel blocco `\header` sono utili solo per la formattazione del testo, ma non consentono un controllo preciso sul posizionamento dei titoli. Per personalizzare il posizionamento dei campi testuali, cambiare una o entrambe le seguenti variabili `\paper`:

- `bookTitleMarkup`
- `scoreTitleMarkup`

Il posizionamento dei titoli secondo i valori predefiniti di queste variabili `\markup` è mostrato negli esempi in [Formattazione predefinita dei titoli delle parti e dei brani], pagina 502.

Le impostazioni predefinite di `scoreTitleMarkup`, definite in `ly/titling-init.ly`, sono:

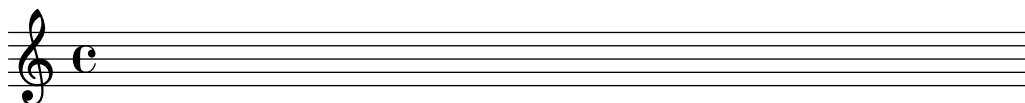
```
scoreTitleMarkup = \markup { \column {
  \if \should-print-all-headers { \bookTitleMarkup \hspace #1 }
  \fill-line {
    \fromproperty #'header:piece
    \fromproperty #'header:opus
  }
}
```

Questo pone i campi testuali `piece` e `opus` alle estremità opposte della stessa riga:

```
\score {
  \header {
    piece = "PRAELUDIUM I"
    opus = "BWV 846"
  }
  { s1 }
```

PRAELUDIUM I

BWV 846



L'esempio seguente ridefinisce `scoreTitleMarkup` in modo che il campo testuale di `piece` sia centrato e in un tipo di carattere più grande e in grassetto.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:opus
      }
    }
  }
  \header { tagline = ##f }
  \score {
```



```

\header {
  piece = "PRAELUDIUM I"
  opus = "BWV 846"
}
{ s1 }
}

```



I campi testuali che non sono normalmente attivi nei blocchi `\header` compresi in un blocco `\score` possono essere stampati nello spazio dedicato al titolo del brano aggiungendo `print-all-headers` nel blocco `\paper`. Lo svantaggio di questo metodo è che i campi testuali intesi esclusivamente per lo spazio del titolo della parte devono essere manualmente soppressi in ogni blocco `\score`. Vedi [Come funzionano i titoli], pagina 499.

Per evitare ciò, è meglio mettere il campo testuale desiderato nella definizione di `scoreTitleMarkup`. Nell'esempio seguente il campo `composer` (solitamente associato a `bookTitleMarkup`) viene aggiunto a `scoreTitleMarkup`, facendo sì che ogni brano possa elencare un diverso compositore:

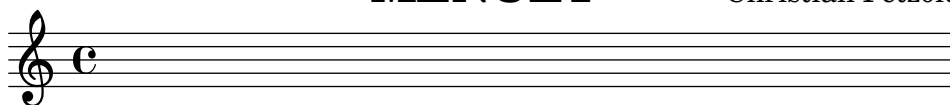
```

\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \fontsize #4 \bold \fromproperty #'header:piece
        \fromproperty #'header:composer
      }
    }
  }
}
\header { tagline = ##f }
\score {
  \header {
    piece = "MENUET"
    composer = "Christian Petzold"
  }
  { s1 }
}
\score {
  \header {
    piece = "RONDEAU"
    composer = "François Couperin"
  }
  { s1 }
}
}

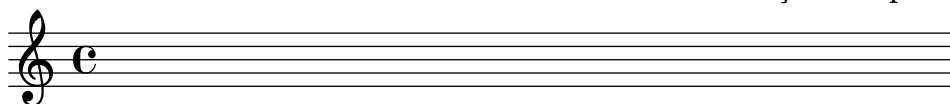
```

MENUET

Christian Petzold

**RONDEAU**

François Couperin



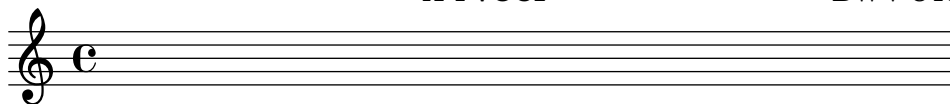
È anche possibile creare un campo testuale personalizzato e fare riferimento ad esso nella definizione di markup.

```
\book {
  \paper {
    indent = 0\mm
    scoreTitleMarkup = \markup {
      \fill-line {
        \null
        \override #`(direction . ,UP)
        \dir-column {
          \center-align \fontsize #-1 \bold
          \fromproperty #'header:mycustomtext %% Campo definito dall'utente
          \center-align \fontsize #4 \bold
          \fromproperty #'header:piece
        }
        \fromproperty #'header:opus
      }
    }
  }
}
\header { tagline = ##f }
\score {
  \header {
    piece = "FUGA I"
    mycustomtext = "A 4 VOCI" %% Campo definito dall'utente
    opus = "BWV 846"
  }
  { s1 }
}
```

FUGA I

A 4 VOCI

BWV 846



Vedi anche

Guida alla notazione: [Come funzionano i titoli], pagina 499.

Formattazione personalizzata di intestazioni e piè di pagina

I comandi `\markup` nel blocco `\header` sono utili solo per la formattazione del testo, ma non consentono un controllo preciso sul posizionamento di intestazioni e piè di pagina. Per personalizzare il posizionamento dei campi testuali, usare una o entrambe le seguenti variabili `\paper`:

- `oddHeaderMarkup`
- `evenHeaderMarkup`
- `oddFooterMarkup`
- `evenFooterMarkup`

Il comando `\on-the-fly` – usato all'interno di un blocco `\markup` – permette di aggiungere del testo a intestazioni e piè di pagina definiti nel blocco `\paper`, solo se certe condizioni sono soddisfatte, tramite la seguente sintassi:

```
variabile = \markup {
  ...
  \on-the-fly \procedura testo
  ...
}
```

La *procedura* viene chiamata ogni volta che viene elaborato il comando `\markup` nel quale essa si trova. La *procedura* verifica una precisa condizione e interpreta (ovvero stampa) l'argomento *testo* se e solo se la condizione è vera.

Sono disponibili varie procedure pronte per verificare varie condizioni:

Nome della procedura	Condizione verificata
<code>print-page-number-check-first</code>	stampare questa pagina?
<code>create-page-number-stencil</code>	<code>print-page-numbers</code> è vero?
<code>print-all-headers</code>	<code>print-all-headers</code> è vero?
<code>first-page</code>	prima pagina del libro?
<code>not-first-page</code>	non la prima pagina del libro?
<code>(on-page nmbr)</code>	numero pagina = <code>nmbr</code> ?
<code>last-page</code>	ultima pagina del libro?
<code>part-first-page</code>	prima pagina della parte?
<code>not-part-first-page</code>	non la prima pagina della parte?
<code>part-last-page</code>	ultima pagina della parte?
<code>not-single-page</code>	pagine della parte > 1?

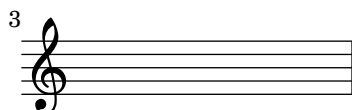
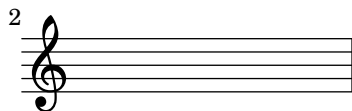
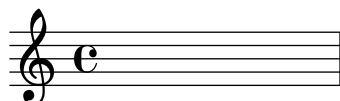
L'esempio seguente centra i numeri di pagina in fondo a ogni pagina. Innanzitutto vengono annullate le impostazioni predefinite per `oddHeaderMarkup` e `evenHeaderMarkup` definendo ciascuno di essi come un markup *null*. Poi `oddFooterMarkup` viene ridefinito col numero di pagina centrato. Infine a `evenFooterMarkup` viene assegnata la stessa formattazione di `\oddFooterMarkup`:

```
\book {
  \paper {
    print-page-number = ##t
    print-first-page-number = ##t
    oddHeaderMarkup = \markup \null
    evenHeaderMarkup = \markup \null
    oddFooterMarkup = \markup {
      \fill-line {
        \if \should-print-page-number
        \fromproperty #'page:page-number-string
```

```

    }
  }
  evenFooterMarkup = \oddFooterMarkup
}
\score {
  \new Staff { s1 \break s1 \break s1 }
}
}

```



1

Varie condizioni `\on-the-fly` possono essere combinate insieme come se si utilizzasse l'operatore logico 'AND' e il testo apparirà solo se tutte le condizioni sono vere. Per esempio, queste due condizioni

```

\if \on-first-page
\if \on-last-page
{ \markup ... \fromproperty #'header: ... }

```

verificano se l'output è una pagina singola.

Vedi anche

Guida alla notazione: [Come funzionano i titoli], pagina 499, [Formattazione predefinita dei titoli delle parti e dei brani], pagina 502.

File installati: `../ly/titling-init.ly`.

3.2.3 Creazione di metadati per i file di output

Oltre ad apparire nell'output, le variabili di `\header` vengono usate anche per impostare i metadati per i file di output. Per esempio, con i file PDF, questi metadati possono essere mostrati dai lettori PDF come proprietà del file PDF. Per ciascun tipo di file di output, verranno considerate solo le definizioni `\header` dei blocchi specifici per quel tipo e dei blocchi che si trovano al livello superiore della gerarchia. Dunque per i file PDF solo le definizioni `\header` del livello `\book` e del livello superiore vengono utilizzate per i metadati PDF del documento, mentre per i file MIDI vengono usate le definizioni che si trovano in un livello inferiore o in un blocco `\score`.

Per esempio, impostando la proprietà `title` del blocco `header` su 'Sinfonia I', questo nome verrà usato come titolo del documento PDF e come nome della sequenza del file MIDI.

```

\header {
  title = "Sinfonia I"
}

```

```
}
```

Se si desidera impostare il titolo dell'output su un valore e la proprietà titolo del PDF su un valore diverso, si può usare `pdftitle`, nel modo seguente:

```
\header {
  title = "Sinfonia I"
  pdftitle = "Sinfonia I di Beethoven"
}
```

Le variabili `title`, `subject`, `keywords`, `subtitle`, `composer`, `arranger`, `poet`, `author` e `copyright` impostano anche le proprietà del PDF e possono essere tutte prefissate con 'pdf' per impostare una proprietà del PDF su un valore diverso da quello dell'output.

La proprietà PDF Creator è automaticamente impostata su 'LilyPond' più la versione di LilyPond utilizzata, e `CreationDate` (data di creazione) e `ModDate` (data di modifica) sono entrambe impostate sulla data e ora correnti. `ModDate` può essere sovrascritta impostando nel blocco header la variabile `moddate` (o `pdfmoddate`) su una valida data per il PDF.

La variabile `title` imposta anche il nome della sequenza dei file MIDI. La variabile `midititle` serve a impostare il nome della sequenza in modo indipendente dal valore usato per l'output.

3.2.4 Creazione di note a piè di pagina

Le note a piè di pagina possono essere usate in situazioni diverse. In tutti i casi, un 'segno della nota a piè di pagina' viene inserito come riferimento vicino al testo o alla musica a cui si riferisce e il corrispondente 'testo della nota a piè di pagina' appare in fondo alla stessa pagina.

Le note a piè di pagina si creano diversamente a seconda che siano applicate a espressioni musicali o a del testo separato e fuori dalle espressioni musicali.

Note a piè di pagina nelle espressioni musicali

Panoramica sulle note a piè di pagina attaccate alla musica

Le note a piè di pagina nelle espressioni musicali appartengono a due categorie:

Note a piè di pagina basate su un evento

sono collegate a un preciso evento. Esempi di tali eventi sono note singole, articolazioni (come le indicazioni di diteggiatura, gli accenti, le dinamiche), e ciò che è successivo a un evento (come le legature di portamento e le travature manuali). La forma generale per le note a piè di pagina basate su un evento è la seguente:

```
[direzione] \footnote [segno] offset nota musica
```

Note a piè di pagina basate sul tempo

sono collegate a un preciso momento temporale in un contesto musicale. Alcuni comandi come `\time` e `\clef` non usano in realtà degli eventi per creare oggetti come le indicazioni di tempo e le chiavi. E nemmeno un accordo crea un suo evento: il suo gambo o coda sono creati alla fine di un'unità di tempo (attraverso un evento di una delle note al suo interno). Non è definito esattamente quale dei molteplici eventi nota di un accordo sarà giudicato la causa ultima di un gambo o di una coda. Dunque per legare una nota a questi, sono preferibili le note a piè di pagina basate sul tempo.

Una nota a piè di pagina basata sul tempo permette a tali oggetti della formattazione di avere delle note senza che sia necessario riferirsi a un evento. La forma generale per le note a piè di pagina basate sul tempo è la seguente:

```
\footnote [segno] offset nota [Contesto].NomeGrob
```

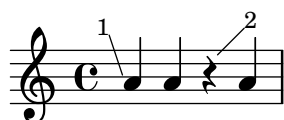
Gli elementi di entrambe le forme sono:

- direzione* se (e solo se) `\footnote` viene applicato a un post-evento o a una articolazione, deve essere preceduto da un indicatore di direzione (`-`, `_`, `^`) per poter collegare la *musica* (col segno della nota a piè di pagina) alla nota o pausa precedenti.
- segno* è del testo – racchiuso tra virgolette o in un blocco `\markup` – che specifica il segno della nota usata per contrassegnare sia il punto di riferimento che la nota stessa in fondo alla pagina. Può essere omesso (o sostituito con `\default`), nel qual caso sarà generato automaticamente un numero in sequenza. Tali sequenze numeriche ricominciano su ogni pagina contenente una nota.
- offset* è una coppia di numeri come `'#'(2 . 1)` che specificano la distanza orizzontale e verticale (X e Y offset), in unità di spazio rigo, dal bordo dell'oggetto in cui il segno deve essere posizionato. Valori positivi degli offset vengono presi dal bordo in alto a destra, valori negativi dal bordo in basso a sinistra e lo zero significa che il segno è centrato sul bordo.
- Contesto* è il contesto in cui è creato il grob a cui è collegata la nota. Può essere omesso se il grob si trova in un contesto più basso, per esempio un contesto *Voice*.
- NomeGrob* indica un tipo di grob a cui assegnare la nota (come `'Flag'`). Se viene specificato, la nota a piè di pagina non è collegata a un'espressione musicale in particolare, bensì a tutti i grob di quel tipo che si trovano in quel momento del tempo musicale.
- nota* è il testo – racchiuso tra virgolette o in un blocco `\markup` – che contiene il testo da usare per la nota a piè di pagina.
- musica* è l'evento musicale o il post-evento o articolazione a cui viene collegata la nota.

Note a piè di pagina basate su un evento

Una nota a piè di pagina può essere collegata a un oggetto della formattazione direttamente causato dall'evento corrispondente a *musica* con la sintassi:

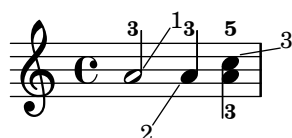
```
\footnote [segno] offset nota musica
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(-1 . 3) "Una nota" a4
    a4
    \footnote #'(2 . 2) "Una pausa" r4
    a4
  }
}
```



¹Una nota
²Una pausa

Contrassegnare un *intero* accordo con una nota a piè di pagina basata su un evento non è possibile: infatti un accordo, perfino uno che contenga una sola nota, non produce un vero evento specifico. Tuttavia possono essere contrassegnate singole note *dentro* l'accordo:

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    \footnote #'(2 . 3) "Non funziona" <a-3>2
    <\footnote #'(-2 . -3) "Funziona" a-3>4
    <a-3 \footnote #'(3 . 1/2) "Anche questo funziona" c-5>4
  }
}
```



¹Non funziona
²Funziona
³Anche questo funziona

Se la nota deve essere attaccata a un post-evento o un'articolazione il comando `\footnote` *deve* essere preceduto da un indicatore di direzione, `-`, `_`, `^`, e seguito dal post-evento o dall'articolazione che si desidera commentare nell'argomento *musica*. In questa forma `\footnote` può essere considerato semplicemente una copia del suo ultimo argomento con un segno di nota a piè di pagina collegato a esso. La sintassi è:

```
direzione \footnote [segno] offset nota musica
```

```
\book {
  \header { tagline = ##f }
  \relative {
    a'4_ \footnote #'(0 . -1) "Una legatura di portamento forzata in giù" (
    b8^ \footnote #'(1 . 0.5) "Una travatura manuale forzata in su" [
    b8 ]
    c4 )
    c- \footnote #'(1 . 1) "Tenuto" --
  }
}
```



¹Una legatura di portamento forzata in giù
²Una travatura manuale forzata in su
³Tenuto

Note a piè di pagina basate sul tempo

Se l'oggetto della formattazione a cui attaccare la nota è *indirettamente* causato da un evento – come un Accidental (alterazione) o Stem (gambo) causati da un evento NoteHead (testa di nota), è necessario specificare il NomeGrob dell'oggetto di formattazione al posto di *musica* dopo il testo della nota:

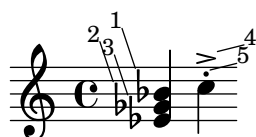
```
\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote #'(-1 . -3) "Un bemolle" Accidental
    aes4 c
    \footnote #'(-1 . 0.5) "Un altro bemolle" Accidental
    ees
    \footnote #'(1 . -2) "Un gambo" Stem
    aes
  }
}
```



-
- ¹Un bemolle
 - ²Un altro bemolle
 - ³Un gambo

Tuttavia nota che, quando si specifica un NomeGrob, una nota a piè di pagina sarà attaccata a tutti i grob di quel tipo che si trovano in quel momento musicale:

```
\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote #'(-1 . 3) "Un bemolle" Accidental
    <ees ges bes>4
    \footnote #'(2 . 0.5) "Articolazione" Script
    c'->- .
  }
}
```



-
- ¹Un bemolle
 - ²Un bemolle
 - ³Un bemolle
 - ⁴Articolazione
 - ⁵Articolazione

È possibile assegnare a una nota di un accordo una singola nota a piè di pagina (basata su un evento). ‘NoteHead’ è l’unico grob causato direttamente dalla nota di un accordo, dunque una nota a piè di pagina basata su un evento è adatta *soltanto* ad aggiungere una nota a piè di pagina al ‘NoteHead’ all’interno di un accordo. Tutti gli altri grob delle note di un accordo sono causati indirettamente. Il comando \footnote non ha una sintassi per specificare *sia* un particolare tipo di grob *sia* un particolare evento a cui collegare la nota. Tuttavia si può usare un comando \footnote basato sul tempo per specificare il tipo di grob e poi precedere tale comando con \single perché venga applicato soltanto all’evento che segue:

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    < \footnote #'(1 . -2) "An A" a
      \single \footnote #'(-1 . -1) "Un diesis" Accidental
      cis
      \single \footnote #'(0.5 . 0.5) "Un bemolle" Accidental
      ees fis
    >2
  }
}
```



¹Un bemolle
²Un diesis
³An A

Nota: Quando le note a piè di pagina sono collegate a diversi elementi musicali nello stesso momento musicale, come nell’esempio precedente, le note sono numerate dall’elemento più alto a quello più in basso come questi appaiono nell’output e non nell’ordine in cui sono inseriti nell’input.

Gli oggetti della formattazione come le chiavi e i cambi di armatura di chiave sono causati principalmente da proprietà modificate piuttosto che da veri eventi. Per questo motivo le note su tali oggetti devono essere basate sul loro tempo musicale. Le note a piè di pagina basate sul tempo sono da preferire anche quando si vogliono contrassegnare elementi come i gambi e le travature in un *accordo*: sebbene tali elementi dell’accordo siano nominalmente assegnati a *un* evento all’interno dell’accordo, affidarsi a una scelta particolare sarebbe imprudente.

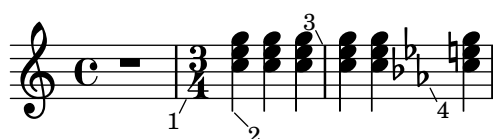
L’oggetto della formattazione in questione deve essere sempre specificato esplicitamente nelle note a piè di pagina basate sul tempo, e il contesto appropriato deve essere indicato se il grob viene creato in un contesto diverso da quello più basso.

```
\book {
  \header { tagline = ##f }
  \relative c'' {
    r1 |
```

```

\footnote #'(-0.5 . -1) "Cambio di tempo" Staff.TimeSignature
\time 3/4
\footnote #'(1 . -1) "Gambo dell'accordo" Stem
<c e g>4 q q
\footnote #'(-0.5 . 1) "Stanghetta" Staff.BarLine
q q
\footnote #'(0.5 . -1) "Cambio di armatura" Staff.KeySignature
\key c \minor
q
}
}

```



-
- ¹Cambio di tempo
 - ²Gambo dell'accordo
 - ³Stanghetta
 - ⁴Cambio di armatura

Si possono usare segni personalizzati in alternativa a quelli numerici, e si può sopprimere la linea che collega l'oggetto commentato al segno:

```

\book {
  \header { tagline = ##f }
  \relative c' {
    \footnote "*" #'(0.5 . -2) \markup { \italic "*" La prima nota" } a'4
    b8
    \footnote \markup { \super "$" } #'(0.5 . 1)
    \markup { \super "$" \italic " La seconda nota" } e
    c4
    \once \override Score.Footnote.annotation-line = ##f
    b-\footnote \markup \tiny "+" #'(0.1 . 0.1)
    \markup { \super "+" \italic " Editoriale" } \p
  }
}

```



-
- * *La prima nota*
 - \$ *La seconda nota*
 - + *Editoriale*

Altri esempi di segni personalizzati si trovano in [Note a piè di pagina nel testo separato], pagina 518.

Note a piè di pagina nel testo separato

Vengono usate all'interno di blocchi `\markup` che si trovano fuori dalle espressioni musicali. Non hanno una linea che le unisce al loro punto di riferimento: i loro segni seguono semplicemente il testo citato. I segni possono essere inseriti automaticamente, nel qual caso sono numerici; altrimenti è possibile inserire manualmente dei segni personalizzati.

Le note a piè di pagina su testo separato vengono create in modo diverso a seconda che si scelgano segni automatici oppure segni personalizzati.

Note a piè di pagina nel testo separato con segni automatici

La sintassi di una nota a piè di pagina nel testo separato con segni automatici è

```
\markup { ... \footnote testo nota ... }
```

Gli elementi sono:

testo è il testo – racchiuso tra virgolette doppie o in un blocco markup – da contrassegnare.

nota è il testo della nota a piè di pagina.

Per esempio:

```
\book {
  \header { tagline = ##f }
  \markup {
    "Una semplice"
    \footnote "canzone" \italic " Scritta da me"
    "è mostrata sotto. È una composizione"
    \footnote "recente" \italic " Agosto 2012"
    "."
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

Una semplice canzone è mostrata sotto. È una composizione recente .



Scritta da me
Agosto 2012

Note a piè di pagina nel testo separato con segni personalizzati

La sintassi di una nota a piè di pagina nel testo separato con segni personalizzati è

```
\markup { ... \footnote segno nota ... }
```

Gli elementi sono:

- segno* è una stringa di testo o un markup che indicano il segno da usare per contrassegnare il punto di riferimento. Tale segno *non* viene inserito automaticamente prima della nota stessa.
- nota* è una stringa di testo o un markup che indicano il testo della nota a piè di pagina, preceduto dal *segno*.

Qualsiasi carattere facile da scrivere, come * o +, può essere usato come segno, come è spiegato in [Note a piè di pagina nelle espressioni musicali], pagina 512. Altrimenti, si possono usare gli alias ASCII (vedi [Alias ASCII], pagina 533):

```
\book {
  \paper { #(include-special-characters) }
  \header { tagline = ##f }
  \markup {
    "Una semplice canzone"
    \footnote "*" \italic "* Scritta da me"
    "è mostrata sotto. È una composizione recente"
    \footnote \super &dagger; \concat {
      \super &dagger; \italic " Agosto 2012"
    }
    "."
  }
  \relative {
    a'4 b8 e c4 d
  }
}
```

Una semplice canzone * è mostrata sotto. È una composizione recente †.



* *Scritta da me*

† *Agosto 2012*

È possibile usare anche i caratteri Unicode per indicare i segni (vedi [Unicode], pagina 532):

```
\book {
  \header { tagline = ##f }
  \markup {
    "Una semplice canzone"
    \footnote \super \char##x00a7 \concat {
      \super \char##x00a7 \italic " Scritta da me"
    }
    "è mostrata sotto. È una composizione recente"
```

```

\footnote \super \char##x00b6 \concat {
  \super \char##x00b6 \italic " Agosto 2012"
}
" ."
}
\relative {
  a'4 b8 e c4 d
}
}

```

Una semplice canzone § è mostrata sotto. È una composizione recente ¶.



§ *Scritta da me*
 ¶ *Agosto 2012*

Vedi anche

Manuale di apprendimento: Sezione “Oggetti e interfacce” in *Manuale di Apprendimento*.

Guida alla notazione: [Alias ASCII], pagina 533, [Nuvoletta di aiuto], pagina 247, Sezione A.14 [Elenco dei caratteri speciali], pagina 792, [Indicazioni testuali], pagina 256, [Scritte], pagina 253, [Unicode], pagina 532.

Guida al funzionamento interno: Sezione “FootnoteEvent” in *Guida al Funzionamento Interno*, Sezione “Footnote” in *Guida al Funzionamento Interno*, Sezione “Footnote_engraver” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Varie note a piè di pagina sulla stessa pagina possono essere messe soltanto una dopo l'altra; non è possibile disporle sulla stessa riga.

Le note a piè di pagina non possono essere collegate a `MultiMeasureRests`, a travature automatiche o a testo vocale.

I segni delle note a piè di pagina potrebbero entrare in collisione con righe, oggetti `\markup`, altri segni e linee delle note a piè di pagina.

3.2.5 Riferimento ai numeri di pagina

È possibile contrassegnare un punto specifico di una partitura usando il comando `\label` (etichetta) nel livello superiore o all'interno della musica. Questa etichetta può quindi essere citata all'interno di un blocco markup, per ottenere il numero di pagina in cui è stato inserito il punto contrassegnato, tramite il comando markup `\page-ref`.

```

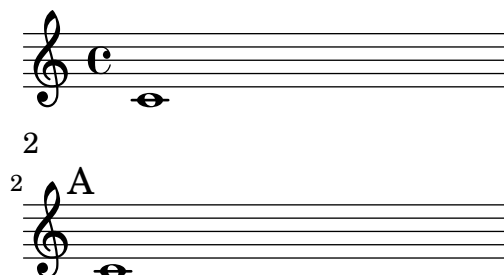
\header { tagline = ##f }
\book {
  \label #'primoBranco

```

```

\score {
  {
    c'1
    \pageBreak \mark A \label #'segnoA
    c'1
  }
}
\markup { Il primo brano inizia a pagina \page-ref #'primoBrano "0" "?" }
\markup { Il segno A è a pagina \page-ref #'segnoA "0" "?" }
}

```



Il primo brano inizia a pagina 1

Il segno A è a pagina 2

Il comando `markup \page-ref` prende tre argomenti:

1. l'etichetta – un simbolo Scheme – per esempio `#'primoBrano`;
2. un testo markup da usare come misura di riferimento per stimare le dimensioni del testo;
3. un testo markup che verrà usato al posto del numero di pagina se l'etichetta non viene trovata.

Il motivo per cui è necessario una misura di riferimento è che, nel momento in cui vengono interpretati i testi (markup), le interruzioni di pagina non sono state decise, quindi i numeri di pagina non sono ancora noti. Per aggirare il problema, la vera interpretazione del testo viene rimandata a un momento successivo; tuttavia le dimensioni del testo devono essere conosciute prima, ecco perché serve una misura di riferimento per decidere tali dimensioni. Se il libro ha un numero di pagine compreso tra 10 e 99, tale misura sarà "00", ovvero un numero di due cifre.

Comandi predefiniti

`\label`, `\page-ref`.

3.2.6 Indice

L'indice si include col comando `\markuplist \table-of-contents`. Gli elementi che devono apparire nell'indice si inseriscono col comando `\tocItem`, che può essere usato nel livello superiore o all'interno di un'espressione musicale.

```

\markuplist \table-of-contents
\pageBreak

```

```

\tocItem \markup "Primo brano"
\score {
  {
    c'4 % ...

```

```

        \tocItem \markup "Un punto preciso nel primo brano"
        d'4 % ...
    }
}

\tocItem \markup "Secondo brano"
\score {
{
    e'4 % ...
}
}

```

I testi markup usati per formattare l'indice sono definiti nel blocco `\paper`. Ce ne sono due predefiniti:

- `tocTitleMarkup`
Usato per formattare il titolo dell'indice.

```

tocTitleMarkup = \markup \huge \column {
    \fill-line { \null "Indice" \null }
    \null
}

```
- `tocItemMarkup`
Usato per formattare gli elementi dell'indice.

```

tocItemMarkup = \markup \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
}

```

Queste variabili possono essere entrambe modificate.

Ecco un esempio che mostra come cambiare il titolo dell'indice in francese:

```

\paper {
    tocTitleMarkup = \markup \huge \column {
        \fill-line { \null "Table des matières" \null }
        \hspace #1
    }
}

```

Ecco un esempio che mostra come cambiare il corpo dei caratteri nell'indice:

```

\paper {
    tocItemMarkup = \markup \large \fill-line {
        \fromproperty #'toc:text \fromproperty #'toc:page
    }
}

```

Nota il modo in cui sono citati il testo e il numero di pagina dell'elemento dell'indice nella definizione di `tocItemMarkup`.

Includendo il comando `\tocItemWithDotsMarkup` dentro `tocItemMarkup` lo spazio tra un elemento dell'indice e la sua pagina corrispondente sarà riempito con dei punti:

```

\header { tagline = ##f }
\paper {
    tocItemMarkup = \tocItemWithDotsMarkup
}

\book {

```

```

\markuplist \table-of-contents
\tocItem \markup { Allegro }
\tocItem \markup { Largo }
\markup \null
}

```

Table of Contents

Allegro	1
Largo	1

Si possono anche definire comandi personalizzati con markup specifici per creare un indice più complesso. Nell'esempio seguente, viene definito un nuovo stile per inserire i nomi degli atti nell'indice di un'opera.

Una nuova variabile di markup (chiamata `tocActMarkup`) viene definita nel blocco `\paper`:

```

\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

```

Viene quindi aggiunta una funzione musicale personalizzata (`tocAct`), che usa la nuova definizione di markup `tocActMarkup`:

```

tocAct =
#(define-music-function (text) (markup?)
  (add-toc-item! 'tocActMarkup text))

```

Un file di input di LilyPond, che usi queste definizioni personalizzate, potrebbe avere il seguente output:

Table of Contents

Atto Primo

Coro. Viva il nostro Alcide	1
Cesare. Presti omai l'Egizia terra	1

Atto Secondo

Sinfonia	1
Cleopatra. V'adoro, pupille, saette d'Amore	1

Ecco un esempio del comando `\fill-with-pattern` usato nel contesto di un indice:

```

\paper {
  tocItemMarkup = \markup { \fill-line {
    \override #'(line-width . 70)
    \fill-with-pattern #1.5 #CENTER . \fromproperty #'toc:text \fromproperty #'toc:page
  }
}
}

```


Vedi anche

File installati: `ly/toc-init.ly`.

Comandi predefiniti

`\table-of-contents`, `\tocItem`.

3.3 Lavorare coi file di input

3.3.1 Inclusione di file LilyPond

Se un progetto è grande e complesso, conviene suddividerlo in file separati. Per citare un altro file si usa

```
\include "altrofile.ly"
```

La riga `\include "altrofile.ly"` equivale a incollare i contenuti di `altrofile.ly` nel file corrente nel punto in cui appare il comando `\include`. Per esempio, in un progetto complesso si possono scrivere file separati per ogni parte strumentale e creare un file per l'“intera partitura” che raccoglie i file dei singoli strumenti. Di norma il file incluso definisce una serie di variabili che poi diventano disponibili nel file della partitura completa. Le sezioni contrassegnate con delle etichette nei file inclusi possono essere usate in varie parti di una partitura, vedi Sezione 3.3.2 [Edizioni diverse da un unico sorgente], pagina 525.

I file nella directory di lavoro corrente possono essere citati indicando semplicemente il nome del file dopo il comando `\include`. I file in altre posizioni possono essere inclusi sia con un percorso assoluto che con un percorso relativo (ma come separatore delle directory occorre usare la barra, o slash, come in UNIX, piuttosto che la barra inversa, o backslash, come in DOS/Windows). Per esempio, se `cose.ly` si trova una directory prima della directory di lavoro corrente, usare

```
\include "../cose.ly"
```

oppure se le parti orchestrali incluse si trovano tutte in una sottodirectory chiamata `parti` all'interno della directory corrente, usare

```
\include "parti/VI.ly"
\include "parti/VII.ly"
... etc
```

I file che devono essere inclusi possono contenere essi stessi delle dichiarazioni `\include`. Per impostazione predefinita, queste dichiarazioni `\include` di secondo livello non sono interpretate finché non vengono portate nel file principale, dunque i nomi dei file che specificano devono essere tutti relativi alla directory del file principale, non alla directory del file incluso. Tuttavia tale comportamento può essere cambiato globalmente tramite l'opzione `-drelative-includes` da linea di comando (oppure aggiungendo `#{ly:set-option 'relative-includes #t}` in cima al file di input principale).

Quando `relative-includes` viene impostato su `#t`, il percorso di ogni comando `\include` sarà considerato relativo al file che contiene quel comando. Questo comportamento è raccomandato e diventerà il comportamento predefinito in una versione futura di `lilypond`.

È possibile includere sia file relativi alla directory principale sia file relativi a qualche altra directory impostando `relative-includes` su `#t` o `#f` nei punti giusti dei file. Per esempio, se è stata creata una libreria generale, `libA`, che usa altri file inclusi dal file di base di quella libreria, tali dichiarazioni `\include` dovranno essere precedute da `#{ly:set-option 'relative-includes #t}` per poter essere interpretate correttamente quando riportate nel file `.ly` principale:

```
libA/
  libA.ly
  A1.ly
```

```
A2.ly
```

```
...
```

quindi il file di base, `libA.ly`, conterrà

```

#(ly:set-option 'relative-includes #t)
\include "A1.ly"
\include "A2.ly"
...
% ritorna alle impostazioni predefinite
#(ly:set-option 'relative-includes #f)

```

Qualsiasi file `.ly` può quindi includere l'intera libreria semplicemente con

```
\include "~/libA/libA.ly"
```

Si possono ideare strutture di file più complesse facendo dei cambi nei punti giusti.

È possibile includere dei file anche da una directory che si trova in un percorso di ricerca specificato come opzione quando si lancia LilyPond da linea di comando. I file inclusi possono allora essere specificati usando soltanto il loro nome. Per esempio, per compilare con questo metodo il file principale.ly che include i file di una sottodirectory chiamata parti, entrare nella directory di principale.ly e eseguire questo comando

```
lilypond --include=parti principale.ly
```

e in principale.ly scrivere

```

\include "VI.ly"
\include "VII.ly"
... etc

```

I file che devono essere inclusi in molte partiture possono essere salvati nella directory di installazione di LilyPond `../ly`. La posizione di questa directory dipende dal tipo di installazione (vedi Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*). Questi file possono quindi essere inclusi semplicemente nominandoli in una dichiarazione `\include`. Questo è il modo in cui sono inclusi i file che dipendono dalla lingua, come `english.ly`.

LilyPond include un certo numero di file quando si lancia il programma. Queste inclusioni non sono evidenti all'utente, ma i file possono essere identificati eseguendo `lilypond --verbose` dalla linea di comando. Così si vedrà un elenco di percorsi e file che LilyPond usa, insieme a tante altre informazioni. I più importanti di questi file sono trattati in Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*. Tali file possono essere modificati, ma le modifiche saranno perse quando si installa una nuova versione di LilyPond.

Alcuni semplici esempi d'uso di `\include` si trovano in Sezione “Partiture e parti” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*, Sezione “Partiture e parti” in *Manuale di Apprendimento*.

Problemi noti e avvertimenti

Se a un file incluso viene assegnato un nome identico a uno dei file di installazione di LilyPond, quest'ultimo ha la precedenza.

3.3.2 Edizioni diverse da un unico sorgente

Esistono vari metodi per generare versioni diverse di una partitura dalla stessa sorgente di musica. Le variabili sono forse le più utili per combinare lunghe sezioni musicali e/o note. Le etichette (*tag*) sono più utili per selezionare una sezione da varie sezioni brevi alternative e possono essere usate anche per unire insieme dei brani in punti diversi.

Qualsiasi metodo venga usato, la separazione delle note dalla struttura della partitura permetterà di cambiare la struttura lasciando le note intatte.

Uso delle variabili

Se le sezioni musicali sono definite in variabili, possono essere riutilizzate in varie parti della partitura, come è stato spiegato in Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*. Per esempio, una partitura vocale *a cappella* spesso comprende, ai fini delle prove, una riduzione per pianoforte delle parti, identiche alla musica vocale, dunque la musica deve essere inserita una volta sola. La musica definita in due variabili può essere combinata in un rigo, come è spiegato in [Combinazione automatica delle parti], pagina 194. Ecco un esempio:

```
sopranoMusic = \relative { a'4 b c b8( a) }
altoMusic = \relative { e'4 e e f }
tenorMusic = \relative { c'4 b e d8( c) }
bassMusic = \relative { a4 gis a d, }
allLyrics = \lyricmode { King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenore" {
    \clef "treble_8"
    \tenorMusic
  }
  \new Lyrics \allLyrics
  \new Staff = "Basso" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "Destra" {
      \partCombine
      \sopranoMusic
      \altoMusic
    }
    \new Staff = "Sinistra" {
      \clef "bass"
      \partCombine
      \tenorMusic
      \bassMusic
    }
  }
>>
>>
```



Partiture separate che mostrino soltanto le parti vocali o soltanto quelle per pianoforte possono essere prodotte semplicemente cambiando le dichiarazioni della struttura, lasciando la notazione musicale intatta.

Nel caso di partiture lunghe, conviene mettere le definizioni delle variabili in file separati da includere, vedi Sezione 3.3.1 [Inclusione di file LilyPond], pagina 524.

Uso delle etichette

Il comando `\tag #'parteA` contrassegna un'espressione musicale col nome *parteA*. Le espressioni contrassegnate in questo modo possono essere incluse o rimosse in base al loro nome successivamente, usando `\keepWithTag #'nome` oppure `\removeWithTag #'nome`. Il risultato dell'applicazione di questi filtri alla musica etichettata è il seguente:

Filtro

Musica etichettata preceduta da `\keepWithTag #'nome` o `\keepWithTag #'(nome1 nome2...)`

Musica etichettata preceduta da `\removeWithTag #'nome` o `\removeWithTag #'(nome1 nome2...)`

Musica etichettata non preceduta da `\keepWithTag` o `\removeWithTag`

Risultato

Viene inclusa la musica non etichettata e quella etichettata con uno dei nomi specificati; la musica etichettata con un nome etichetta diverso viene esclusa.

Viene inclusa la musica non etichettata e quella non etichettata con uno dei nomi specificati; la musica etichettata con uno dei nomi specificati viene esclusa.

Viene inclusa tutta la musica etichettata e non etichettata.

Gli argomenti dei comandi `\tag`, `\keepWithTag` e `\removeWithTag` devono essere un simbolo o una lista di simboli (come `#'score` o `#'(violinoI violinoII)`, seguiti da un'espressione musicale. Se e solo se i simboli sono identificatori LilyPond validi (solo caratteri alfabetici, nessun numero, trattino o trattino basso) che non possono essere confusi con le note, si può omettere il `#'` e, come scorciatoia, una lista di simboli può usare il punto come separatore: quindi `\tag #'(violinoI violinoII)` può essere riscritto come `\tag violinoI.violinoII`. Lo stesso vale per `\keepWithTag` e `\removeWithTag`.

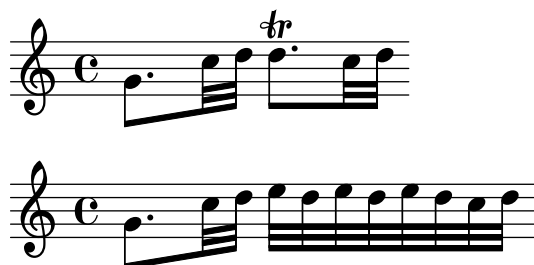
Nell'esempio seguente, vediamo due versioni di un brano musicale, una che mostra i trilli con la notazione abituale e l'altra con i trilli espansi esplicitamente:

```

musica = \relative {
  g'8. c32 d
  \tag #'trilli { d8.\trill }
  \tag #'espandi { \repeat unfold 3 { e32 d } }
  c32 d
}

\score {
  \keepWithTag #'trilli \musica
}
\score {
  \keepWithTag #'espandi \musica
}

```



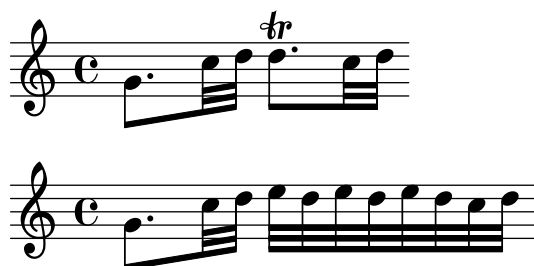
Altrimenti, talvolta è più facile escludere sezioni musicali:

```

musica = \relative {
  g'8. c32 d
  \tag #'trilli { d8.\trill }
  \tag #'espandi { \repeat unfold 3 { e32 d } }
  c32 d
}

\score {
  \removeWithTag #'espandi
  \musica
}
\score {
  \removeWithTag #'trilli
  \musica
}

```



Il filtro delle etichette può essere applicato a articolazioni, testo, etc. scrivendo

```
-\tag #'tua-etichetta
```

prima di un'articolazione. L'esempio seguente definisce una nota con un'indicazione di diteggiatura condizionale e una nota con un commento condizionale:

```
c1-\tag #'dito ^4
c1-\tag #'attenzione ^"Fai attenzione!"
```

Varie etichette possono essere associate a delle espressioni tramite molteplici `\tag`, o unendo molteplici etichette in una lista di simboli:

```
musica = \relative c'' {
  \tag #'a \tag #'entrambi { a4 a a a }
  \tag #'(b entrambi) { b4 b b b }
}
<<
\keepWithTag #'a \musica
\keepWithTag #'b \musica
\keepWithTag #'entrambi \musica
>>
```



Si possono applicare molteplici filtri `\removeWithTag` a una singola espressione musicale per togliere varie sezioni etichettate con nomi diversi. Altrimenti si può usare un solo `\removeWithTag` seguito da una lista di etichette.

```
musica = \relative c'' {
  \tag #'A { a4 a a a }
  \tag #'B { b4 b b b }
  \tag #'C { c4 c c c }
  \tag #'D { d4 d d d }
}
\new Voice {
  \removeWithTag #'B
  \removeWithTag #'C
  \musica
  \removeWithTag #'(B C)
  \musica
}
```



Applicando due o più filtri `\keepWithTag` su una singola espressione musicale toglieranno *tutte* le sezioni etichettate. Il primo filtro toglie tutto tranne la sezione menzionata e qualsiasi filtro successivo toglie il resto. Usando un singolo comando `\keepWithTag` con una lista di varie etichette vengono rimosse soltanto le sezioni etichettate non specificate in quella lista.

```
music = \relative c'' {
```

```

\tag #'violinoI { a4 a a a }
\tag #'violinoII { b4 b b b }
\tag #'viola { c4 c c c }
\tag #'cello { d4 d d d }
}

\new Staff {
  \keepWithTag #'(violinI violinII)
  \music
}

```

farà apparire le etichette *violinoI* e *violinoII* ma non *viola* o *cello*.

Mentre `\keepWithTag` è comodo con *un* gruppo di alternative, la rimozione di musica contrassegnata con etichette *indipendenti* è problematico se si usano le etichette con diverse finalità. In questo caso, è possibile dichiarare ‘gruppi’ di etichette:

```
\tagGroup #'(violinI violinII viola cello)
```

dichiara che queste etichette appartengono a un gruppo.

```
\keepWithTag #'violinoI ...
```

ora mostrerà soltanto la musica contrassegnata con le etichette presenti nel gruppo cui appartiene *violinoI* e verrà tolta la musica contrassegnata con una delle *altre* etichette.

```

music = \relative {
  \tagGroup #'(violinI violinII viola cello)
  \tag #'violinoI { c''4^"violinI" c c c }
  \tag #'violinoII { a2 a }
  \tag #'viola { e8 e e2. }
  \tag #'cello { d'2 d4 d }
  R1^"non contrassegnato"
}

\new Voice {
  \keepWithTag #'violinoI
  \music
}

```



Quando si usa il comando `\keepWithTag`, sono visibili solo le etichette provenienti dai gruppi cui appartengono le etichette specificate dopo il comando.

Talvolta si ha necessità di combinare insieme della musica in un punto preciso di un’espressione musicale esistente. `\pushToTag` e `\appendToTag` permettono di aggiungere materiale prima o dopo gli elementi di un costrutto musicale esistente. Non tutti i costrutti musicali hanno elementi, ma nel caso di musica sequenziale e simultanea si può esserne sicuri:

```

music = { \tag #'qui { \tag #'qui <<c''>> } }

{
  \pushToTag #'qui c'
  \pushToTag #'qui e'
  \pushToTag #'qui g' \music
  \appendToTag #'qui c'
}

```

```

\appendToTag #'qui e'
\appendToTag #'qui g' \music
}

```



Entrambi i comandi prendono tre argomenti: un’etichetta, il materiale da combinare ad ogni occorrenza dell’etichetta e l’espressione contrassegnata.

Vedi anche

Manuale di apprendimento: Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*.

Guida alla notazione: [Combinazione automatica delle parti], pagina 194, Sezione 3.3.1 [Inclusione di file LilyPond], pagina 524.

Problemi noti e avvertimenti

Se si usa `\relative` prima di un’espressione musicale ottenuta filtrando la musica con `\keepWithTag` o `\removeWithTag`, i rapporti di ottava potrebbero cambiare, perché verranno considerate solo le altezze rimaste nell’espressione filtrata. Per evitare questo rischio, usare `\relative` prima di `\keepWithTag` o `\removeWithTag`, in modo che `\relative` agisca su tutte le altezze prima del filtro.

Impostazioni globali

È possibile includere impostazioni globali da un altro file:

```
lilypond -dinclue-settings=MIE_IMPOSTAZIONI.ly MIA_PARTITURA.ly
```

Gruppi di impostazioni come dimensioni del foglio e tipo di carattere possono essere salvati in file separati. Ciò permette di ottenere diverse edizioni dalla stessa partitura o di applicare delle impostazioni tipiche a molte partiture, semplicemente indicando il relativo file delle impostazioni.

Questa tecnica è la stessa usata per i fogli di stile, trattati in Sezione “Fogli di stile” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*, Sezione “Fogli di stile” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 3.3.1 [Inclusione di file LilyPond], pagina 524.

3.3.3 Caratteri speciali

Codifica del testo

LilyPond usa il repertorio di caratteri definito dall’Unicode Consortium e dalla ISO/IEC 10646. Questo sistema di codifica definisce un nome e un numero univoci per gli insiemi di caratteri utilizzati in tutte le lingue moderne e anche in molte altre. Unicode può essere implementato in varie codifiche diverse. LilyPond usa la codifica UTF-8 (UTF sta per Unicode Transformation Format) che rappresenta tutti i comuni caratteri latini con un byte, e gli altri caratteri con un formato di lunghezza variabile fino a quattro byte.

L’aspetto grafico dei caratteri è determinato dai glifi definiti nei tipi di carattere disponibili - un tipo di carattere definisce la mappatura tra un sottoinsieme dei numeri Unicode e i glifi. LilyPond usa la libreria Pango per rappresentare e formattare i testi multilingua.

LilyPond non esegue alcuna conversione della codifica dell'input. Ciò significa che qualsiasi testo, sia esso un titolo, un testo vocale o un'istruzione musicale contenente caratteri non-ASCII, deve essere codificato in UTF-8. Il modo più semplice per inserire tale testo è usare un editor che sappia riconoscere la codifica Unicode e salvare il file con la codifica UTF-8. La maggior parte dei moderni editor supporta la codifica UTF-8, per esempio vim, Emacs, jEdit e Gedit. Tutti i sistemi MS Windows successivi a NT usano Unicode come codifica dei caratteri nativa, quindi perfino Notepad può modificare e salvare un file in formato UTF-8. Un'alternativa più efficiente per Windows è BabelPad.

Se un file di input LilyPond contenente un caratter non-ASCII non viene salvato in formato UTF-8, apparirà il seguente messaggio di errore:

```
FT_Get_Glyph_Name () error: invalid argument
```

Ecco un esempio che mostra del testo cirillico, ebraico e portoghese:



Unicode

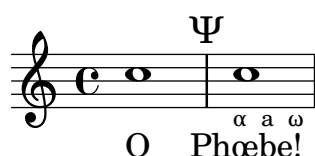
Per inserire un singolo carattere per il quale è noto il codice Unicode ma che non è disponibile nell'editor in uso, usare `\char ##xhhhh` o `\char #dddd` dentro un blocco `\markup`, dove `hhhh` è il codice esadecimale del carattere richiesto e `dddd` è il valore decimale corrispondente. Gli zero iniziali possono essere omessi, ma di norma nella rappresentazione esadecimale si specificano tutti e quattro i caratteri. (Fare attenzione al fatto che la codifica UTF-8 del codice *non* deve essere usata dopo `\char`, perché le codifiche UTF-8 contengono bit extra che indicano il numero di ottetti.) Le tabelle dei codici Unicode e un indice dei nomi dei caratteri col proprio codice esadecimale sono disponibili sul sito dell'Unicode Consortium, <https://www.unicode.org/>.

Per esempio, `\char ##x03BE` e `\char #958` corrispondono entrambi al carattere Unicode U+03BE, che ha il nome Unicode "Greek Small Letter Xi".

Qualsiasi codice Unicode può essere inserito in questo modo e se tutti i caratteri speciali sono inseriti in questo formato non è necessario salvare il file di input in formato UTF-8. Ovviamente, un tipo di carattere contenente tutti questi caratteri codificati deve essere installato e disponibile per LilyPond.

L'esempio seguente mostra valori esadecimali Unicode inseriti in quattro posti diversi: come numero di chiamata, come articolazione, nel testo vocale e come testo separato sotto il brano:

```
\score {
  \relative {
    c''1 \mark \markup { \char ##x03A8 }
    c1_\markup { \tiny { \char ##x03B1 " a " \char ##x03C9 } }
  }
  \addlyrics { 0 \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2023" \char ##x00A9 }
```



Copyright 2008--2023 ©

Per inserire il segno del copyright nell'apposito campo usare:

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

Alias ASCII

È possibile includere una lista di alias ASCII per i caratteri speciali:

```
\paper {
  #(include-special-characters)
}

\markup "&flqq; &ndash; &OE;uvre incomplète&hellip; &frqq;"

\score {
  \new Staff { \repeat unfold 9 a'4 }
  \addlyrics {
    Fun -- ziona an -- che nel~tes -- to: &ndash;_&OE;&hellip;
  }
}

\markup \column {
  "La sostituzione può essere disabilitata:"
  "&ndash; &OE; &hellip;"
  \override #'(replacement-alist . ()) "&ndash; &OE; &hellip;"
}

« – Œuvre incomplète... »
```



Funziona anche nel testo: – Œ...

La sostituzione può essere disabilitata:

– Œ ...

– &OE; …

Si possono creare i propri alias, sia globalmente:

```
\paper {
  #(add-text-replacements!
    '(("100" . "hundred")
      ("dpi" . "dots per inch")))
}
\markup "A 100 dpi."
```

A hundred dots per inch.

che localmente:

```
\markup \replace #'(("100" . "hundred")
  ("dpi" . "dots per inch")) "A 100 dpi."
```

A hundred dots per inch.

Vedi anche

Guida alla notazione: Sezione A.14 [Elenco dei caratteri speciali], pagina 792.

File installati: `ly/text-replacements.ly`.

3.4 Controllo dell'output

3.4.1 Estrarre frammenti musicali

È possibile creare output separati di uno o più frammenti di una partitura definendo i punti della musica da estrarre nel blocco `\layout` del file di input tramite la funzione `clip-regions`, e poi eseguendo LilyPond con l'opzione `-dclip-systems`;

```
\layout {
  clip-regions
  = #(list
    (cons
      (make-rhythmic-location 5 1 2)
      (make-rhythmic-location 7 3 4)))
}
```

Questo esempio estrarrà dal file di input un unico frammento che *inizia* dopo una minima nella quinta misura (5 1 2) e *termina* dopo la terza semiminima nella settima misura (7 3 4).

Si possono estrarre ulteriori frammenti aggiungendo altre coppie di `make-rhythmic-location` alla lista `clip-regions` del blocco `\layout`.

Ciascun frammento musicale verrà salvato in formato EPS, ma se necessario si possono creare anche altri formati come PDF o PNG. La musica estratta viene generata come se fosse letteralmente ‘tagliata’ dalla partitura a stampa originale; ciò significa che se un frammento supera una o più linee, verrà creato un file di output separato per ciascuna linea.

Vedi anche

Guida alla notazione: Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

Utilizzo: Sezione “Uso da linea di comando” in *Uso del Programma*.

3.4.2 Saltare la musica già corretta

Quando si inserisce o si copia della musica, di solito è utile visualizzare e correggere soltanto la musica vicina alla fine (ovvero dove si stanno inserendo le note). Per velocizzare il processo di correzione, è possibile far sì che il compositore tipografico salti tutte le misure eccetto le ultime. Per farlo basta inserire per esempio

```
showLastLength = R1*5
\score { ... }
```

nel file sorgente. In questo modo verranno elaborate soltanto le ultime 5 misure (assumendo che il tempo sia 4/4) di ogni blocco `\score` nel file di input. Per i brani più lunghi, elaborare solo una piccola parte è spesso molto più veloce di elaborarli completamente. Se si lavora sull'inizio di un brano già scritto (per esempio per aggiungere una nuova parte), si userà invece la proprietà `showFirstLength`.

È possibile saltare parti di una partitura in un modo più preciso tramite la proprietà `Score.skipTypesetting`. Quando è impostata su vero, la composizione tipografica è disattivata.

Questa proprietà viene usata anche per controllare l'output da inviare al file MIDI. Attenzione: salta tutti gli eventi, inclusi i cambi di tempo e di strumento.

```
\relative c' {
```

```

c1
\set Score.skipTypesetting = ##t
\tempo 4 = 80
c4 c c c
\set Score.skipTypesetting = ##f
d4 d d d
}

```



Nella musica polifonica, `Score.skipTypesetting` agisce su tutte le voci e su tutti i righi, facendo risparmiare tempo ulteriormente.

3.4.3 Formati di output alternativi

I formati di output predefiniti per la partitura stampata sono Portable Document Format (PDF) e PostScript (PS). Sono disponibili anche i formati di output Portable Network Graphics (PNG), Scalable Vector Graphics (SVG) e Encapsulated PostScript (EPS) attraverso le opzioni da linea di comando, vedi Sezione “Opzioni di base della linea di comando per LilyPond” in *Uso del Programma*.

Output SVG

L'output SVG può contenere opzionalmente dei metadati per gli oggetti grafici (grob) come teste di nota, pause, etc. Questi metadati possono essere attributi SVG standard come `id` e `class` oppure attributi personalizzati e non standard. Gli attributi e i loro valori si specificano sovrascrivendo la proprietà `output-attributes` di un grob con una lista associativa Scheme (alist). I valori possono essere numeri, stringhe o simboli. Per esempio:

```

{
  \once \override NoteHead.output-attributes =
  #'((id . 123)
     (class . "questo quello")
     (data-testo . qualcosa))
  c
}

```

L'input precedente produrrà il seguente elemento `<g>` (gruppo) nel file SVG:

```

<g id="123" class="questo quello" data-testo="qualcosa">
  ...Elementi SVG del grob NoteHead...
</g>

```

L'elemento `<g>` contiene tutti gli elementi SVG di un certo grob (alcuni grob generano molteplici elementi SVG). Nella sintassi SVG il prefisso `data-` è utilizzato per gli attributi di metadati personalizzati non standard.

3.4.4 Cambiare il tipo di carattere della notazione

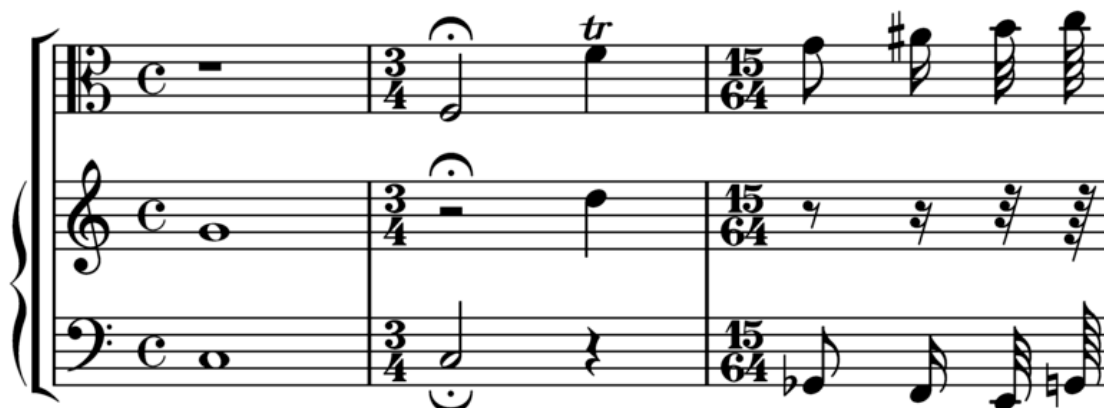
Gonville è un insieme di glifi alternativo a *Feta*, parte del font Emmentaler, e può essere usato in LilyPond. Può essere scaricato da:

```

http://www.chiark.greenend.org.uk/~sgtatham/gonville/ (http://www.chiark.
greenend.org.uk/~sgtatham/gonville/ )

```

Ecco alcune battute di musica che usa Gonville:



E alcune battute di musica che usa i glifi Feta di LilyPond:



Instruzioni di installazione

Scaricare e estrarre i file dei font. Copiare i file `gonville-11.otf`, `gonville-13.otf`, `gonville-14.otf`, `gonville-16.otf`, `gonville-18.otf`, `gonville-20.otf`, `gonville-23.otf`, `gonville-26.otf` e `gonville-brace.otf` in `.../share/lilypond/current/fonts/otf` o `.../share/lilypond/X.Y.Z/fonts/otf`. Se sono presenti file `gonville-*.svg` e `gonville-*.woff`, copiarli in `.../share/lilypond/current/fonts/svg` o `.../share/lilypond/X.Y.Z/fonts/svg`. Maggiori informazioni in Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Nota: i file `gonville-*.otf` sono utilizzati dal backend `ps` e `eps` (per la creazione di file PDF e PostScript). I file `gonville-*.svg` sono utilizzati dal backend `svg` quando non è attiva l'opzione `svg-woff`. I file `gonville-*.woff` sono utilizzati dal backend `svg` quando è attiva l'opzione `svg-woff`. Maggiori informazioni in Sezione “Opzioni avanzate della linea di comando per LilyPond” in *Uso del Programma*.

La sintassi seguente imposta il font Gonville come font della notazione (generale e delle parentesi).

```
\paper {
  property-defaults.fonts.music = "gonville"
}
```

Nota: ogni chiamata della funzione `set-global-fonts` reimposta completamente i font sia della notazione principale che del testo. Se una categoria non è specificata, allora verrà usato il font predefinito per quella categoria. Ogni chiamata di `set-global-fonts` cambia i font di ciascun blocco `\book` che la segue, sia esso creato esplicitamente o implicitamente. Ciò significa

che ogni blocco `\book` può avere il suo insieme di font principali usando `set-global-fonts` prima di esso. Maggiori informazioni in [Tipi di carattere per l'intero documento], pagina 277.

Vedi anche

Manuale di apprendimento: Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione A.8 [Il font Emmentaler], pagina 693, [Tipi di carattere per l'intero documento], pagina 277.

Problemi noti e avvertimenti

Gonville non può essere usato per scrivere notazione in ‘musica antica’ ed è probabile che i nuovi glifi introdotti in rilasci successivi di LilyPond non esistano in Gonville. Fare riferimento al sito web dell'autore per maggiori informazioni su queste e altre problematiche, inclusa la licenza di Gonville.

Altri font della notazione

Se si hanno altri font della notazione come `nomefont-*.otf`, `nomefont-*.svg` e `nomefont-*.woff`, si possono usare nello stesso modo che abbiamo visto con Gonville.

Ovvero si copiano i file `nomefont-*.otf` in `.../share/lilypond/current/fonts/otf` o `.../share/lilypond/X.Y.Z/fonts/otf`. E i file `nomefont-*.svg` e `nomefont-*.woff`, se presenti, in `.../share/lilypond/current/fonts/svg` o `.../share/lilypond/X.Y.Z/fonts/svg`.

Nota: per il momento, LilyPond richiede che i nomi dei file dei font abbiano i seguenti suffissi, che devono essere tutti presenti nelle cartelle di installazione menzionate sopra perché possa funzionare correttamente: -11, -13, -14, -16, -18, -20, -23, -26, -brace. Per esempio, `emmentaler-11.otf`, `emmentaler-20.svg` e `emmentaler-brace.woff`, etc.

La sintassi seguente imposta il font `nomefont` come font della notazione (generale e delle parentesi).

```
\paper {
  #(define fonts
    (set-global-fonts
      #:music "fontname" ; nome del file del font senza suffisso né estensione
      #:brace "fontname" ; nome del file del font senza suffisso né estensione
    ))
}
```

Nota: per le categorie `music` e `brace`, specificare il nome del file del font senza suffisso né estensione.

3.5 Creazione dell'output MIDI

LilyPond è in grado di produrre dei file conformi allo standard MIDI (Musical Instrument Digital Interface) e dunque permette di fare un controllo uditivo dell'output musicale (con l'aiuto di un'applicazione o un dispositivo che comprendano il MIDI). L'ascolto dell'output MIDI può aiutare anche a individuare errori, come note inserite in modo erraneo o note senza alterazioni e così via.

I file MIDI non contengono audio (come i file AAC, MP3 o Vorbis), bensì richiedono un software esterno che produca il suono a partire da essi.

3.5.1 Notazione supportata nel MIDI

I seguenti elementi della notazione musicale saranno resi nell'output MIDI usando le capacità predefinite di LilyPond:

- Respiri

- Accordi inseriti come nomi degli accordi
- Crescendi e decrescendi su varie note. Il volume viene modificato in modo proporzionato fra i due estremi
- Segni di dinamica da ppppp a fffff, inclusi mp, mf e sf
- Microtoni ma *non* gli accordi microtonali. È necessario anche un lettore MIDI che supporti la “piegatura” (*bending*) delle altezze.
- Testo vocale
- Altezze
- Ritmi inseriti come durate, inclusi i gruppi irregolari
- Articolazioni ‘semplici’: staccato, staccatissimo, accento, marcato e portato
- Cambi di tempo tramite la funzione `\tempo`
- Legature di valore
- Tremoli che *non* sono inseriti con un valore ‘:[*numero*]’

Alcuni effetti, come panning, bilanciamento, espressione, riverbero e chorus, si gestiscono impostando le proprietà di contesto, vedi Sezione 3.5.8 [Proprietà di contesto per gli effetti MIDI], pagina 548.

Se si usa lo script *articulate*, anche i seguenti elementi della notazione musicale saranno presenti nell’output MIDI:

- Appoggiature. Servono a rubare metà del valore della nota che le segue (senza considerare i punti). Per esempio:

```
\appoggiatura c8 d2.
```

Il c avrà il valore di una semiminima.

- Ornamenti (mordenti, trilli, gruppetti, etc.)
- Rallentando, accelerando, ritardando e a tempo
- Legature di portamento, incluse le legature di frase
- Tenuto

Vedi Sezione 3.5.9 [Miglioramento dell’output MIDI], pagina 549.

3.5.2 Notazione non supportata nel MIDI

I seguenti elementi della notazione musicale non possono essere trasferiti nel file MIDI:

- Articolazioni diverse da staccato, staccatissimo, accento, marcato e portato
- Crescendi e decrescendi su una *singola* nota
- Corona
- Basso numerato (continuo)
- Glissandi
- Portamenti indeterminati verso il basso e verso l’alto
- Accordi microtonali
- Ritmi inseriti come note, per esempio lo swing
- Cambi di tempo senza usare `\tempo` (per esempio, inseriti come note)
- Tremoli che *sono* inseriti con un valore ‘:[*numero*]’

3.5.3 Il blocco MIDI

Per creare un file MIDI da un file di input LilyPond, inserire un blocco `\midi`, che può essere vuoto, all'interno di un blocco `\score`;

```
\score {
  ... musica ...
  \layout { }
  \midi { }
}
```

Nota: Un blocco `\score` che, oltre alla musica, contenga soltanto un blocco `\midi` (ovvero *sia privo* del blocco `\layout`), produrrà come output soltanto file MIDI e nessun file visuale.

L'estensione predefinita del file di output (`.midi`) può essere modificata tramite l'opzione `-dmidi-extension` del comando `lilypond`:

```
lilypond -dmidi-extension=mid MyFile.ly
```

Altrimenti, aggiungere la seguente espressione Scheme prima dell'inizio dei blocchi `\book`, `\bookpart` o `\score`. Vedi Sezione 3.1.5 [Struttura del file], pagina 497.

```
#{ly:set-option 'midi-extension "mid"}
```

Vedi anche

Guida alla notazione: Sezione 3.1.5 [Struttura del file], pagina 497, Sezione 3.2.3 [Creazione di metadati per i file di output], pagina 511.

File installati: `scm/midi.scm`.

Problemi noti e avvertimenti

Sono disponibili quindici canali MIDI e un canale ulteriore (`#10`) per le percussioni. I righi sono assegnati ai canali in sequenza: in una partitura di più di quindici righi, i righi extra condivideranno (senza sovrascriverlo) lo stesso canale MIDI. Ciò potrebbe essere un problema se i righi in questione sono impostati con proprietà MIDI in conflitto e basate sul canale, come ad esempio diversi strumenti MIDI.

L'uso di un blocco `midi` con notazione polimetrica potrebbe produrre avvisi imprevisti di controllo della battuta. In questo caso occorre spostare `Timing_translator` dal contesto `Score` al contesto `Staff` all'interno del blocco `midi`.

```
\midi {
  \context {
    \Score
    \remove Timing_translator
  }
  \context {
    \Staff
    \consists Timing_translator
  }
}
```

3.5.4 Gestione delle dinamiche nel MIDI

È possibile regolare il volume MIDI complessivo, il volume relativo dei segni di dinamica e il volume relativo dei diversi strumenti.

Le dinamiche vengono trasferite automaticamente sui livelli di volume nella gamma di volume MIDI disponibile, mentre crescendo e decrescendo variano il volume gradualmente tra i loro due estremi. È possibile regolare il volume relativo delle dinamiche e i livelli del volume complessivo dei diversi strumenti.

Dinamiche nel MIDI

Solo i segni di dinamica compresi tra *ppppp* e *fffff*, inclusi *mp*, *mf* e *sf* hanno dei valori assegnati. Questo valore viene poi applicato al valore della gamma del volume MIDI complessivo per ottenere il volume finale incluso nell'output MIDI per quella particolare dinamica. Le frazioni predefinite vanno da 0.25 per *ppppp* a 0.95 per *fffff*. L'insieme completo di dinamiche e delle loro frazioni associate si trova in `scm/midi.scm`.

Frammenti di codice selezionati

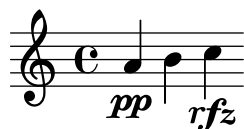
Creare dinamiche personalizzate nell'output MIDI

L'esempio seguente mostra come creare un segno di dinamica, non incluso nell'elenco predefinito, e assegnargli un valore specifico così che possa essere usato per cambiare l'output MIDI.

Al segno di dinamica `\rfz` viene assegnato il valore 0.9.

```
#(define (myDynamics dynamic)
  (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = "cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



File installati: `ly/script-init.ly` `scm/midi.scm`.

Frammenti: Sezione “MIDI” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Dynamic-performer” in *Guida al Funzionamento Interno*.

Impostazione del volume MIDI

I valori minimo e massimo del volume complessivo delle dinamiche MIDI sono regolati dalle proprietà `midiMinimumVolume` e `midiMaximumVolume` nel livello `Score`. Tali proprietà hanno

effetto soltanto all'inizio di una voce e sui segni di dinamica. La frazione corrispondente a ciascun segno di dinamica viene modificata con la seguente formula:

$$\text{midiMinimumVolume} + (\text{midiMaximumVolume} - \text{midiMinimumVolume}) * \text{frazione}$$

Nell'esempio seguente la gamma dinamica del volume MIDI complessivo è limitata alla gamma 0.2 - 0.5.

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flauto"
      ... musica ...
    }
    \new Staff {
      \set Staff.midiInstrument = "clarinetto"
      ... musica ...
    }
  >>
  \midi {
    \context {
      \Score
      midiMinimumVolume = #0.2
      midiMaximumVolume = #0.5
    }
  }
}
```

Una semplice equalizzazione degli strumenti MIDI si può ottenere impostando le proprietà `midiMinimumVolume` e `midiMaximumVolume` nel contesto `Staff`.

```
\score {
  \new Staff {
    \set Staff.midiInstrument = "flauto"
    \set Staff.midiMinimumVolume = #0.7
    \set Staff.midiMaximumVolume = #0.9
    ... musica ...
  }
  \midi { }
}
```

In caso di partiture con molti righe e molti strumenti MIDI, i volumi relativi di ogni strumento possono essere impostati individualmente;

```
\score {
  <<
    \new Staff {
      \set Staff.midiInstrument = "flauto"
      \set Staff.midiMinimumVolume = #0.7
      \set Staff.midiMaximumVolume = #0.9
      ... musica ...
    }
    \new Staff {
      \set Staff.midiInstrument = "clarinetto"
      \set Staff.midiMinimumVolume = #0.3
      \set Staff.midiMaximumVolume = #0.6
      ... musica ...
    }
  >>
}
```

```

    }
  >>
  \midi { }
}

```

In questo esempio il volume del clarinetto è diminuito in modo proporzionale al volume del flauto.

Se queste proprietà del volume non sono impostate, LilyPond applica comunque “un po” di equalizzazione a certi strumenti. Vedi `scm/midi.scm`.

File installati: `scm/midi.scm`.

Vedi anche

Guida alla notazione: Sezione 4.2 [Formattazione della partitura], pagina 563.

Guida al funzionamento interno: Sezione “Dynamic_performer” in *Guida al Funzionamento Interno*.

Frammenti di codice selezionati

Modificare l’equalizzazione predefinita degli strumenti MIDI

L’equalizzatore predefinito degli strumenti MIDI può essere modificato impostando la proprietà `instrumentEqualizer` nel contesto `Score` come una procedura Scheme definita dall’utente che usi il nome dello strumento MIDI come argomento insieme a una coppia di frazioni indicanti i volumi minimi e massimi da applicare a quello specifico strumento.

L’esempio seguente imposta i volumi massimo e minimo per il flauto e per il clarinetto.

```

#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g'\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
  \new Staff {

```

```

\key g \major
\set Staff.midiInstrument = "clarinet"
\new Voice \relative {
  b'1\p a2. b8 a
  g2. fis8 e
  fis2 r
}
}
>>
\layout { }
\midi { }
}

```



Problemi noti e avvertimenti

Le modifiche al volume MIDI si verificano soltanto all'inizio di una nota, quindi i crescendo e i decrescendo non possono cambiare il volume di una singola nota.

Impostazione delle proprietà del blocco MIDI

Il blocco `\midi` può contenere modifiche del contesto, nuove definizioni del contesto o codice che imposti i valori di certe proprietà.

```

\score {
  ... musica ...
  \midi {
    \tempo 4 = 72
  }
}

```

In questo esempio il tempo è impostato su 72 battiti da un quarto per minuto. Il segno di tempo nel blocco `\midi` non appare nella partitura, mentre qualsiasi altra indicazione di `\tempo` specificata nel blocco `\score` sarà trasferita anche nell'output MIDI.

Se all'interno di un blocco `\midi`, il comando `\tempo` imposta le proprietà durante l'interpretazione della musica e nel contesto delle definizioni di output; dunque viene interpretato *come se* fosse una modifica di contesto.

Le definizioni di contesto seguono la stessa sintassi di quelle di un blocco `\layout`;

```

\score {
  ... musica ...
  \midi {
    \context {
      \Voice
      \remove Dynamic_performer
    }
  }
}

```

Questo esempio toglie l'effetto delle dinamiche dall'output MIDI. Nota bene: i moduli di traduzione di LilyPond usati per l'audio si chiamano 'performer'.

Vedi anche

Manuale di apprendimento: Sezione "Altre fonti di informazione" in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.3 [Segni di espressione], pagina 129, Sezione 4.2 [Formattazione della partitura], pagina 563.

File installati: ly/performer-init.ly.

Frammenti: Sezione "MIDI" in *Frammenti di codice*.

Guida al funzionamento interno: Sezione "Dynamic_performer" in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Alcuni lettori MIDI non sempre gestiscono correttamente i cambi di tempo.

I cambi di midiInstrument, e di altre opzioni MIDI, all'*inizio* di un rigo potrebbero apparire due volte nell'output MIDI.

3.5.5 Uso degli strumenti MIDI

Gli strumenti MIDI si impostano tramite la proprietà midiInstrument del contesto Staff.

```
\score {
  \new Staff {
    \set Staff.midiInstrument = "glockenspiel"
    ... musica ...
  }
  \midi { }
}

o

\score {
  \new Staff \with {midiInstrument = "cello"} {
    ... musica ...
  }
  \midi { }
}
```

Se il nome dello strumento non corrisponde a nessuno degli strumenti elencati nella sezione 'Strumenti MIDI', verrà usato lo strumento acoustic grand. Vedi Sezione A.6 [Strumenti MIDI], pagina 689.

Vedi anche

Manuale di apprendimento: Sezione "Altre fonti di informazione" in *Manuale di Apprendimento*.

Guida alla notazione: Sezione A.6 [Strumenti MIDI], pagina 689, Sezione 4.2 [Formattazione della partitura], pagina 563.

File installati: scm/midi.scm.

Problemi noti e avvertimenti

Gli strumenti percussivi che sono scritti in un contesto DrumStaff verranno inviati, correttamente, al canale MIDI 10, ma alcuni strumenti percussivi con un tono musicale, come xilofono, marimba, vibrafono o timpano, sono trattati come strumenti "normali", quindi la musica di

tali strumenti deve essere inserita in un contesto Staff (non DrumStaff) per ottenere l'output MIDI corretto. L'elenco completo delle voci del set di percussioni del canale 10 channel 10 drum-kits si trova in scm/midi.scm. Vedi Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

3.5.6 Uso delle ripetizioni nel MIDI

Le ripetizioni possono essere applicate all'output MIDI tramite il comando `\unfoldRepeats`.

```
\score {
  \unfoldRepeats {
    \repeat tremolo 8 { c'32 e' }
    \repeat percent 2 { c''8 d'' }
    \repeat volta 2 { c'4 d' e' f' }
    \alternative {
      { g' a' a' g' }
      { f' e' d' c' }
    }
  }
  \midi { }
}
```

Per limitare l'effetto di `\unfoldRepeats` all'output MIDI soltanto, senza modificare la partitura stampata, occorre creare *due* blocchi `\score`; uno per il MIDI (con ripetizioni ricopiate) e uno per la partitura scritta (con ripetizioni con volta, tremolo e percentuale):

```
\score {
  ... musica ...
  \layout { }
}
\score {
  \unfoldRepeats {
    ... musica ...
  }
  \midi { }
}
```

Quando si usa più di una voce, ciascuna voce deve avere tutte le ripetizioni ricopiate per generare un output MIDI corretto.

Vedi anche

Guida alla notazione: Sezione 1.4 [Ripetizioni], pagina 160.

3.5.7 Mappatura dei canali MIDI

Quando genera un file MIDI da una partitura, LilyPond assegna automaticamente ogni nota a un canale MIDI, in cui essa dovrà essere suonata quando inviata a un dispositivo MIDI. Un canale MIDI mette a disposizione un certo numero di controlli per selezionare, per esempio, lo strumento da usare per riprodurre le note in quel canale o per chiedere al dispositivo MIDI di applicare vari effetti al suono prodotto nel canale. In ogni momento, ogni controllo di un canale MIDI può avere un solo valore assegnato (che tuttavia può essere modificato, per esempio, per passare a un altro strumento nel mezzo di un brano).

Lo standard MIDI supporta soltanto 16 canali per dispositivo MIDI. Questo limite al numero di canali limita anche il numero dei diversi strumenti che possono essere eseguiti in contemporanea.

LilyPond crea tracce MIDI separate per ogni rigo (o strumento o voce separati, a seconda del valore di `Score.midiChannelMapping`) e anche per ogni contesto lyrics. Non c'è limite al numero di tracce.

Per aggirare il limite del numero di canali MIDI, LilyPond supporta varie modalità per l'assegnazione dei canali MIDI, scelti attraverso la proprietà di contesto `Score.midiChannelMapping`. In ogni caso, se è necessario un numero di canali MIDI superiore al limite, i numeri canale assegnati ripartono da 0, causando possibili assegnazioni erranee degli strumenti di alcune note. Questa proprietà di contesto può essere impostata su uno dei seguenti valori:

`'staff`

Assegna un canale MIDI separato a ogni rigo della partitura (comportamento predefinito). Tutte le note in tutte le voci di ogni rigo condivideranno lo stesso canale MIDI del rigo che le racchiude, e si trovano tutte nella stessa traccia MIDI.

Il limite dei 16 canali è applicato al numero totale dei contesti staff (rigo) e lyrics (testo vocale), anche se il testo vocale nel MIDI non occupa un canale MIDI.

`'instrument`

Assegna un canale MIDI separato a ogni strumento MIDI diverso specificato nella partitura. Ciò significa che tutte le note suonate con lo stesso strumento MIDI condivideranno lo stesso canale (e traccia) MIDI, anche se le note appartengono a voci o righe diversi.

In questo caso i contesti del testo vocale (lyrics) non contano nel calcolo del limite dei 16 canali MIDI (perché non saranno assegnati a uno strumento MIDI), dunque questa impostazione consente una migliore assegnazione dei canali MIDI quando il numero di righe e contesti lyrics in una partitura è superiore a 16.

`'voice`

Assegna un canale MIDI separato a ogni voce che abbia un nome unico tra le voci del rigo che le racchiude. Alle voci in righe diversi vengono sempre assegnati canali MIDI separati, ma due voci di uno stesso rigo condivideranno lo stesso canale MIDI se hanno lo stesso nome. `midiInstrument` e i vari controlli MIDI per gli effetti, essendo proprietà del contesto del rigo (staff), non possono essere impostati separatamente per ogni voce. La prima voce verrà suonata con lo strumento e gli effetti specificati per il rigo, mentre alle voci con un nome diverso da quello della prima saranno assegnati lo strumento e gli effetti predefiniti.

Nota bene: è possibile assegnare diversi strumenti e/o effetti a varie voci dello stesso rigo spostando `Staff_performer` dal contesto Staff al contesto Voice e lasciando `midiChannelMapping` sul valore predefinito `'staff` oppure impostandolo su `'instrument`; vedi il frammento in basso.

Per esempio, la mappatura predefinita dei canali MIDI di una partitura può essere modificata per usare l'impostazione `'instrument`:

```
\score {
  ...musica...
  \midi {
    \context {
      \Score
      midiChannelMapping = #'instrument
    }
  }
}
```

Frammenti di codice selezionati

Impostare l'output MIDI su un canale per voce

Nella creazione del file di output MIDI, il comportamento predefinito prevede che ogni rigo sia assegnato a un canale MIDI, con tutte le voci del rigo amalgamate in un canale. Ciò diminuisce il rischio di esaurire i canali MIDI disponibili, dato che ce ne sono solo 16 per traccia.

Tuttavia, spostando `Staff_performer` nel contesto `Voice`, ogni voce in un rigo può avere il proprio canale MIDI, come è illustrato nell'esempio seguente: sebbene le voci siano sullo stesso rigo, vengono creati due canali MIDI, ciascuno con un diverso strumento MIDI (`midiInstrument`).

```
\score {
  \new Staff <<
    \new Voice \relative c'' {
      \set midiInstrument = "flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = "clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```



3.5.8 Proprietà di contesto per gli effetti MIDI

Le seguenti proprietà di contesto possono essere usate per applicare vari effetti MIDI alle note suonate sul canale MIDI associato col rigo, strumento MIDI o voce correnti (a seconda del valore della proprietà di contesto `Score.midiChannelMapping` e del contesto in cui si trova `Staff_performer`; vedi Sezione 3.5.7 [Mappatura dei canali MIDI], pagina 545).

La modifica di queste proprietà di contesto avrà effetto su tutte le note suonate sul canale dopo il cambio, tuttavia alcuni effetti potrebbero essere applicati anche a note che stanno già suonando (a seconda di come è stato implementato il dispositivo MIDI).

Sono supportate le seguenti proprietà di contesto:

`Staff.midiPanPosition`

La posizione del panning regola il modo in cui il suono in un canale MIDI è distribuito tra gli altoparlanti stereo di sinistra e di destra. Questa proprietà di contesto accetta un numero compreso tra -1.0 (`#LEFT`) e 1.0 (`#RIGHT`); il valore -1.0 sposterà tutto il suono sull'altoparlante sinistro (rendendo muto quello destro), il valore 0.0 (`#CENTER`) distribuirà il suono in modo uniforme tra gli altoparlanti sinistro e destro, e il valore 1.0 sposterà tutto il suono sull'altoparlante destro. I valori intermedi tra -1.0 e 1.0 permettono di ottenere distribuzioni miste tra gli altoparlanti sinistro e destro.

`Staff.midiBalance`

Il bilanciamento stereo di un canale MIDI. In modo simile al panning, questa proprietà di contesto accetta un numero compreso tra -1.0 (`#LEFT`) e 1.0 (`#RIGHT`). Varia il volume relativo inviato ai due altoparlanti stereo senza alterare la distribuzione dei segnali stereo.

`Staff.midiExpression`

Livello dell'espressione (come frazione del livello massimo disponibile) da applicare a un canale MIDI. Un dispositivo MIDI combina insieme il livello di espressione del canale MIDI con l'attuale livello di dinamica di una voce (stabilito dai comandi `\p` o `\ff`) per ottenere il volume totale di ogni nota nella voce. Il controllo dell'espressione può essere usato, per esempio, per implementare effetti di crescendo o decrescendo su singole note sostenute (non supportato automaticamente da LilyPond).

La gamma del livello di espressione va da 0.0 (nessuna espressione, ovvero volume zero) a 1.0 (piena espressione).

`Staff.midiReverbLevel`

Livello di riverbero (come frazione del livello massimo disponibile) da applicare a un canale MIDI. Questa proprietà accetta numeri compresi tra 0.0 (nessun riverbero) e 1.0 (pieno effetto).

`Staff.midiChorusLevel`

Livello del chorus (come frazione del livello massimo disponibile) da applicare a un canale MIDI. Questa proprietà accetta numeri compresi tra 0.0 (nessun effetto chorus) e 1.0 (pieno effetto).

Problemi noti e avvertimenti

Dato che i file MIDI non contengono veri dati audio, le modifiche in queste proprietà di contesto si traducono soltanto in richieste di modifica dei controlli del canale MIDI nei file MIDI generati. Se un particolare dispositivo MIDI (come un lettore MIDI) sia in grado di gestire queste richieste dipende interamente dall'implementazione del dispositivo: un dispositivo potrebbe scegliere di ignorare alcune o tutte le richieste. Dipende dall'implementazione del dispositivo MIDI anche il modo in cui un dispositivo MIDI interpreta valori diversi di questi controlli (generalmente lo standard MIDI corregge il comportamento solo alle estremità della gamma di valori disponibile).

per ogni controllo) e se un cambiamento nel valore di un controllo avrà effetto anche su note che stanno già suonando su quel canale MIDI.

Quando genera file MIDI, LilyPond trasformerà semplicemente le frazioni comprese in ciascuna gamma in valori di una gamma di numeri interi corrispondenti (0-127 per i 7-bit, 0-16383 per i 14 bit per i controlli del canale MIDI che supportano una migliore risoluzione), arrotondando i valori delle frazioni sul numero intero più vicino. I valori interi convertiti sono salvati così come sono nel file MIDI generato. Consultare la documentazione del proprio dispositivo MIDI per sapere come il dispositivo interpreta questi valori.

3.5.9 Miglioramento dell'output MIDI

L'output MIDI predefinito fornisce le funzionalità basilari, ma può essere migliorato impostando gli strumenti MIDI, le proprietà del blocco `\midi` e/o usando lo script `articulate`.

Lo script `articulate`

Per usare lo script `articulate` aggiungere il relativo comando `\include` all'inizio del file di input:

```
\include "articulate.ly"
```

Lo script crea file MIDI in cui la durata delle note è correttamente “ridimensionata” per rappresentare molte articolazioni e indicazioni di tempo. Tuttavia anche l'output grafico verrà modificato per corrispondere esattamente all'output MIDI.

```
\score {
  \articulate <<
    ... musica ...
  >>
  \midi { }
}
```

Il comando `\articulate` permette di elaborare le abbreviazioni (come i trilli e i gruppetti). L'elenco completo degli elementi supportati si trova nello script stesso. Vedi `ly/articulate.ly`.

Vedi anche

Manuale di apprendimento: Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 4.2 [Formattazione della partitura], pagina 563.

File installati: `ly/articulate.ly`.

Nota: Lo script `articulate` potrebbe accorciare la durata degli accordi, cosa che potrebbe non essere appropriata per alcuni tipi di strumento, come nel caso della musica per organo. Potrebbe essere ridotta anche la durata di note prive di articolazioni. Dunque, per consentire tutto ciò, limitare l'uso del comando `\articulate` a brevi segmenti di musica o modificare i valori delle variabili definite nello script `articulate` per compensare il comportamento di accorciamento della durata delle note.

3.6 Estrazione dell'informazione musicale

Oltre a creare l'output grafico e il MIDI, LilyPond può mostrare l'informazione musicale in forma testuale.

3.6.1 Mostrare la notazione LilyPond

La funzione musicale `\displayLilyMusic` permette di mostrare un'espressione musicale nella notazione di LilyPond. Per vedere l'output, di norma si esegue LilyPond da linea di comando. Per esempio,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

mostrerà

```
{ a,4 cis4 e4 fis4 g4 }
```

LilyPond stampa questi messaggi nella console insieme a tutti gli altri messaggi della compilazione. Per separare questi messaggi e salvare i risultati di `\displayLilyMusic`, reindirizzare l'output su un file:

```
lilypond file.ly >display.txt
```

Nota bene che LilyPond non soltanto mostra in console l'espressione musicale, ma la interpreta anche (infatti l'espressione musicale di `\displayLilyMusic` appare nell'output oltre a essere mostrata in console). Basta inserire `\displayLilyMusic` in file esistenti, senza doverne modificare le note di input, per ottenere informazioni su di essi. Per interpretare e mostrare una sezione musicale nella console ma, allo stesso tempo, toglierla dal file di output, usare il comando `\void`:

```
{
  \void \displayLilyMusic \transpose c a, { c4 e g a bes }
  c1
}
```

3.6.2 Mostrare le espressioni musicali Scheme

Vedi Sezione “Displaying music expressions” in *Estendere*.

3.6.3 Salvare eventi musicali in un file

Gli eventi musicali possono essere salvati in file diversi, un file per ogni rigo, includendo un file nella partitura principale:

```
\include "event-listener.ly"
```

Così verranno creati uno o più file `NOMEFILE-NOMERIGO.notes` o `NOMEFILE-unnamed-staff.notes` per ogni rigo. Se ci sono vari rigi senza nome, gli eventi di tutti i rigi saranno mischiati insieme nello stesso file. I file hanno una struttura di questo tipo:

```
0.000  note      57      4  p-c 2 12
0.000  dynamic  f
0.250  note      62      4  p-c 7 12
0.500  note      66      8  p-c 9 12
0.625  note      69      8  p-c 14 12
0.750  rest      4
0.750  breathe
```

La sintassi prevede una riga delimitata da tabulatori, con due campi fissi su ogni riga seguiti da parametri opzionali.

```
tempo  tipo  ...parametri...
```

Questa informazione può essere letta facilmente da altri programmi, per esempio da uno script python, e può essere molto utile per ricercatori che desiderano fare delle analisi musicali o degli esperimenti di *playback* con LilyPond.

Problemi noti e avvertimenti

Non tutti gli eventi musicali di lilypond sono supportati da `event-listener.ly`, che vuole essere una semplice “prova di concetto” ben congeniata. Se alcuni eventi che si vorrebbero vedere non sono inclusi, copiare `event-listener.ly` nella propria directory lilypond e modificare il file in modo che produca l’informazione desiderata.

4 Gestione dello spazio

La formattazione globale del foglio è determinata da tre fattori: la formattazione della pagina, le interruzioni di linea e la spaziatura. Ciascun fattore influenza l'altro. La scelta della spaziatura determina la densità con cui vengono disposti i sistemi musicali, che a sua volta influenza la scelta delle interruzioni di linea e quindi infine quante pagine occupa un brano.

Generalmente, questa procedura si svolge in quattro stadi. Inizialmente vengono scelte le distanze flessibili ('springs'), in base alle durate. Poi vengono tentate tutte le possibili combinazioni per le interruzioni di linea e viene calcolato un punteggio 'negativo' per ciascuna di esse. Quindi viene calcolata l'altezza di ogni possibile sistema. Infine viene scelta una combinazione di interruzioni di linea e di interruzioni di pagina che assicuri che la spaziatura verticale e quella orizzontale non siano né troppo compresse né troppo allungate.

Due tipi di blocchi possono contenere le impostazioni di formattazione: `\paper {...}` e `\layout {...}`. Il blocco `\paper` contiene le impostazioni di formattazione della pagina da applicare a tutte le partiture di un libro o di una parte, come l'altezza del foglio o se stampare o meno i numeri di pagina, etc. (vedi Sezione 4.1 [Formattazione della pagina], pagina 552). Il blocco `\layout` contiene le impostazioni di formattazione della partitura, come il numero di sistemi da usare o lo spazio tra i gruppi di righe, etc. (vedi Sezione 4.2 [Formattazione della partitura], pagina 563).

4.1 Formattazione della pagina

Questa sezione tratta le opzioni di formattazione della pagina per il blocco `\paper`.

4.1.1 Il blocco `\paper`

I blocchi `\paper` possono essere posizionati in tre punti diversi, in modo da formare una gerarchia discendente di blocchi `\paper`:

- All'inizio del file di input, prima di tutti i blocchi `\book`, `\bookpart` e `\score`.
- Dentro un blocco `\book` ma fuori da tutti i blocchi `\bookpart` e `\score` in esso racchiusi.
- Dentro un blocco `\bookpart` ma fuori da tutti i blocchi `\score` in esso racchiusi.

Non è possibile inserire un blocco `\paper` in un blocco `\score`.

I valori dei campi vengono filtrati dall'alto verso il basso attraverso questa gerarchia: i valori impostati nei livelli più alti della gerarchia persistono finché non sono sovrascritti da un valore impostato in un livello più basso.

Vari blocchi `\paper` possono apparire in ognuno di questi livelli, per esempio come parti di vari file inclusi con `\include`. In questo caso, i campi di ciascun livello vengono combinati e i valori riscontrati per ultimi avranno la precedenza in caso di campi duplicati.

Le impostazioni che possono apparire in un blocco `\paper` comprendono:

- la funzione Scheme `set-paper-size`,
- le variabili `\paper` per personalizzare la formattazione della pagina e
- le definizioni di markup usate per personalizzare la formattazione di intestazioni, piè di pagina e titoli.

La funzione `set-paper-size` è trattata nella prossima sezione, Sezione 4.1.2 [Formato carta e ridimensionamento automatico], pagina 553. Le variabili `\paper` che si occupano della formattazione della pagina sono trattate in sezioni successive. Le definizioni di markup relative a intestazioni, piè di pagina e titoli sono trattate in Sezione 3.2.2 [Titoli intestazioni e piè di pagina personalizzati], pagina 506.

La maggior parte delle variabili `\paper` funzionano soltanto in un blocco `\paper`. Le poche che funzionano anche in un blocco `\layout` sono elencate in Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

Se non indicato diversamente, tutte le variabili `\paper` che corrispondono a distanze sulla pagina sono misurate in millimetri, a meno che un'unità di misura diversa non sia specificata dall'utente. Per esempio, la seguente dichiarazione imposta `top-margin` su dieci millimetri:

```
\paper {
  top-margin = 10
}
```

Per impostarla su 0.5 pollici, usare il suffisso di unità `\in` (inch = pollice):

```
\paper {
  top-margin = 0.5\in
}
```

I suffissi di unità disponibili sono `\mm`, `\cm`, `\in` e `\pt`. Queste unità sono semplici valori utili per convertire dai millimetri e sono definite in `ly/paper-defaults-init.ly`. Solo per chiarezza, quando si usano i millimetri, di solito si usa il suffisso `\mm`, anche se non è tecnicamente necessario.

È anche possibile definire i valori di `\paper` con Scheme. L'equivalente Scheme dell'esempio precedente è:

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

Vedi anche

Guida alla notazione: Sezione 4.1.2 [Formato carta e ridimensionamento automatico], pagina 553, Sezione 3.2.2 [Titoli intestazioni e piè di pagina personalizzati], pagina 506, Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

File installati: `ly/paper-defaults-init.ly`.

4.1.2 Formato carta e ridimensionamento automatico

Impostare il formato carta

‘A4’ è il valore predefinito quando non viene impostato esplicitamente alcun formato carta. Esistono due funzioni che permettono di cambiare formato:

```
set-default-paper-size
  #(set-default-paper-size "quarto")
  che deve sempre trovarsi nel livello superiore, e
```

```
set-paper-size
  \paper {
    #(set-paper-size "tabloid")
  }
```

che deve sempre trovarsi in un blocco `\paper`.

La funzione `set-default-paper-size`, se usata nel livello superiore, deve precedere qualsiasi blocco `\paper`. `set-default-paper-size` imposta il formato di tutte le pagine, mentre `set-paper-size` imposta il formato soltanto di quelle pagine a cui è applicato il blocco `\paper`. Per esempio, se il blocco `\paper` si trova all'inizio del file, applicherà il formato a tutte le pagine. Se il blocco `\paper` si trova dentro un blocco `\book`, il formato verrà applicato a quel libro soltanto.

Quando si usa la funzione `set-paper-size`, questa deve essere posta *prima* di qualsiasi altra funzione usata nello stesso blocco `\paper`. Vedi [Ridimensionamento automatico al formato carta], pagina 554.

I formati carta sono definiti in `scm/paper.scm`, e sebbene sia possibile aggiungere formati personalizzati in questo file, tali aggiunte verrebbero sovrascritte da successivi aggiornamenti del software. I formati disponibili sono elencati in Sezione A.5 [Formati carta predefiniti], pagina 687.

È tuttavia possibile aggiungere un formato personalizzato nel file di input per poi utilizzarlo con `set-default-paper-size` o `set-paper-size`:

```
#(set! paper-alist (cons '("mio formato" . (cons (* 15 in) (* 3 in))) paper-alist))

\paper {
  #(set-paper-size "mio formato")
}
```

Si può usare qualsiasi unità di misura: in (inch, o pollici), cm (centimetri) e mm (millimetri).

Aggiungendo il simbolo `'landscape` alla funzione del formato, le pagine vengono ruotate di 90 gradi e le linee occupano il maggior spazio a disposizione.

```
#(set-default-paper-size "a6" 'landscape)
```

Appendendo `'landscape` (orizzontale) al nome del formato, è possibile scambiare le dimensioni della carta *senza* ruotare la stampa (come quando si stampa in formato cartolina o si creano dei file grafici da includere invece di un documento indipendente):

```
#(set-default-paper-size "a6landscape")
```

Quando il formato termina con un esplicito `'landscape` (orizzontale) o `'portrait` (verticale), la presenza di un simbolo `'landscape` influisce *solo* sull'orientamento della stampa, non sul formato usato per la formattazione.

Vedi anche

Guida alla notazione: [Ridimensionamento automatico al formato carta], pagina 554, Sezione A.5 [Formati carta predefiniti], pagina 687.

File installati: `scm/paper.scm`.

Ridimensionamento automatico al formato carta

Se il formato viene cambiato con una delle funzioni Scheme (`set-default-paper-size` o `set-paper-size`), i valori di diverse variabili `\paper` sono automaticamente ricalcolati in base alla nuova dimensione. Per aggirare il ridimensionamento automatico di una certa variabile, impostare la variabile dopo aver impostato il formato. Fare attenzione al fatto che il ridimensionamento automatico non viene attivato se si impostano le variabili `paper-height` o `paper-width`, anche se `paper-width` può influenzare altri valori (ma questo è un argomento diverso dal ridimensionamento ed è trattato in seguito). Le funzioni `set-default-paper-size` e `set-paper-size` sono descritte in [Impostare il formato carta], pagina 553.

Le dimensioni verticali interessate dal ridimensionamento automatico sono `top-margin` e `bottom-margin` (vedi Sezione 4.1.3 [Variabili `\paper` della spaziatura verticale fissa], pagina 555). Le dimensioni orizzontali interessate dal ridimensionamento automatico sono `left-margin`, `right-margin`, `inner-margin`, `outer-margin`, `binding-offset`, `indent` e `short-indent` (vedi Sezione 4.1.5 [Variabili `\paper` della spaziatura orizzontale], pagina 557).

I valori predefiniti di queste dimensioni sono impostati in `ly/paper-defaults-init.ly` e salvati in variabili interne chiamate `top-margin-default`, `bottom-margin-default`, etc. Questi valori si riferiscono al formato predefinito `a4`. Per riferimento, nel formato `a4` il valore di `paper-height` è `297\mm` e quello di `paper-width` è `210\mm`.

Vedi anche

Guida alla notazione: Sezione 4.1.3 [Variabili `\paper` della spaziatura verticale fissa], pagina 555, Sezione 4.1.5 [Variabili `\paper` della spaziatura orizzontale], pagina 557.

File installati: `ly/paper-defaults-init.ly`, `scm/paper.scm`.

4.1.3 Variabili `\paper` della spaziatura verticale fissa

Nota: Alcune dimensioni definite nel blocco `\paper` sono ridimensionate automaticamente in base al formato carta, portando a possibili risultati imprevisti. Vedi [Ridimensionamento automatico al formato carta], pagina 554.

I valori predefiniti (prima del ridimensionamento) sono definiti in `ly/paper-defaults-init.ly`.

`paper-height`

L'altezza della pagina (valore predefinito: non impostata). Il ridimensionamento automatico di alcune dimensioni verticali non è influenzato da questa.

`top-margin`

Il margine tra il punto più alto della pagina e il punto più alto dell'area di stampa. Se il formato viene modificato, il valore predefinito di questa dimensione viene ridimensionato di conseguenza.

`bottom-margin`

Il margine tra il punto più basso dell'area di stampa e il punto più basso della pagina. Se il formato viene modificato, il valore predefinito di questa dimensione viene ridimensionato di conseguenza.

`ragged-bottom`

Se impostato su vero, i sistemi avranno la loro naturale spaziatura. Non saranno giustificati, ovvero non saranno né compressi né allungati verticalmente per stare nella pagina.

`ragged-last-bottom`

Se impostato su falso, l'ultima pagina, e l'ultima pagina di ogni sezione creata con un blocco `\bookpart`, saranno giustificate verticalmente come nelle pagine precedenti.

Vedi anche

Guida alla notazione: [Ridimensionamento automatico al formato carta], pagina 554.

File installati: `ly/paper-defaults-init.ly`.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Problemi noti e avvertimenti

I titoli (definiti nel blocco `\header`) sono considerati un sistema, dunque `ragged-bottom` e `ragged-last-bottom` aggiungeranno spazio tra i titoli e il primo sistema della partitura.

Formati carta definiti esplicitamente sovrascriveranno qualsiasi impostazione di margine superiore o inferiore definito dall'utente.

4.1.4 Variabili `\paper` della spaziatura verticale flessibile

Nella maggior parte dei casi è preferibile che le distanze verticali tra certi elementi (come margini, titoli, sistemi e partiture separate) siano flessibili, così che siano ben compresse o allungate a

seconda della situazione. Sono disponibili diverse variabili `\paper` (elencate sotto) per regolare il comportamento di allungamento di queste dimensioni.

Nota bene che le variabili `\paper` trattate in questa sezione non regolano la spaziatura dei righe dei singoli sistemi. La spaziatura interna ai sistemi è controllata dalle proprietà del `grob`, solitamente inserite in un blocco `\score` o `\layout` e non in un blocco `\paper`. Vedi Sezione 4.4.1 [Spaziatura verticale flessibile all'interno dei sistemi], pagina 574.

Struttura delle liste associative flessibili della spaziatura verticale

Ognuna delle variabili `\paper` flessibili della spaziatura verticale è una lista associativa (`alist`) composta da quattro *elementi*:

- `basic-distance` – la distanza verticale, misurata in spazi rigo, tra i *punti di riferimento* dei due elementi, quando non si verificano collisioni e non sono attivi allungamenti o compressioni. Il punto di riferimento di un markup (un titolo o un markup di un livello superiore) è il suo punto più alto, mentre quello di un sistema è il centro verticale del più vicino `StaffSymbol`, anche se una linea diversa da un rigo (come un contesto `Lyrics`) si trova nel mezzo. Valori di `basic-distance` inferiori a quelli di `padding` o `minimum-distance` non hanno senso, dato che la distanza risultante non sarà mai inferiore a quella definita per `padding` o `minimum-distance`.
- `minimum-distance` – la minima distanza verticale consentita, misurata in spazi rigo, tra i punti di riferimento dei due elementi, quando la compressione è attiva. Valori di `minimum-distance` inferiori a quelli di `padding` non hanno senso, perché la distanza risultante non sarà mai inferiore a quella definita per `padding`.
- `padding` – la quantità minima di spazio verticale libero tra i profili (*skyline*) dei due elementi, misurata in spazi rigo.
- `stretchability` – una misura, priva di unità di misura, della relativa propensione della dimensione a allungarsi. Se pari a zero, la distanza non si allungherà (a meno che non ci siano delle collisioni). Se positiva, il significato del valore di `stretchability` di una certa dimensione sta nella sua relazione con i valori di `stretchability` delle altre dimensioni. Per esempio, se una dimensione ha un valore di `stretchability` doppio rispetto a un'altra, si allungherà il doppio. I valori devono essere non negativi e finiti. Il valore `+inf.0` causa un errore di programmazione e viene ingorato, ma si può usare `1.0e7` per ottenere un'elasticità allungabile quasi all'infinito. Se non impostata, il valore predefinito equivale a quello di `basic-distance`. Nota che la propensione della dimensione a *comprimersi* non può essere impostata direttamente dall'utente ed è uguale a $(\text{basic-distance} - \text{minimum-distance})$.

Se in una pagina lo spazio verticale non è giustificato (ovvero con `ragged-bottom` impostato su vero) la distanza risultante è data dal valore più grande assegnato a una di queste tre proprietà:

- `basic-distance`,
- `minimum-distance` e
- `padding` con l'aggiunta della più piccola distanza necessaria a eliminare collisioni.

In caso di partiture con molte pagine in cui l'ultima pagina non è giustificata (ovvero con `ragged-last-bottom` impostato su vero), l'ultima pagina usa la stessa spaziatura della pagina precedente, purché ci sia abbastanza spazio.

I metodi specifici per modificare le liste associative (`alist`) sono trattati in Sezione 5.3.7 [Modifica delle liste associative], pagina 639. L'esempio seguente illustra i due modi in cui queste possono essere modificate. La prima dichiarazione aggiorna un elemento-valore individualmente, mentre la seconda ridefinisce completamente la variabile:

```
\paper {
  system-system-spacing.basic-distance = #8
  score-system-spacing =
```

```

#'( (basic-distance . 12)
    (minimum-distance . 6)
    (padding . 1)
    (stretchability . 12))
}

```

Elenco delle variabili `\paper` flessibili della spaziatura verticale

I nomi di queste variabili hanno il formato *superiore-inferiore-spacing*, dove *superiore* e *inferiore* indicano gli elementi di cui stabilire la distanza. Ogni distanza viene misurata tra i punti di riferimento dei due elementi (vedi la descrizione precedente della struttura della lista associativa). Nei nomi di queste variabili il termine ‘markup’ si riferisce sia ai *titoli* (`bookTitleMarkup` o `scoreTitleMarkup`) sia ai *markup di livello superiore* (vedi Sezione 3.1.5 [Struttura del file], pagina 497). Tutte le distanze sono misurate in spazi rigo.

Le impostazioni predefinite sono definite in `ly/paper-defaults-init.ly`.

`markup-system-spacing`

la distanza tra il testo (titolo o markup di livello superiore) e il sistema che lo segue.

`score-markup-spacing`

la distanza tra l’ultimo sistema di un brano e il testo (titolo o markup di livello superiore) che lo segue.

`score-system-spacing`

la distanza tra l’ultimo sistema di un brano e il primo sistema del brano successivo, quando non c’è alcun testo (titolo o markup di livello superiore) tra di loro.

`system-system-spacing`

la distanza tra due sistemi di uno stesso brano.

`markup-markup-spacing`

la distanza tra due testi (titoli o markup di livello superiore).

`last-bottom-spacing`

la distanza tra l’ultimo sistema o l’ultimo markup di livello superiore di una pagina e la fine dello spazio stampabile (ovvero la parte superiore del margine inferiore).

`top-system-spacing`

la distanza tra l’inizio dello spazio stampabile (ovvero la parte inferiore del margine superiore) e il primo sistema di una pagina, quando non c’è alcun testo (titolo o markup di livello superiore) tra di loro.

`top-markup-spacing`

la distanza tra l’inizio dello spazio stampabile (ovvero la parte inferiore del margine superiore) e il primo testo (titolo o markup di livello superiore) di una pagina, quando non c’è alcun sistema tra i due.

Vedi anche

Guida alla notazione: Sezione 4.4.1 [Spaziatura verticale flessibile all’interno dei sistemi], pagina 574.

File installati: `ly/paper-defaults-init.ly`.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.1.5 Variabili `\paper` della spaziatura orizzontale

Nota: Alcune dimensioni in `\paper` sono ridimensionate automaticamente in base al formato carta, producendo talvolta un risultato inatteso. Vedi [Ridimensionamento automatico al formato carta], pagina 554.

Variabili `\paper` per larghezze e margini

I valori predefiniti (prima del ridimensionamento) non elencati qui si trovano in `ly/paper-defaults-init.ly`.

`paper-width`

Larghezza della pagina (non impostata inizialmente). `paper-width`, pur non avendo effetto sul ridimensionamento automatico di alcune dimensioni orizzontali, influenza la variabile `line-width`. Se sono impostate sia `paper-width` che `line-width`, saranno modificate anche `left-margin` e `right-margin`. Vedi anche `check-consistency`.

`line-width`

Se specificata in un blocco `\paper` definisce la lunghezza orizzontale disponibile per i righi musicali nei sistemi non indentati. Se non specificata, la larghezza della linea (`line-width`) del foglio è determinata dalla sottrazione (`paper-width` - `left-margin` - `right-margin`). Se `line-width` è specificata nel blocco `\paper` e entrambi i margini, `left-margin` e `right-margin`, non lo sono, questi saranno aggiornati in modo da centrare i sistemi sulla pagina automaticamente. Vedi anche `check-consistency`.

Le variabili `line-width` per i singoli brani possono essere specificate nel blocco `\layout` di ogni brano, ovvero dentro un blocco `\score`. Questi valori regolano la larghezza delle linee prodotte brano per brano. Se `line-width` non viene specificata per un brano, il suo valore predefinito coincide con la larghezza della linea del foglio, ovvero col valore della variabile `line-width` del blocco `\paper`. Impostando `line-width` per un brano in particolare, ovvero all'interno di un blocco `\score`, i margini del foglio non vengono modificati. Le linee del rigo, la cui lunghezza è determinata dalla variabile `line-width` del brano, sono allineate a sinistra all'interno dello spazio definito dalla variabile `line-width` del foglio. Se le variabili `line-width` del brano e del foglio sono uguali, le linee del rigo si estenderanno esattamente dal margine sinistro al margine destro, ma se quella del brano è maggiore di quella del foglio le linee del rigo andranno oltre il margine destro.

`left-margin`

Margine tra il bordo sinistro della pagina e l'inizio delle linee del rigo nei sistemi non indentati. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Se `left-margin` non è impostato, e sono impostati sia `line-width` che `right-margin`, allora `left-margin` equivale alla differenza (`paper-width` - `line-width` - `right-margin`). Se è impostata soltanto `line-width`, entrambi i margini sono impostati su $((\text{paper-width} - \text{line-width}) / 2)$, e di conseguenza i sistemi sono centrati sulla pagina. Vedi anche `check-consistency`.

`right-margin`

Margine tra il bordo destro della pagina e la fine delle linee del rigo in sistemi giustificati. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Se `right-margin` non è impostato, e sono impostati sia `line-width` che `left-margin`, allora `right-margin` equivale a (`paper-width` - `line-width` - `left-margin`). Se è impostata soltanto `line-width`, entrambi i margini sono impostati su

$((\text{paper-width} - \text{line-width}) / 2)$, e di conseguenza i sistemi sono centrati sulla pagina. Vedi anche `check-consistency`.

`check-consistency`

Se impostata su vero (valore predefinito), appare un avvertimento nella console se la somma di `left-margin`, `line-width` e `right-margin` non corrisponde esattamente a `paper-width`, e a ciascuna di queste variabili (eccetto `paper-width`) sarà assegnato il suo valore predefinito (ridimensionato in base al formato, se richiesto). Se impostato su falso, qualsiasi incongruenza viene ignorata e i sistemi potrebbero andare oltre il bordo della pagina.

`ragged-right`

Se impostato su vero, i sistemi non occuperanno interamente la larghezza della linea, ovvero non saranno giustificati. Termineranno invece alla loro lunghezza orizzontale naturale. Valori predefiniti: `#t` (true = vero) per i brani con un unico sistema, e `#f` (false = falso) per i brani con due o più sistemi. Questa variabile può essere impostata anche in un blocco `\layout`.

`ragged-last`

Se impostata su vero, l'ultimo sistema del brano non occuperà interamente la larghezza della linea, ovvero non sarà giustificato. Terminerà invece alla sua lunghezza orizzontale naturale. Valore predefinito: `#f` (false = falso). Questa variabile può essere impostata anche in un blocco `\layout`.

Vedi anche

Guida alla notazione: [Ridimensionamento automatico al formato carta], pagina 554.

File installati: `ly/paper-defaults-init.ly`.

Problemi noti e avvertimenti

Formati carta definiti esplicitamente sovrascriveranno qualsiasi margine destro o sinistro definito dall'utente.

Variabili `\paper` per la modalità due pagine per foglio

I valori predefiniti (prima del ridimensionamento) sono definiti in `ly/paper-defaults-init.ly`.

`two-sided`

Se impostata su vero, vengono usati `inner-margin`, `outer-margin` e `binding-offset` per determinare i margini a seconda che il numero di pagina sia dispari o pari. Questa variabile sovrascrive `left-margin` e `right-margin`.

`inner-margin`

Margine del lato interno di tutte le pagine se queste fanno parte di un libro. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Funziona soltanto con `two-sided` impostato su vero.

`outer-margin`

È il margine del lato esterno di tutte le pagine se queste fanno parte di un libro. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Funziona soltanto con `two-sided` impostato su vero.

`binding-offset`

Quantità di spazio da aggiungere a `inner-margin` per assicurarsi che niente sia nascosto dalla rilegatura. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Funziona soltanto con `two-sided` impostato su vero.

Vedi anche

Guida alla notazione: [Ridimensionamento automatico al formato carta], pagina 554.

File installati: `ly/paper-defaults-init.ly`.

Variabili `\paper` per spostamenti e indentazioni

I valori predefiniti (prima del ridimensionamento) non elencati qui sono visibili in `ly/paper-defaults-init.ly`.

`horizontal-shift`

È la quantità di spazio di cui spostare a destra tutti i sistemi (inclusi i titoli e i separatori dei sistemi). Valore predefinito: `0.0\mm`.

`indent`

Livello di indentazione del primo sistema di un brano. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Lo spazio compreso in `line-width` disponibile per il primo sistema è ridotto di questa quantità. `indent` può essere specificato anche in blocchi `\layout` per impostare l'indentazione brano per brano.

`short-indent`

Livello di indentazione di tutti i sistemi di un brano eccetto il primo sistema. Se il formato è modificato, il valore predefinito di questa dimensione viene ridimensionato proporzionalmente. Lo spazio compreso in `line-width` disponibile per tutti i sistemi tranne il primo è ridotto di questa quantità. `short-indent` può essere specificato anche in blocchi `\layout` per impostare le brevi indentazioni brano per brano.

Vedi anche

Guida alla notazione: [Ridimensionamento automatico al formato carta], pagina 554.

File installati: `ly/paper-defaults-init.ly`.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.1.6 Altre variabili di `\paper`

Variabili di `\paper` per l'interruzione di linea

`max-systems-per-page`

Il numero massimo di sistemi che saranno disposti in una pagina. Attualmente ciò è supportato soltanto dall'algoritmo `ly:optimal-breaking`. Valore predefinito: non impostato.

`min-systems-per-page`

Il numero minimo di sistemi che saranno disposti in una pagina. Ciò potrebbe riempire troppo le pagine, se il numero è troppo grande. Attualmente ciò è supportato soltanto dall'algoritmo `ly:optimal-breaking`. Valore predefinito: non impostato.

`systems-per-page`

Il numero di sistemi da disporre in una pagina. Attualmente ciò è supportato soltanto dall'algoritmo `ly:optimal-breaking`. Valore predefinito: non impostato.

`system-count`

Il numero di sistemi da usare per un brano. Valore predefinito: non impostato. Questa variabile può essere impostata anche in un blocco `\layout`.

Vedi anche

Guida alla notazione: Sezione 4.3.1 [Interruzioni di linea], pagina 567.

Variabili di `\paper` per l'interruzione di pagina

I valori predefiniti non elencati qui sono visibili in `ly/paper-defaults-init.ly`

`page-breaking`

L'algoritmo di interruzione di pagina da usare. Le opzioni sono `ly:minimal-breaking`, `ly:page-turn-breaking`, `ly:one-page-breaking`, `ly:one-line-breaking`, `ly:one-line-auto-height-breaking` e `ly:optimal-breaking` (predefinito).

`page-breaking-system-system-spacing`

Induce il sistema di interruzione di pagina a credere che `system-system-spacing` sia impostato in modo diverso dal modo in cui è impostato realmente. Per esempio, se `page-breaking-system-system-spacing.padding` è impostato su un valore molto più grande di `system-system-spacing.padding`, il sistema di interruzione di pagina disporrà meno sistemi su ciascuna pagina. Valore predefinito: non impostato.

`page-count`

Il numero di pagine da usare per un brano, non impostato.

Le variabili seguenti sono attive soltanto quando `page-breaking` è impostato su `ly:page-turn-breaking`. Le interruzioni di pagina vengono quindi scelte per minimizzare il numero di voltate di pagina. Dato che le voltate di pagina si rendono necessarie quando si passa da una pagina con numero dispari a una con numero pari, verrà solitamente favorita una formattazione in cui l'ultima pagina abbia un numero dispari. I punti preferiti per le voltate di pagina possono essere indicati manualmente col comando `\allowPageTurn` o automaticamente includendo l'incisore `Page_turn_engraver` (vedi [Voltata di pagina ottimale], pagina 572).

Se le scelte disponibili per avere voltate di pagina adeguate sono insufficienti, LilyPond potrebbe inserire una pagina vuota all'interno di un brano o tra i brani (se ce ne sono almeno due), oppure potrebbe terminare un brano su una pagina pari. Aumentando i valori delle seguenti tre variabili diminuisce la probabilità che queste entrino in azione.

I valori sono penalità: maggiore il valore, minore la probabilità che si verifichi l'azione associata rispetto alle altre scelte.

`blank-page-penalty`

La penalità per avere una pagina vuota nel mezzo di un brano. Se il valore di `blank-page-penalty` è grande ed è selezionato `ly:page-turn-breaking`, sarà meno probabile che LilyPond inserisca una pagina nel mezzo del brano. Aggiungerà invece maggior spazio tra i sistemi per occupare la pagina vuota e quella successiva. Valore predefinito: 5.

`blank-last-page-penalty`

La penalità per terminare il brano su una pagina pari. Se il valore di `blank-last-page-penalty` è grande ed è selezionato `ly:page-turn-breaking`, sarà meno probabile che LilyPond produca una partitura in cui l'ultima pagina sia pari. Regolerà invece la spaziatura in modo da usare una pagina in più o in meno. Valore predefinito: 0.

`blank-after-score-page-penalty`

La penalità per avere una pagina vuota dopo la fine di un brano e prima di quello successivo. Il valore predefinito è inferiore a `blank-page-penalty`, in modo che siano inserite preferibilmente pagine vuote al termine di un brano piuttosto che nel mezzo di un brano. Valore predefinito: 2.

Vedi anche

Guida alla notazione: Sezione 4.3.2 [Interruzioni di pagina], pagina 570, [Interruzione di pagina ottimale], pagina 571, [Voltata di pagina ottimale], pagina 572, [Interruzione di pagina

minimale], pagina 572, [Interruzione di pagina di una pagina], pagina 572, [Interruzione di pagina su una linea], pagina 572, [Interruzione di pagina su una linea con altezza automatica], pagina 572.

File installati: `ly/paper-defaults-init.ly`.

Variabili di `\paper` per la numerazione delle pagine

I valori predefiniti non elencati qui sono visibili in `ly/paper-defaults-init.ly`

`auto-first-page-number`

L'algoritmo di interruzione di pagina si comporta diversamente a seconda che il numero della prima pagina sia dispari o pari. Se questa variabile è impostata su vero, l'algoritmo di interruzione di pagina deciderà se iniziare con un numero dispari o un numero pari. Quindi il numero della prima pagina resterà lo stesso o aumenterà di uno. Valore predefinito: `#f` (falso).

`first-page-number`

Il valore del numero di pagina della prima pagina.

`print-first-page-number`

Se impostato su vero, appare il numero di pagina sulla prima pagina.

`print-page-number`

Se impostato su falso, non appariranno i numeri di pagina.

`page-number-type`

Il tipo di numero usato per numerare le pagine. Le opzioni possibili sono `roman-lower`, `roman-upper` e `arabic`. Valore predefinito: `'arabic`.

Vedi anche

File installati: `ly/paper-defaults-init.ly`.

Problemi noti e avvertimenti

I numeri di pagina dispari sono sempre sulla destra. Se la musica deve iniziare a pagina 1, ci deve essere una pagina vuota sulla seconda di copertina in modo che pagina 1 sia sul lato destro.

Svariate variabili di `\paper`

`page-spacing-weight`

L'importanza della spaziatura (verticale) e della linea (orizzontale) della pagina. Valori più grandi renderanno la spaziatura della pagina più importante. Valore predefinito: 10.

`print-all-headers`

Se impostato su vero, appariranno nell'output tutte le intestazioni di ciascun blocco `\score`. Di norma appaiono soltanto le variabili `piece` e `opus`. Valore predefinito: `#f`.

`system-separator-markup`

Un oggetto markup inserito tra i sistemi, spesso usato per le partiture orchestrali. Valore predefinito: non impostato. È disponibile il comando `markup \slashSeparator`, definito in `ly/titling-init.ly`, che fornisce un ragionevole valore predefinito, per esempio:

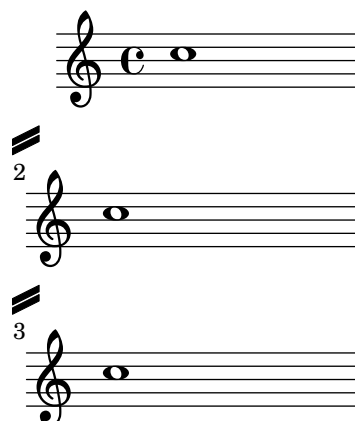
```
#(set-default-paper-size "a8")
```

```
\book {
```

```

\paper {
  system-separator-markup = \slashSeparator
}
\header {
  tagline = ##f
}
\score {
  \relative { c''1 \break c1 \break c1 }
}

```



Vedi anche

File installati: ly/titling-init.ly.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Problemi noti e avvertimenti

L’ intestazione di pagina predefinita pone sulla stessa linea il numero di pagina e il campo `instrument` del blocco `\header`.

4.2 Formattazione della partitura

Questa sezione tratta le opzioni di formattazione della partitura disponibili nel blocco `\layout`.

4.2.1 Il blocco `\layout`

Mentre il blocco `\paper` contiene le impostazioni relative alla formattazione della pagina per l’intero documento, il blocco `\layout` contiene impostazioni di formattazione specifica di un brano. Per impostare globalmente le opzioni di formattazione del brano, inserirle in un blocco `\layout` del livello superiore. Per impostare le impostazioni di formattazione per un singolo brano, inserirle in un blocco `\layout` racchiuso in un blocco `\score`, dopo la musica. Le impostazioni che possono apparire in un blocco `\layout` includono:

- la funzione `Scheme layout-set-staff-size`,
- le modifiche di contesto nei blocchi `\context` e
- le variabili `\paper` che agiscono sulla formattazione del brano.

La funzione `layout-set-staff-size` è spiegata nella prossima sezione, Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565. Le modifiche di contesto sono trattate in un capitolo separato; vedi Sezione 5.1.4 [Modifica dei componenti aggiuntivi di un contesto], pagina 613, e Sezione 5.1.5 [Modifica delle impostazioni predefinite di un contesto], pagina 615.

Le variabili `\paper` che possono apparire in un blocco `\layout`, con valori predefiniti presi dal blocco `\paper`, sono:

- `line-width`, `ragged-right` e `ragged-last` (vedi [Variabili `\paper` per larghezze e margini], pagina 558)
- `indent` e `short-indent` (vedi [Variabili `\paper` per spostamenti e indentazioni], pagina 560)
- `system-count` (vedi [Variabili di `\paper` per l'interruzione di linea], pagina 560)

Ecco un esempio di blocco `\layout`:

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGroup.staff-staff-spacing.basic-distance = #8
  }
  \context {
    \Voice
    \override TextScript.padding = #1
    \override Glissando.thickness = #3
  }
}
```

Si possono inserire molteplici blocchi `\layout` come espressioni di livello superiore. Ciò può essere utile, per esempio, se impostazioni diverse sono salvate in file separati e incluse facoltativamente. Internamente, quando un blocco `\layout` viene elaborato, viene fatta una copia della configurazione del blocco `\layout` corrente, poi qualsiasi modifica apportata nel blocco viene applicata e il risultato viene salvato come la nuova configurazione corrente. Dalla prospettiva dell'utente, i blocchi `\layout` sono combinati, ma in situazioni di conflitto (ovvero quando viene modificata la stessa proprietà in blocchi diversi) hanno precedenza le definizioni più recenti.

Per esempio, se questo blocco

```
\layout {
  \context {
    \Voice
    \override TextScript.color = #magenta
    \override Glissando.thickness = #1.5
  }
}
```

viene posto dopo quello dell'esempio precedente, le sovrascritture di `'padding` e `'color` per `TextScript` sono combinate, mentre quella più recente di `'thickness` per `Glissando` sostituisce (o nasconde) quella precedente.

I blocchi `\layout` possono essere assegnati a delle variabili per poterli riutilizzare successivamente, ma il modo in cui ciò funziona è leggermente ma significativamente diverso rispetto a scriverle davvero.

Se una variabile è definita in questo modo,

```
layoutVariable = \layout {
  \context {
    \Voice
    \override NoteHead.font-size = #4
  }
}
```

manterrà la configurazione attuale di `\layout` con l'aggiunta della sovrascrittura `NoteHead.font-size`, ma questa combinazione *non* è salvata come la nuova configurazione

corrente. Attenzione al fatto che la “configurazione corrente” viene letta quando la variabile è definita e non quando viene usata, dunque il contenuto della variabile dipende dalla sua posizione nel file di input.

La variabile può quindi essere usata all'interno di un altro blocco `\layout`, per esempio:

```
\layout {
  \layoutVariable
  \context {
    \Voice
    \override NoteHead.color = #red
  }
}
```

Un blocco `\layout` che contiene una variabile, come nell'esempio precedente, *non* copia la configurazione corrente bensì usa il contenuto di `\layoutVariable` come configurazione di base per altre aggiunte. Ciò significa che qualsiasi modifica definita tra la definizione della variabile e il momento in cui essa viene usata è perduta.

Se `layoutVariable` è definita (o inclusa con `\include`) subito prima di essere usata, il suo contenuto comprende soltanto la configurazione corrente più le sovrascritture definite al suo interno. Quindi nell'esempio precedente che illustra l'uso di `\layoutVariable` il blocco `\layout` finale conterrebbe:

```
TextScript.padding = #1
TextScript.color = #magenta
Glissando.thickness = #1.5
NoteHead.font-size = #4
NoteHead.color = #red
```

più le sovrascritture `indent` e `StaffGrouper`.

Ma se la variabile fosse già stata definita prima del primo blocco `\layout`, la configurazione corrente conterrebbe soltanto

```
NoteHead.font-size = #4 % (scritta nella definizione della variabile)
NoteHead.color = #red % (aggiunta dopo l'uso della variabile)
```

Se ben organizzate, le variabili `\layout` possono essere un valido strumento per strutturare le formattazioni dei sorgenti, e anche per ripristinare la configurazione di `\layout` a uno stato conosciuto.

Vedi anche

Guida alla notazione: Sezione 5.1.5 [Modifica delle impostazioni predefinite di un contesto], pagina 615.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.2.2 Impostare la dimensione del rigo

La dimensione del rigo (in inglese **staff size**) è 20 punti, che corrispondono a un'altezza del rigo di 7.03mm (un punto è uguale a 100/7227 di un pollice o a 2540/7227 mm). La dimensione del rigo può essere modificata in tre modi:

1. Per impostare la dimensione del rigo globalmente per tutti i brani di un file (o di un blocco `\book`, per essere precisi), usare `set-global-staff-size`:

```
 #(set-global-staff-size 14)
```

L'esempio precedente imposta la dimensione del rigo globale a 14pt (4.92mm) e ridimensiona proporzionalmente tutti i tipi di carattere.

2. Per impostare la dimensione del rigo di una singola partitura in un libro, usare `layout-set-staff-size` all'interno del blocco `\layout` di quel brano:

```
\score {
  ...
  \layout {
    #(layout-set-staff-size 14)
  }
}
```

3. Per impostare la dimensione del rigo di un singolo rigo di un sistema, usare il comando `\magnifyStaff`. Per esempio, le partiture di musica da camera incise in modo tradizionale spesso usavano righe per pianoforte di 7mm mentre gli altri righe erano solitamente tra 3/5 e 5/7 più grandi (tra 60% e 71%). Per ottenere la proporzione 5/7, usare:

```
\score {
  <<
    \new Staff \with {
      \magnifyStaff #5/7
    } { ... }
    \new PianoStaff { ... }
  >>
}
```

Se si desidera una dimensione del tipo di carattere (`fontSize`) ben precisa, si può usare la seguente forma:

```
\score {
  <<
    \new Staff \with {
      \magnifyStaff #(magstep -3)
    } { ... }
    \new PianoStaff { ... }
  >>
}
```

Per emulare l'aspetto delle partiture incise coi metodi tradizionali, è meglio evitare di ridurre lo spessore delle linee del rigo.

Corpo automatico dei tipi di carattere a dimensioni diverse

Il font *Emmentaler* fornisce l'insieme di glifi musicali *Feta* in otto dimensioni diverse, ciascuna calibrata per una specifica dimensione del rigo. Più piccola è la dimensione del glifo, più “grosso” diventa, per abbinarsi alle linee del rigo più spesse. Le dimensioni dei glifi consigliate sono elencate nella seguente tabella:

nome del tipo di carattere	altezza del rigo (pt)	altezza del rigo (mm)	uso
feta11	11.22	3.9	partiture tascabili
feta13	12.60	4.4	
feta14	14.14	5.0	
feta16	15.87	5.6	
feta18	17.82	6.3	canzonieri
feta20	20	7.0	parti standard
feta23	22.45	7.9	
feta26	25.2	8.9	

Vedi anche

Guida alla notazione: Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565, Sezione A.8 [Il font Emmentaler], pagina 693.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.3 Interruzioni

4.3.1 Interruzioni di linea

Le interruzioni di linea di solito sono determinate automaticamente. Sono decise in modo che le linee non sembrino né fitte né troppo spaziate, e che le linee consecutive abbiano una densità simile.

Per forzare manualmente un'interruzione di linea dopo una stanghetta, usare il comando `\break`:

```
\relative c' ' {
  c4 c c c | \break
  c4 c c c |
}
```



Per impostazione predefinita, un comando `\break` inserito nel mezzo di una misura viene ignorato (e apparirà un messaggio di avvertimento durante la compilazione del file LilyPond). L'aggiunta di una stanghetta invisibile – `\bar ""` – prima del comando `\break` forzerà l'interruzione di linea:

```
\relative c' ' {
  c4 c c
  \bar "" \break
  c |
  c4 c c c |
}
```



Un comando `\break` successivo a una stanghetta viene ignorato se la misura precedente termina nel mezzo di una nota (per esempio, quando un gruppo irregolare inizia in una misura e termina in un'altra). In questo caso conviene togliere l'incisore `Forbid_line_break_engraver` dal contesto `Voice` e usare una costruzione musicale simultanea inserendo il `\break` nel punto giusto della seconda voce:

```
\new Voice \with {
  \remove Forbid_line_break_engraver
```

```

} \relative {
  <<
    { c' '2. \tuplet 3/2 { c4 c c } c2. | }
    { s1 | \break s1 | }
  >>
}

```



Analogamente, le interruzioni di linea vengono ignorate quando le travature oltrepassano una stanghetta. Tale comportamento predefinito può essere modificato col comando `\override Beam.breakable = ##t`:

```

\relative c' {
  \override Beam.breakable = ##t
  c2. c8[ c | \break
  c8 c] c2. |
}

```



Il comando `\noBreak` impedisce un'interruzione di linea sulla stanghetta in cui è inserito.

In una partitura, l'interruzione di linea automatica è vietata per la musica compresa tra i comandi `\autoLineBreaksOff` e `\autoLineBreaksOn`. Per impedire anche le interruzioni di pagina, usare i comandi `\autoBreaksOff` e `\autoBreaksOn`. Le interruzioni manuali non sono interessate da questi comandi. Nota che bloccare le interruzioni di linea automatiche potrebbe far andare la musica oltre il margine destro se questa non può essere contenuta in una linea.

Le interruzioni di linea automatiche (ma non le interruzioni di pagina) possono essere abilitate per singole stanghetta usando `\once \autoLineBreaksOn` all'inizio di una stanghetta. Ciò identifica un'interruzione di linea permessa, invece che forzata.

Le impostazioni fondamentali che influenzano la spaziatura della linea sono `indent` e `line-width`, impostate nel blocco `\layout`: regolano l'indentazione della prima linea e la lunghezza delle linee.

Se `ragged-right` è impostato su `vero` nel blocco `\layout`, allora i sistemi terminano alla loro naturale lunghezza orizzontale, invece di essere allungati orizzontalmente per riempire l'intera linea. Ciò è utile per brevi frammenti e per verificare quanto è stretta la spaziatura naturale.

L'opzione `ragged-last` è simile a `ragged-right`, ma agisce soltanto sull'ultima linea del brano.

```

\layout {
  indent = 0\mm
}

```

```

    line-width = 150\mm
    ragged-last = ##t
}

```

Per inserire interruzioni di linea a intervalli regolari usare `\break` separato da pause spaziatrici e ripetuto con `\repeat`. Per esempio, per interrompere le seguenti 28 misure (considerando un tempo di 4/4) esattamente ogni 4 misure, usare:

```

<<
\repeat unfold 7 {
    s1 \noBreak s1 \noBreak
    s1 \noBreak s1 \break
}
{ la vera musica... }
>>

```

Comandi predefiniti

`\break`, `\noBreak`, `\autoBreaksOff`, `\autoBreaksOn`, `\autoLineBreaksOff`, `\autoLineBreaksOn`.

Frammenti di codice selezionati

Usare una voce apposita per le interruzioni

Spesso è più facile gestire l'informazione sulle interruzioni di linea e di pagina tenendola separata dalla musica grazie a un'ulteriore voce che contenga solo pause spaziatrici e i comandi `\break`, `\pageBreak` e altre informazioni di formattazione.

Questo modello diventa utile specialmente quando si modifica `line-break-system-details` e altre utili ma lunghe proprietà di `NonMusicalPaperColumnGrob`.

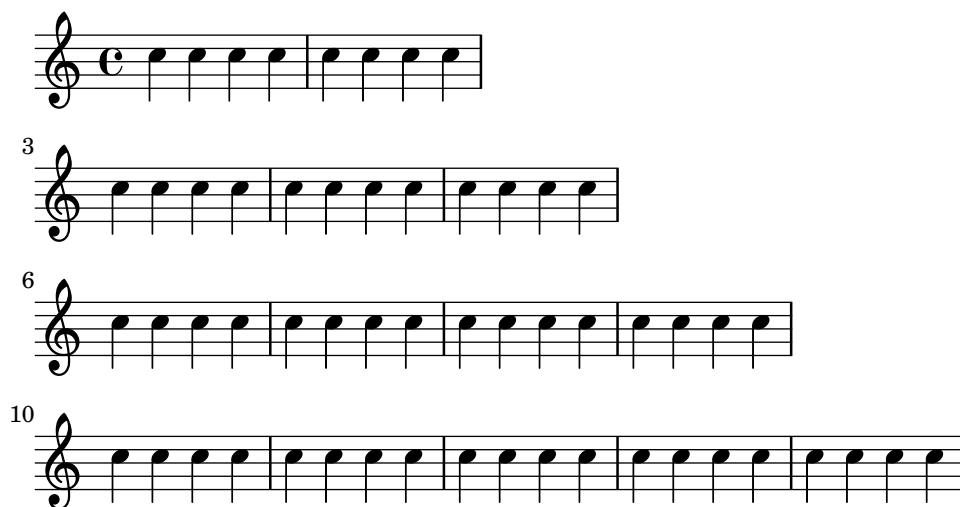
```

music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1*2 \break
      s1*3 \break
      s1*4 \break
      s1*5 \break
    }
    \new Voice {
      \repeat unfold 2 { \music }
      \repeat unfold 3 { \music }
      \repeat unfold 4 { \music }
      \repeat unfold 5 { \music }
    }
  }
}

\paper {
  indent = 0
  line-width = 140\mm
  ragged-right = ##t
}

```



Vedi anche

Guida alla notazione: [Variabili di `\paper` per l'interruzione di linea], pagina 560, Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “LineBreakEvent” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

L'inserimento dei comandi `\autoLineBreaksOff` o `\autoBreaksOff` prima della musica produrrà dei messaggi di errore. Inserire sempre questi comandi dopo la musica.

4.3.2 Interruzioni di pagina

Questa sezione descrive i diversi metodi di interruzione di pagina e spiega come modificarli.

Interruzione di pagina manuale

L'interruzione di pagina predefinita può essere sovrascritta con i comandi `\pageBreak` o `\noPageBreak`. Questi comandi, analoghi a `\break` e `\noBreak`, devono essere inseriti dopo una stanghetta e forzano o proibiscono l'interruzione di pagina in quel punto. Ovviamente il comando `\pageBreak` forza anche un'interruzione di linea.

I comandi `\pageBreak` e `\noPageBreak` possono essere inseriti anche nel livello superiore, tra le partiture e i testi (nel blocco markup) di livello superiore.

In un brano, le interruzioni di pagina automatiche sono vietate per la musica compresa tra i comandi `\autoPageBreaksOff` e `\autoPageBreaksOn`. Le interruzioni di pagina manuali non sono interessate da questi comandi.

Esistono impostazioni analoghe a `ragged-right` e `ragged-last` che hanno lo stesso effetto sulla spaziatura verticale. Se `ragged-bottom` è impostato su `#t`, i sistemi non saranno giustificati verticalmente. Quando `ragged-last-bottom` è impostato su `#t` (valore predefinito), è permesso dello spazio vuoto in fondo all'ultima pagina (o in fondo all'ultima pagina di ciascun `\bookpart`). Vedi Sezione 4.1.3 [Variabili `\paper` della spaziatura verticale fissa], pagina 555.

Le interruzioni di pagina sono calcolate dalla funzione `page-breaking`. LilyPond fornisce vari algoritmi per calcolare le interruzioni di pagina, tra cui `ly:optimal-breaking`, `ly:page-turn-breaking` e `ly:minimal-breaking`. Quello predefinito è `ly:optimal-breaking`, ma il valore può essere modificato nel blocco `\paper`:

```
\paper {
```

```

    page-breaking = #ly:page-turn-breaking
}

```

Quando un libro ha molte partiture e pagine, il problema delle interruzioni di pagina potrebbe essere difficile da risolvere e richiedere lunghi tempi di elaborazione e molta memoria. Per semplificare il processo di interruzione delle pagine, si usano i blocchi `\bookpart` per dividere il libro in varie parti: in questo modo l'interruzione di pagina si verifica separatamente in ciascuna parte. Si possono anche usare algoritmi di interruzione di pagina diversi per le diverse parti del libro.

```

\bookpart {
  \header {
    subtitle = "Prefazione"
  }
  \paper {
    %% In una parte contenente soprattutto testo,
    %% ly:minimal-breaking potrebbe essere preferibile
    page-breaking = #ly:minimal-breaking
  }
  \markup { ... }
  ...
}
\bookpart {
  %% In questa parte, contenente musica, si usa l'algoritmo di
  %% interruzione di pagina ottimale.
  \header {
    subtitle = "Primo movimento"
  }
  \score { ... }
  ...
}

```

Comandi predefiniti

`\pageBreak`, `\noPageBreak`, `\autoPageBreaksOn`, `\autoPageBreaksOff`.

Vedi anche

Guida alla notazione: [Variabili di `\paper` per l'interruzione di pagina], pagina 561.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Problemi noti e avvertimenti

Il prefisso `\once` non funziona con i comandi `\autoPageBreaksOn` e `\autoPageBreaksOff`. Se l'interruzione di pagina automatica è disabilitata e poi viene abilitata per permettere un'interruzione di pagina, deve restare attiva per alcune battute (il numero preciso di battute dipende dalla partitura) prima di essere disattivata, altrimenti la possibilità di interrompere la pagina non verrà considerata.

Interruzione di pagina ottimale

La funzione `ly:optimal-breaking` è il metodo predefinito di LilyPond per determinare le interruzioni di pagina. Tenta di individuare un'interruzione di pagina che minimizzi la densità e l'allungamento, sia orizzontalmente che verticalmente. Diversamente da `ly:page-turn-breaking`, non prende in considerazione le voltate di pagina.

Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Interruzione di pagina minimale

La funzione `ly:minimal-breaking` fa dei calcoli minimi per valutare l'interruzione di pagina: riempie una pagina col maggior numero possibile di sistemi prima di passare a quella successiva. Dunque potrebbe essere preferibile per le partiture con molte pagine, per le quali le altre funzioni di interruzione di pagina potrebbero essere troppo lente o richiedere troppa memoria, o con molto testo. Si abilita con:

```
\paper {
  page-breaking = #ly:minimal-breaking
}
```

Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Interruzione di pagina di una pagina

La funzione `ly:one-page-breaking` è un algoritmo di interruzione di pagina per casi speciali che regola automaticamente l'altezza della pagina per farci entrare la musica, in modo che stia tutto in una sola pagina. La variabile `paper-height` del blocco `paper` viene ignorata, ma le altre impostazioni funzionano normalmente. In particolare, la spaziatura tra l'ultimo sistema (o il testo markup di livello superiore) e il piè di pagina può essere personalizzata grazie a `last-bottom-spacing` nel blocco `paper`. La larghezza della pagina non viene modificata ma può essere impostata con `paper-width` nel blocco `paper`.

Problemi noti e avvertimenti

`ly:one-page-breaking` non è attualmente compatibile con `\bookpart`.

Interruzione di pagina su una linea

La funzione `ly:one-line-breaking` è un algoritmo di interruzione di pagina per uso speciale in quanto pone ogni brano su una sola pagina e una singola linea. Non appaiono né titoli né margini, ma viene mostrato solo il brano musicale.

La larghezza della pagina è regolata in modo che il brano più lungo stia su una sola linea. In particolare, le variabili `paper-width`, `line-width` e `indent` del blocco `\paper` vengono ignorate, sebbene `left-margin` e `right-margin` siano comunque considerate. L'altezza della pagina resta invariata.

Interruzione di pagina su una linea con altezza automatica

La funzione `ly:one-line-auto-height-breaking` funziona proprio come `ly:one-line-breaking` con la differenza che l'altezza della pagina viene modificata automaticamente per adattarsi all'altezza della musica. Più precisamente, la variabile `paper-height` del blocco `\paper` viene impostata in modo che abbracci l'altezza della partitura più alta e i margini superiore e inferiore (`top-margin` e `bottom-margin`).

Fare attenzione al fatto che l'impostazione `top-system-spacing` avrà effetto sulla posizione verticale della musica. Impostarla su `##f` in un blocco `\paper` per posizionare la musica tra i margini superiore e inferiore.

Voltata di pagina ottimale

È spesso necessario trovare una configurazione delle interruzioni di pagina in cui ci sia una pausa al termine di ogni due pagine. In questo modo il musicista può voltare la pagina senza perdere

le note. La funzione `ly:page-turn-breaking` tenta di trovare un'interruzione di pagina che minimizzi densità e allungamento, ma con l'ulteriore restrizione che le voltate di pagina sono permesse solo in punti specifici.

Ci sono due passi da seguire per usare questa funzione. Prima occorre abilitarla nel blocco `\paper`, come è spiegato in Sezione 4.3.2 [Interruzioni di pagina], pagina 570. Poi bisogna indicare alla funzione dove sono permesse le interruzioni di pagina.

Ci sono due modi per fare il secondo passo. Si può specificare manualmente ogni potenziale voltata di pagina, inserendo `\allowPageTurn` nei punti adatti del file di input.

Oppure, se ciò è troppo noioso, si può aggiungere l'incisore `Page_turn_engraver` a un contesto `Staff` o `Voice`. L'incisore `Page_turn_engraver` analizzerà il contesto in cerca di sezioni senza note (non cerca pause, bensì l'assenza di note, in modo che la polifonia su un singolo rigo con pause in una delle parti non confonda `Page_turn_engraver`). Quando trova una sezione senza note abbastanza lunga, `Page_turn_engraver` inserirà il comando `\allowPageTurn` nella stanghetta finale di quella sezione, a meno che non ci sia una stanghetta 'speciale' (come una doppia stanghetta), nel qual caso il comando `\allowPageTurn` sarà inserito nella stanghetta finale "speciale" della sezione.

L'incisore `Page_turn_engraver` legge la proprietà di contesto `pageTurnMinimumRestLength` per determinare quanto deve essere lunga una sezione senza note prima che una voltata di pagina sia considerata. Il valore predefinito di `pageTurnMinimumRestLength` è 1. Per disabilitare le voltate di pagina, impostarla su un valore "molto grande".

```
\new Staff \with { \consists Page_turn_engraver }
{
  a4 b c d |
  R1 | % voltata di pagina permessa qui
  a4 b c d |
  \set Staff.pageTurnMinimumRestLength = #5/2
  R1 | % voltata di pagina non permessa qui
  a4 b r2 |
  R1*2 | % voltata di pagina permessa qui
  a1
}
```

Quando si usano ripetizioni con finali alternativi, l'incisore `Page_turn_engraver` permetterà una voltata di pagina durante la ripetizione soltanto se c'è abbastanza tempo all'inizio e alla fine della ripetizione per voltare indietro la pagina. Se la ripetizione è troppo breve, si può usare `Page_turn_engraver` per *disabilitare* le voltate impostando un valore appropriato per la proprietà di contesto `pageTurnMinimumRepeatLength`. In questo caso `Page_turn_engraver` consentirà le voltate soltanto nelle ripetizioni la cui durata sia maggiore del valore specificato.

I comandi per le voltate di pagina (`\pageTurn`, `\noPageTurn` e `\allowPageTurn`), possono essere usati anche nel livello superiore, nei blocchi markup di livello superiore e tra una partitura e l'altra.

Comandi predefiniti

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Vedi anche

Guida alla notazione: [Variabili di `\paper` per l'interruzione di linea], pagina 560.

Frammenti: Sezione "Spacing" in *Frammenti di codice*.

Problemi noti e avvertimenti

Usare soltanto un incisore `Page_turn_engraver` per partitura. Se ce n'è più d'uno, interferiranno uno con l'altro.

Vedi anche

Guida alla notazione: Sezione 4.4 [Spaziatura verticale], pagina 574.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.4 Spaziatura verticale

La spaziatura verticale è regolata da tre elementi: la quantità di spazio disponibile (ovvero il formato e i margini), la quantità di spazio tra i sistemi e la quantità di spazio tra i righi di un sistema.

4.4.1 Spaziatura verticale flessibile all'interno dei sistemi

Tre meccanismi distinti regolano la spaziatura verticale flessibile all'interno dei sistemi, uno per ognuna delle seguenti categorie:

- *righi non raggruppati*,
- *righi raggruppati* (rigi con un gruppo come `ChoirStaff`, etc.), e
- *linee che non sono rigi* (come `Lyrics`, `ChordNames`, etc.).

L'altezza di ogni sistema è determinata in due fasi. Prima vengono spaziati tutti i rigi in base alla quantità di spazio disponibile. Poi le linee che non sono rigi sono distribuite tra i rigi.

Nota che i meccanismi di spaziatura trattati in questa sezione regolano soltanto la spaziatura verticale dei rigi e delle linee (che non sono rigi) all'interno di singoli sistemi. La spaziatura verticale tra sistemi, partiture, testi e margini separati è regolata dalle variabili `\paper` trattate in Sezione 4.1.4 [Variabili `\paper` della spaziatura verticale flessibile], pagina 555.

Proprietà della spaziatura dentro un sistema

I meccanismi di spaziatura verticale dentro un sistema sono regolati da due gruppi di proprietà dei grob. Il primo gruppo è associato al grob `VerticalAxisGroup`, creato da tutti i rigi e tutte le linee che non sono rigi. Il secondo gruppo è associato al grob `StaffGrouper`, che può essere creato da gruppi di rigi, ma solo se richiamato esplicitamente. Queste proprietà sono descritte una per una alla fine di questa sezione.

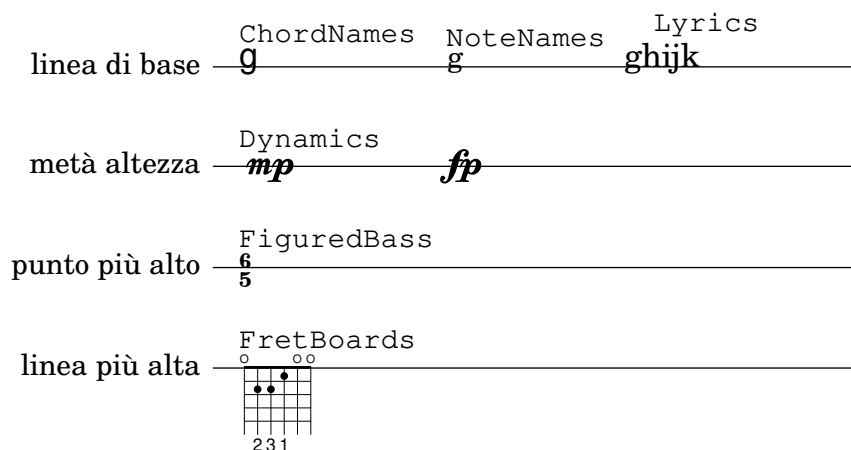
Il nome di queste proprietà (con l'eccezione di `staff-affinity`) hanno il formato `elemento1-elemento2-spacing`, dove `elemento1` e `elemento2` sono gli elementi di cui determinare la distanza. Nota che `elemento2` non è necessariamente sotto `elemento1`; per esempio, `nonstaff-relatedstaff-spacing` prenderà le misure verso l'alto a partire dalla linea che non è un rigo (`nonstaff`) se `staff-affinity` è impostato su `UP`.

Ogni distanza è calcolata tra i *punti di riferimento* dei due elementi. Il punto di riferimento di un rigo è il centro verticale del suo `StaffSymbol` (ovvero la linea centrale se `line-count` (il numero di linee) è dispari; lo spazio centrale se `line-count` è pari). I punti di riferimento per ciascuna linea che non è un rigo sono elencati nella seguente tabella:

Linea non-rigo	Punto di riferimento
<code>ChordNames</code>	linea di base
<code>NoteNames</code>	linea di base
<code>Lyrics</code>	linea di base
<code>Dynamics</code>	metà altezza di 'm'

FiguredBass punto più alto
FretBoards linea più alta

Nell'immagine seguente, le linee orizzontali indicano le posizioni di questi punti di riferimento:



Tutte le proprietà di spaziatura verticale del grob (eccetto staff-affinity) usano la stessa struttura della lista associativa usata dalle variabili di spaziatura di `\paper` trattate in Sezione 4.1.4 [Variabili `\paper` della spaziatura verticale flessibile], pagina 555. Metodi specifici per modificare queste liste sono spiegati in Sezione 5.3.7 [Modifica delle liste associative], pagina 639. Le proprietà dei grob devono essere modificate con un comando `\override` dentro un blocco `\score` o `\layout` e non in un blocco `\paper`.

L'esempio seguente illustra i due modi con cui si possono modificare queste liste associative (alist). La prima dichiarazione trasforma un elemento-valore singolarmente, mentre la seconda ridefinisce completamente la proprietà:

```
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 10)
      (minimum-distance . 9)
      (padding . 1)
      (stretchability . 10))
} { ... }
```

Per cambiare le impostazioni di spaziatura globalmente, inserirle in un blocco `\layout`:

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing.basic-distance = #10
  }
}
```

Le impostazioni predefinite delle proprietà di spaziatura verticale dei grob sono elencate in Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno* e Sezione “StaffGrouper” in *Guida al Funzionamento Interno*. Le modifiche predefinite con `\override` per tipologie specifiche di linee che non sono righe sono elencate nelle descrizioni del relativo contesto in Sezione “Contexts” in *Guida al Funzionamento Interno*.

Proprietà del grob `VerticalAxisGroup`

Le proprietà di `VerticalAxisGroup` sono solitamente modificate con un `\override` nel livello `Staff` (o equivalente).

`staff-staff-spacing`

Usata per determinare la distanza tra il rigo corrente e il rigo inferiore nello stesso sistema, anche se tra i due si trovano una o più linee che non sono righi (come `Lyrics`). Non è applicata all'ultimo rigo di un sistema.

Inizialmente, la proprietà `staff-staff-spacing` di un `VerticalAxisGroup` è una funzione `Scheme` che applica le proprietà di `StaffGrouper` se il rigo fa parte di un gruppo, o la proprietà `default-staff-staff-spacing` del rigo altrimenti. Questo permette ai righi di essere spaziati diversamente quando sono raggruppati. Per ottenere una spaziatura uniforme indipendentemente dal raggruppamento, questa funzione può essere sostituita da un `alist` di spaziatura flessibile, usando la forma di `override` che ridefinisce completamente la variabile, come mostrato prima.

`default-staff-staff-spacing`

Un `alist` di spaziatura flessibile che definisce la proprietà `staff-staff-spacing` usata per i righi isolati, a meno che `staff-staff-spacing` non sia stata impostata esplicitamente con un `\override`.

`staff-affinity`

La direzione del rigo da usare per spaziare la linea che non è un rigo. Le opzioni sono `UP` (su), `DOWN` (giù) e `CENTER` (centro). Se impostata su `CENTER`, la linea fuori dal rigo si troverà in un punto equidistante tra i due righi più vicini su qualunque lato, a meno che delle collisioni o altre costrizioni di spazio non lo impediscano. Linee (che non sono righi) adiacenti dovrebbero avere un valore di `staff-affinity` che non cresce: per esempio, una linea che non è un rigo impostata su `UP` non deve seguire immediatamente una linea impostata su `DOWN`. Linee che non sono righi in cima a un sistema devono usare `DOWN`; quelle in fondo `UP`. Impostando `staff-affinity` per un rigo, questo sarà trattato come una linea che non è un rigo. Impostando `staff-affinity` su `#f`, una linea che non è un rigo sarà trattata come un rigo. Impostando `staff-affinity` su `UP`, `CENTER` o `DOWN`, un rigo verrà spaziato come se fosse una linea che non è un rigo.

`nonstaff-relatedstaff-spacing`

La distanza fra la linea (che non è un rigo) corrente e il rigo più vicino nella direzione di `staff-affinity`, se non ci sono linee che non sono righi tra le due e `staff-affinity` è impostato su `UP` o `DOWN`. Se `staff-affinity` è impostato su `CENTER`, viene usato `nonstaff-relatedstaff-spacing` per i righi più vicini su *entrambi* i lati, anche se appaiono altre linee tra quella corrente e uno qualsiasi dei righi. Ciò significa che il posizionamento di una linea dipende sia dai righi che dalle linee circostanti. Impostando la proprietà `stretchability` di uno di questi tipi di spaziatura su un piccolo valore, quella spaziatura sarà dominante. Impostando `stretchability` su un grande valore, quella spaziatura avrà poco effetto.

`nonstaff-nonstaff-spacing`

La distanza fra la linea (che non è un rigo) corrente e quella successiva nella direzione di `staff-affinity`, se entrambe sono sullo stesso lato del rigo in questione e se `staff-affinity` è impostata su `UP` o `DOWN`.

`nonstaff-unrelatedstaff-spacing`

La distanza fra la linea (che non è un rigo) corrente e il rigo nella direzione opposta rispetto a `staff-affinity`, se non ci sono altre linee tra i due e se `staff-affinity`

è impostato su UP o DOWN. Ciò può servire, per esempio, a imporre un padding minimo tra una linea Lyrics e il rigo al quale non appartiene.

Proprietà del grob **StaffGrouper**

Le proprietà di **StaffGrouper** sono solitamente modificate con un `\override` nel livello **StaffGroup** (o livello equivalente).

staff-staff-spacing

La distanza tra righi consecutivi del gruppo di righi corrente. La proprietà **staff-staff-spacing** del grob **VerticalAxisGroup** di un singolo rigo può essere sovrascritta con varie impostazioni di spaziatura per quel rigo.

staffgroup-staff-spacing

La distanza tra l'ultimo rigo del gruppo di righi corrente e il rigo immediatamente successivo nello stesso sistema, anche se tra i due righi ci sono una o più linee che non sono righi (come Lyrics). Non è applicata al rigo inferiore di un sistema. La proprietà **staff-staff-spacing** del grob **VerticalAxisGroup** di un singolo rigo può essere sovrascritta con varie impostazioni di spaziatura per quel rigo.

Vedi anche

Guida alla notazione: Sezione 4.1.4 [Variabili `\paper` della spaziatura verticale flessibile], pagina 555, Sezione 5.3.7 [Modifica delle liste associative], pagina 639.

File installati: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Guida al funzionamento interno: Sezione “Contexts” in *Guida al Funzionamento Interno*, Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno*, Sezione “StaffGrouper” in *Guida al Funzionamento Interno*.

Spaziatura dei righi non raggruppati

I *righi* (come **Staff**, **DrumStaff**, **TabStaff**, etc.) sono contesti che possono contenere uno o più contesti voce, ma non possono contenere altri righi.

Le seguenti proprietà influenzano la spaziatura di righi *non raggruppati*:

- Proprietà di **VerticalAxisGroup**:
 - **default-staff-staff-spacing**
 - **staff-staff-spacing**

Queste proprietà del grob sono state descritte una a una in precedenza; vedi [Proprietà della spaziatura dentro un sistema], pagina 574.

Altre proprietà entrano in gioco per i righi che sono parte di un gruppo; vedi [Spaziatura dei righi raggruppati], pagina 578.

L'esempio seguente mostra come la proprietà **default-staff-staff-spacing** possa influenzare la spaziatura di righi non raggruppati. Le stesse modifiche applicate a **staff-staff-spacing** avrebbero lo stesso effetto, ma verrebbero applicate anche nel caso in cui i righi siano combinati in uno o più gruppi.

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup.default-staff-staff-spacing =
      #'((basic-distance . 8)
        (minimum-distance . 7)
        (padding . 1))
  }
}
```

```

}

<<
% Questa nota molto bassa ha bisogno di più spazio di quanto 'basic-distance
% possa fornirne, dunque la distanza tra questo rigo e quello successivo
% è determinato da 'padding.
\new Staff { b,2 r | }

% Qui 'basic-distance fornisce abbastanza spazio, e non c'è bisogno
% di comprimere lo spazio (verso 'minimum-distance) per far spazio
% per qualcos'altro sulla pagina, dunque la distanza tra questo
% rigo e quello successivo è determinato da 'basic-distance.
\new Staff { \clef bass g2 r | }

% Impostando 'padding su un valore negativo, è possibile far sì che
% i rigi entrino in collisione. Il più basso valore accettabile per
% 'basic-distance è 0.
\new Staff \with {
  \override VerticalAxisGroup.default-staff-staff-spacing =
    #'((basic-distance . 3.5)
      (padding . -10))
} { \clef bass g2 r | }
\new Staff { \clef bass g2 r | }
>>

```



Vedi anche

File installati: scm/define-grobs.scm.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno*.

Spaziatura dei rigi raggruppati

Nelle partiture orchestrali e in alte grosse partiture, di norma i rigi vengono raggruppati. Lo spazio tra i gruppi è più ampio dello spazio tra i rigi dello stesso gruppo.

I *gruppi di rigi* (come `StaffGroup`, `ChoirStaff`, etc.) sono contesti che possono contenere uno o più rigi simultaneamente.

Le seguenti proprietà influenzano la spaziatura dei rigi nei gruppi:

- Proprietà di `VerticalAxisGroup`:
 - `staff-staff-spacing`
- Proprietà di `StaffGroup`:
 - `staff-staff-spacing`

- `staffgroup-staff-spacing`

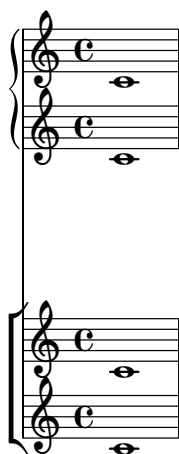
Queste proprietà dei grob sono descritte una a una in una sezione precedente; vedi [Proprietà della spaziatura dentro un sistema], pagina 574.

L'esempio seguente mostra come le proprietà del grob `StaffGrouper` possano influenzare la spaziatura dei righi raggruppati:

```
\layout {
  \context {
    \Score
    \override StaffGrouper.staff-staff-spacing.padding = #0
    \override StaffGrouper.staff-staff-spacing.basic-distance = #1
  }
}

<<
\new PianoStaff \with {
  \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #20
} <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>

\new StaffGroup <<
  \new Staff { c'1 }
  \new Staff { c'1 }
>>
>>
```



Vedi anche

File installati: `scm/define-grobs.scm`.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno*, Sezione “StaffGrouper” in *Guida al Funzionamento Interno*.

Spaziatura delle linee che non sono righi

Le *linee che non sono righi* (come `Lyrics`, `ChordNames`, etc.) sono contesti i cui oggetti della formattazione sono disposti come se fossero su dei righi (ovvero su linee orizzontali all’interno

dei sistemi). Precisamente, le linee che non sono righe sono contesti non-rigo che contengono l'incisore Sezione "Axis_group_engraver" in *Guida al Funzionamento Interno*.

Le seguenti proprietà influenzano la spaziatura delle linee che non sono righe:

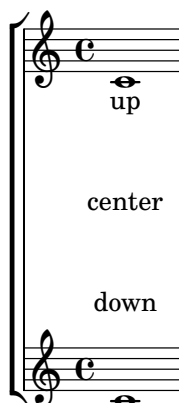
- Proprietà di VerticalAxisGroup:
 - staff-affinity
 - nonstaff-relatedstaff-spacing
 - nonstaff-nonstaff-spacing
 - nonstaff-unrelatedstaff-spacing

Queste proprietà del grob sono descritte una a una in una sezione precedente, vedi [Proprietà della spaziatura dentro un sistema], pagina 574.

L'esempio seguente mostra come la proprietà nonstaff-nonstaff-spacing influenzi la spaziatura di linee che non sono un rigo consecutive. Impostando l'elemento stretchability su un valore molto alto, il testo vocale riesce a allungarsi molto più del solito:

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.nonstaff-nonstaff-spacing.stretchability = #1000
  }
}

\new StaffGroup
<<
  \new Staff \with {
    \override VerticalAxisGroup.staff-staff-spacing = #'((basic-distance . 30))
  } { c'1 }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #UP
  } \lyricmode { up }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #CENTER
  } \lyricmode { center }
  \new Lyrics \with {
    \override VerticalAxisGroup.staff-affinity = #DOWN
  } \lyricmode { down }
  \new Staff { c'1 }
>>
```



Vedi anche

File installati: `ly/engraver-init.ly`, `scm/define-grobs.scm`.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Contexts” in *Guida al Funzionamento Interno*, Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno*.

4.4.2 Posizionamento esplicito di righe e sistemi

Un modo per comprendere i meccanismi di spaziatura verticale appena spiegati è di considerarli come un insieme di impostazioni che regolano la quantità di *padding* verticale tra righe e tra sistemi.

È possibile gestire la spaziatura verticale in un modo diverso usando `NonMusicalPaperColumn.line-break-system-details`. Mentre i meccanismi di spaziatura verticale flessibile specificano il padding verticale, `NonMusicalPaperColumn.line-break-system-details` indica precisamente le posizioni verticali esatte sulla pagina.

`NonMusicalPaperColumn.line-break-system-details` accetta una lista associativa di quattro diverse impostazioni:

- `X-offset`
- `Y-offset`
- `extra-offset`
- `alignment-distances`

Le modifiche del grob con `\override`, incluse quelle per `NonMusicalPaperColumn` come nell'esempio successivo, possono trovarsi in uno di questi tre diversi punti del file di input:

- direttamente in mezzo alle note
- in un blocco `\context`
- nel blocco `\with`

Quando si modifica `NonMusicalPaperColumn`, si usa il solito comando `\override` nei blocchi `\context` e nel blocco `\with`. Invece quando si modifica `NonMusicalPaperColumn` in mezzo alle note, si usa il comando speciale `\overrideProperty`. Ecco alcuni esempi di modifiche di `NonMusicalPaperColumn` col comando speciale `\overrideProperty`:

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((Y-offset . 40))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
      (Y-offset . 40))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((alignment-distances . (15)))
```

```
\overrideProperty NonMusicalPaperColumn.line-break-system-details
  #'((X-offset . 20)
      (Y-offset . 40)
      (alignment-distances . (15))))
```

Per comprendere come funziona ognuna di queste impostazioni, iniziamo vedendo un esempio che non contiene alcuna modifica.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
    \new Staff {
      \repeat unfold 15 { d'4 d' d' d' }
    }
  }
  >>
}
```



Questa partitura isola l'informazione sulle interruzioni di linea e di pagina in una voce apposita. Questa tecnica di creare una voce per le interruzioni permette di tenere la formattazione separata dalla musica via via che il nostro esempio diventa più complicato. Vedi anche Sezione 4.3 [Interruzioni], pagina 567.

Usando comandi `\break` espliciti, la musica viene divisa proporzionalmente in cinque misure per linea. La spaziatura verticale è quella predefinita di LilyPond ma il punto di inizio verticale di ogni sistema è impostato esplicitamente con la coppia `Y-offset` dell'attributo `line-break-system-details` del grob `NonMusicalPaperColumn`:

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
```

```

\score {
  <<
    \new Staff <<
      \new Voice {
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 0))
        s1*5 \break
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 40))
        s1*5 \break
        \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
          #'((Y-offset . 60))
        s1*5 \break
      }
      \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
    >>
    \new Staff {
      \repeat unfold 15 { d'4 d' d' d' }
    }
  >>
}

```



Nota che `line-break-system-details` accetta una lista associativa di molti valori, ma ne abbiamo impostato solo uno in questo esempio. Nota anche che la proprietà `Y-offset` qui determina la posizione verticale esatta sulla pagina in cui ogni nuovo sistema verrà visualizzato.

Oltre al posizionamento assoluto che si ottiene con `Y-offset` e `X-offset`, è possibile anche il posizionamento relativo tramite la proprietà `extra-offset` di `line-break-system-details`. Il

posizionamento è relativo alla formattazione predefinita o al posizionamento assoluto determinato dall'uso di X-offset e Y-offset. La proprietà `extra-offset` accetta una coppia di numeri che determinano lo spostamento lungo gli assi X e Y.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((extra-offset . (0 . 10)))
          s1*5 \break
          \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
            #'((extra-offset . (0 . 10)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Ora che abbiamo impostato esplicitamente il punto di inizio verticale di ogni sistema, possiamo impostare manualmente anche le distanze verticali tra i righi. Per farlo usiamo la sottoproprietà `alignment-distances` di `line-break-system-details`.

```

\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 20)
          (alignment-distances . (10)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 60)
          (alignment-distances . (15)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 85)
          (alignment-distances . (20)))
      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new Staff {
    \repeat unfold 15 { d'4 d' d' d' }
  }
  >>
}
}

```



Nota che qui assegnamo due valori diversi all'attributo `line-break-system-details` del grob `NonMusicalPaperColumn`. Sebbene l'attributo `alist line-break-system-details` accetti molti altri parametri di spaziatura (inclusa, per esempio, una coppia corrispondente di `X-offset`), è sufficiente impostare soltanto le coppie `Y-offset` e `alignment-distances` per regolare il punto di inizio verticale di ogni sistema e ogni rigo. Infine nota che `alignment-distances` specifica il posizionamento verticale dei rigi ma non dei gruppi di rigi.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
    \new Staff <<
    \new Voice {
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 0)
           (alignment-distances . (30 10)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 60)
           (alignment-distances . (10 10)))
      s1*5 \break
      \overrideProperty Score.NonMusicalPaperColumn.line-break-system-details
        #'((Y-offset . 100)
```

```

        (alignment-distances . (10 30)))
      s1*5 \break
    }
    \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
  >>
  \new StaffGroup <<
    \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
    \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
  >>
  >>
}
}

```


The image displays three systems of musical notation, each consisting of three staves. The first system shows a single melodic line on the top staff, with the middle and bottom staves empty. The second system shows a piano accompaniment with a treble and bass staff, and a single melodic line on the top staff. The third system shows a piano accompaniment with a treble and bass staff, and a single melodic line on the top staff. The notation is in common time (C) and features a sequence of eighth notes.

Alcuni punti da considerare:

- Quando si usa `alignment-distances`, il testo vocale e altre linee che non sono righe non contano come rigo.
- Le unità dei numeri assegnati a `X-offset`, `Y-offset`, `extra-offset` e `alignment-distances` sono interpretati come multipli della distanza tra linee

del rigo adiacenti. Valori positivi spostano in su righe e testo, valori negativi li spostano in giù.

- Dato che le impostazioni di `NonMusicalPaperColumn.line-break-system-details` illustrate qui permettono il posizionamento di righe e sistemi ovunque sulla pagina, è possibile violare i confini del foglio o dei margini o perfino sovrapporre righe e sistemi uno sopra l'altro. Ciò può essere evitato assegnando valori ragionevoli a queste diverse impostazioni.

Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.4.3 Elusione delle collisioni verticali

Intuitivamente, ci sono alcuni oggetti della notazione musicale che appartengono al rigo e altri che devono essere disposti fuori dal rigo. Gli oggetti esterni al rigo comprendono i numeri di chiamata, il testo e le dinamiche (d'ora in avanti tutti questi elementi saranno chiamati oggetti esterni al rigo). La regola di LilyPond per il posizionamento verticale degli oggetti esterni al rigo è di disporli il più vicino possibile al rigo ma non così vicino da farli collidere con un altro oggetto.

LilyPond usa la proprietà `outside-staff-priority` per determinare se un grob è un oggetto fuori dal rigo: se `outside-staff-priority` è un numero, il grob è un oggetto esterno al rigo. `outside-staff-priority` indica a LilyPond anche in quale ordine disporre gli oggetti.

LilyPond posiziona prima tutti gli oggetti che non sono esterni al rigo. Quindi ordina gli oggetti esterni al rigo in base al loro valore di `outside-staff-priority` (in ordine crescente). Uno per volta, LilyPond prende gli oggetti esterni al rigo e li dispone in modo che non entrino in collisione con alcun oggetto che sia già stato disposto. Ovvero, se due grob esterni al rigo si contendono lo stesso spazio, quello col valore di `outside-staff-priority` più basso sarà posto più vicino al rigo.

Un elenco delle proprietà esterne al rigo si trova in Sezione “La proprietà `outside-staff-priority`” in *Manuale di Apprendimento*.

```
\relative c' ' {
  c4_"Testo"\pp
  r2.
  \once \override TextScript.outside-staff-priority = #1
  c4_"Testo"\pp % stavolta il testo sarà più vicino al rigo
  r2.
  % impostando outside-staff-priority su un non-numero,
  % disabilitiamo l'elusione automatica delle collisioni
  \once \override TextScript.outside-staff-priority = ##f
  \once \override DynamicLineSpanner.outside-staff-priority = ##f
  c4_"Testo"\pp % qui entrano in collisione
}
```



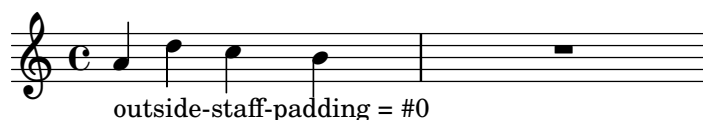
Il padding verticale intorno agli oggetti esterni al rigo può essere regolato con `outside-staff-padding`.

```
\relative {
  \once \override TextScript.outside-staff-padding = #0
```

```

a'4-"outside-staff-padding = #0"
\once \override TextScript.outside-staff-padding = #3
d-"outside-staff-padding = #3"
c-"outside-staff-padding predefinito"
b-"outside-staff-padding predefinito"
R1
}

```



outside-staff-padding = #0

outside-staff-padding = #3

outside-staff-padding predefinito

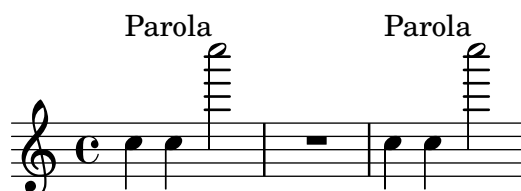
outside-staff-padding predefinito

Per impostazione predefinita, gli oggetti esterni al rigo sono disposti in modo da evitare la collisione orizzontale con grob posizionati precedentemente. Ciò può portare a situazioni in cui gli oggetti sono posizionati uno vicino all'altro orizzontalmente. Come è dimostrato nell'esempio successivo, impostando `outside-staff-horizontal-padding` si aumenta la spaziatura orizzontale richiesta e in questo caso si sposta in su il testo per impedire che si avvicini troppo ai tagli aggiuntivi.

```

\relative {
  c'4^"Parola" c'2
  R1
  \once \override TextScript.outside-staff-horizontal-padding = #1
  c,,4^"Parola" c'2
}

```



Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.5 Spaziatura orizzontale

4.5.1 Panoramica sulla spaziatura orizzontale

Il motore della spaziatura traduce le differenze delle durate delle note in distanze allungabili ('springs') di diversa lunghezza. Durate più lunghe occupano più spazio, quelle più brevi ne occupano meno. Le durate più brevi occupano una quantità fissa di spazio (regolata da `shortest-duration-space` nell'oggetto Sezione “SpacingSpanner” in *Guida al Funzionamento Interno*). Più lunga è la durata, più spazio occupa: raddoppiando una durata si aggiunge spazio alla nota di una quantità pari al valore di `spacing-increment`.

Per esempio, il brano seguente contiene molte minime, semiminime e crome; la croma (1/8) è seguita da 1 Larghezza della Testa di Nota (LTN). La semiminima (1/4) è seguita da 2 LTN, la minima (1/2) da 3 LTN, etc.

```

\relative c' {

```

```

c2 c4. c8
c4. c8 c4. c8
c8 c c4 c c
}

```



Solitamente, `spacing-increment` è impostato su 1.2 di spazio rigo, che equivale all'incirca alla larghezza della testa di nota, e `shortest-duration-space` è impostato su 2.0, che significa che la nota più breve occupa 2.4 di spazio rigo (2 volte `spacing-increment`) di spazio orizzontale. Questo spazio è calcolato dal margine sinistro del simbolo, dunque le note più brevi sono generalmente seguite da un LTN di spazio.

Se si seguisse esattamente la procedura descritta, aggiungendo una sola biscroma (1/32) a un brano che usa solo crome e semicrome, la spaziatura orizzontale dell'intero brano sarebbe troppo larga. Infatti la nota più breve non è più una semicroma ma una biscroma, aggiungendo quindi 1 LTN a ogni nota. Per impedire ciò, la durata più breve per la spaziatura non deriva dalla nota più breve, bensì dalla nota che ricorre più frequentemente nel brano.

La durata più breve più comune viene individuata nel modo seguente. In ogni misura viene determinata la durata più breve e quella più frequente viene scelta come base per la spaziatura, con la condizione che tale durata debba essere sempre uguale o inferiore a una nota di un ottavo.

Tali durate possono anche essere personalizzate. Impostando `common-shortest-duration` in Sezione “SpacingSpanner” in *Guida al Funzionamento Interno*, si imposta la durata di base per la spaziatura. La durata massima per essa (solitamente un ottavo), si imposta con `base-shortest-duration`.

Note ancora più brevi della nota più breve più comune sono seguite da uno spazio proporzionale alla loro durata rispetto a essa. Dunque se aggiungessimo solo alcuni sedicesimi all'esempio precedente, sarebbero seguiti dalla metà di LTN:

```

\relative { c''2 c4. c8 | c4. c16[ c] c4. c8 | c8 c c4 c c }

```



Come è spiegato nel saggio *Essay on automated music engraving*, le direzioni del gambo influenzano la spaziatura (vedi Sezione “Spaziatura ottica” in *Saggio*) e possono essere aggiustate usando la proprietà `stem-spacing-correction` dell'oggetto Sezione “NoteSpacing” in *Guida al Funzionamento Interno* (che sono generati per ogni contesto Sezione “Voice” in *Guida al Funzionamento Interno*).

L'oggetto `StaffSpacing` (generato nel contesto Sezione “Staff” in *Guida al Funzionamento Interno*) contiene la stessa proprietà per regolare la spaziatura tra gambo e stanghetta.

L'esempio seguente mostra tutto questo: una volta con le impostazioni predefinite e una volta con un aggiustamento esagerato:



LilyPond supporta la notazione proporzionale, vedi Sezione 4.5.5 [Notazione proporzionale], pagina 595.

Vedi anche

Essay on automated music engraving: Sezione “Spaziatura ottica” in *Saggio*.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “SpacingSpanner” in *Guida al Funzionamento Interno*, Sezione “NoteSpacing” in *Guida al Funzionamento Interno*, Sezione “StaffSpacing” in *Guida al Funzionamento Interno*, Sezione “NonMusicalPaperColumn” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Non esiste un modo semplice per modificare manualmente la spaziatura. Per aggirare il problema si può inserire dell’ulteriore spazio in una partitura, regolando il valore di padding di quanto è necessario:

```
\override Score.NonMusicalPaperColumn.padding = #10
```

Non esiste alcun trucco per diminuire la quantità di spazio.

4.5.2 Nuova spaziatura nel corso di un brano

Nuove sezioni con parametri di spaziatura diversi possono essere iniziati col comando `\newSpacingSection`. Ciò può essere utile per sezioni con nozioni diverse di note ‘lunghe’ e note ‘brevi’. Il comando `\newSpacingSection` crea un nuovo oggetto `SpacingSpanner` in quel momento musicale.

Nell’esempio seguente, il cambio di indicazione di tempo introduce una nuova sezione, e i sedicesimi hanno automaticamente una spaziatura un po’ più larga.

```
\relative c' {
  \time 2/4
  c4 c8 c
  c8 c c4 c16[ c c8] c4
  \newSpacingSection
  \time 4/16
  c16[ c c8]
}
```



Se gli aggiustamenti automatici della spaziatura non producono la spaziatura richiesta, si possono applicare degli `\override` manuali alle sue proprietà. Questi devono essere applicati nello stesso momento musicale del comando `\newSpacingSection` stesso e avranno effetto sulla spaziatura di tutta la musica seguente finché le proprietà non vengono cambiate in una nuova sezione. Per esempio:

```
\relative c' {
  \time 4/16
  c16[ c c8]
  \newSpacingSection
  \override Score.SpacingSpanner.spacing-increment = #2
  c16[ c c8]
  \newSpacingSection
  \revert Score.SpacingSpanner.spacing-increment
  c16[ c c8]
}
```



Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “SpacingSpanner” in *Guida al Funzionamento Interno*.

4.5.3 Modifica della spaziatura orizzontale

La spaziatura orizzontale può essere modificata tramite la proprietà `base-shortest-duration`. Nel prossimo esempio confrontiamo la stessa musica, prima senza cambiare la proprietà e poi cambiandola. Valori più grandi di `ly:make-moment` produrranno musica più densa. Nota che `ly:make-moment` costituisce una durata, dunque 1 4 è una durata più lunga di 1 16.

```
\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```



```
\score {
  \relative {
    g'4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration = \musicLength 16
    }
  }
}
```

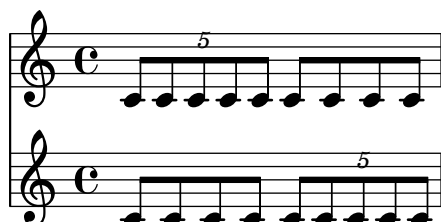




Allungamento uniforme dei gruppi irregolari

L'impostazione predefinita prevede che la spaziatura nei gruppi irregolari dipenda da vari fattori diversi dalla durata (come alterazioni, cambi di chiave, etc). Per ignorare tali simboli e forzare la spaziatura perché sia uniforme, usare `Score.SpacingSpanner.uniform-stretching`. Questa proprietà può essere modificata soltanto all'inizio di una partitura:

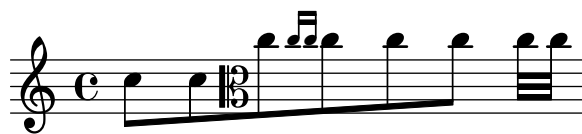
```
\score {
  <<
    \new Staff \relative c' {
      \tuplet 5/4 { c8 c c c c } c8 c c c
    }
    \new Staff \relative c' {
      c8 c c c \tuplet 5/4 { c8 c c c c }
    }
  >>
  \layout {
    \context {
      \Score
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```



Spaziatura rigorosa della nota

Se si imposta `strict-note-spacing`, la spaziatura tra le note non tiene conto di chiavi, stanghette e abbellimenti:

```
\override Score.SpacingSpanner.strict-note-spacing = ##t
\new Staff \relative {
  c''8[ c \clef alto c \grace { c16 c } c8 c c] c32[ c] }
```



Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.5.4 Larghezza della linea

Le impostazioni fondamentali che influenzano la spaziatura sono `indent` e `line-width`, impostate nel blocco `\layout`. Regolano l’indentazione della prima linea musicale e la lunghezza delle linee.

Se `ragged-right` è impostato su `vero` nel blocco `\layout`, i sistemi terminano alla loro naturale lunghezza orizzontale, invece di essere espansi orizzontalmente per riempire tutta la linea. Ciò è utile in caso di brevi frammenti e per verificare quanto è compatta la spaziatura naturale. L’impostazione predefinita è solitamente `falso`, ma se la partitura ha un solo sistema il valore predefinito è `vero`.

L’opzione `ragged-last` è simile a `ragged-right`, ma ha effetto soltanto sull’ultima linea del brano. Nessune restrizioni vengono poste su quella linea. Il risultato è simile alla formattazione dei paragrafi di testo. In un paragrafo l’ultima linea occupa la sua naturale lunghezza orizzontale.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

Vedi anche

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.5.5 Notazione proporzionale

LilyPond supporta la notazione proporzionale, un tipo di spaziatura orizzontale in cui ogni nota occupa una quantità di spazio orizzontale esattamente equivalente alla sua durata musicale. Questo tipo di spaziatura proporzionale può essere paragonata alla spaziatura orizzontale su carta quadrettata. Alcune partiture della fine del ventesimo secolo e dell’inizio del ventunesimo usano la notazione proporzionale per chiarire relazioni ritmiche complesse o per agevolare il posizionamento della linea del tempo o di altri elementi grafici direttamente nella partitura.

LilyPond supporta cinque diverse impostazioni per la notazione proporzionale, che possono essere usate insieme o da sole:

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove Separating_line_group_engraver`
- `\override PaperColumn.used = ##t`

Nell’esempio seguente analizziamo queste cinque diverse impostazioni di notazione proporzionale e valutiamo come esse interagiscono tra loro.

Iniziamo con l’esempio seguente di una misura, che usa la spaziatura classica con la giustificazione del rigo disattivata.

```
\score {
  <<
```



```

\new RhythmicStaff {
  c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
}
>>
}

```



La minima all'inizio della misura occupa uno spazio molto inferiore alla metà dello spazio orizzontale della misura. Ugualmente, i sedicesimi e le quintine di sedicesimi alla fine della battuta insieme occupano molto più spazio della metà dello spazio orizzontale della misura.

Nell'incisione tipografica classica, questa spaziatura è solitamente proprio ciò che si desidera, perché è possibile prendere in prestito dello spazio orizzontale dalla minima e economizzare lo spazio orizzontale complessivo della misura.

D'altra parte, se vogliamo inserire una linea del tempo con tacche o altri elementi grafici sopra o sotto la partitura, abbiamo bisogno della notazione proporzionale. Per attivarla si usa l'impostazione `proportionalNotationDuration`.

```

\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #1/20
    }
  }
}

```



La minima all'inizio della misura e le note più veloci nella seconda metà della misura ora occupano la stessa quantità di spazio orizzontale. Potremmo inserire una linea del tempo con tacche o un'immagine grafica sopra o sotto questo esempio.

L'impostazione `proportionalNotationDuration` è un'impostazione di contesto che si trova in `Score`. Ricordiamo che le impostazioni di contesto possono apparire in tre luoghi del file di input: in un blocco `\with`, in un blocco `\context` o direttamente in mezzo alle note preceduta dal comando `\set`. Come per tutte le impostazioni di contesto, l'utente può scegliere in quale di questi tre luoghi impostare `proportionalNotationDuration`.

L'impostazione `proportionalNotationDuration` prende un solo argomento, che è la durata di riferimento in base alla quale tutta la musica verrà spaziata. La funzione `Scheme` di LilyPond `make-moment` prende due argomenti: un numeratore e un denominatore che insieme esprimono una qualche frazione di una nota intera. La funzione di chiamata `(ly:make-moment 1/20)` produce quindi una durata di riferimento di un ventesimo di nota. Sono ammessi anche valori come `(ly:make-moment 1/16)`, `(ly:make-moment 1/8)` e `(ly:make-moment 3/97)`.

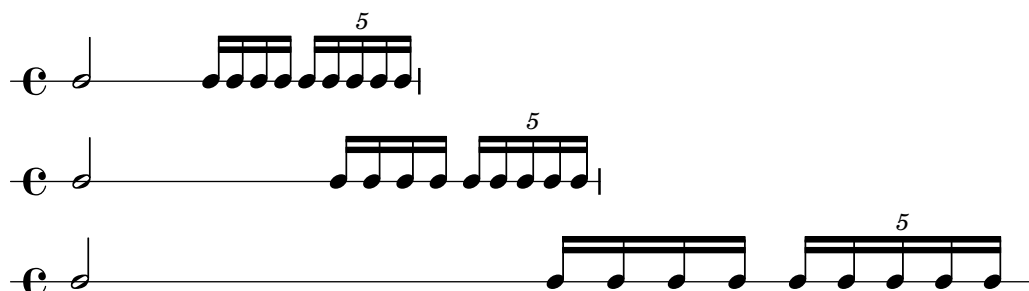
Come scegliamo la durata di riferimento corretta da passare a `proportionalNotationDuration`? Solitamente attraverso un processo di prova e errore, iniziando con una durata vicina alla più veloce (o piccola) durata del brano. Durate di

riferimento più piccole determinano una spaziatura della musica più larga; quelle più grandi causano una spaziatura più stretta.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #1/8
    }
  }
}
```

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #1/16
    }
  }
}
```

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #1/32
    }
  }
}
```

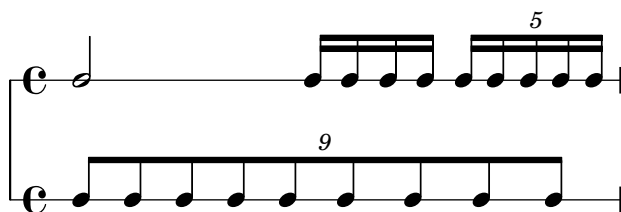


Nota che una durata di riferimento troppo grande – come la nota di un ottavo, sopra – determina una spaziatura della musica troppo stretta e può causare collisioni tra le teste di nota. Fare attenzione anche al fatto che la notazione proporzionale in generale occupa più spazio orizzontale della spaziatura classica. Insomma, la spaziatura proporzionale fornisce chiarezza ritmica al costo dello spazio orizzontale.

Ora vediamo come spaziare in modo ottimale i gruppi irregolari sovrapposti.

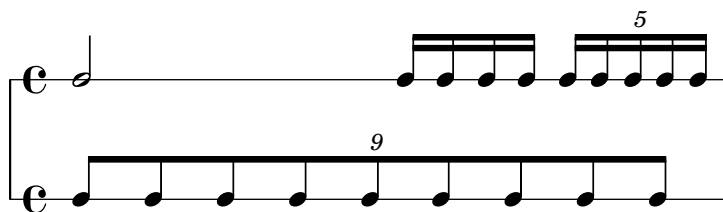
Iniziamo esaminando cosa succede al nostro esempio di partenza, con la spaziatura classica, quando aggiungiamo un secondo rigo con un diverso tipo di gruppo irregolare.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
}
```



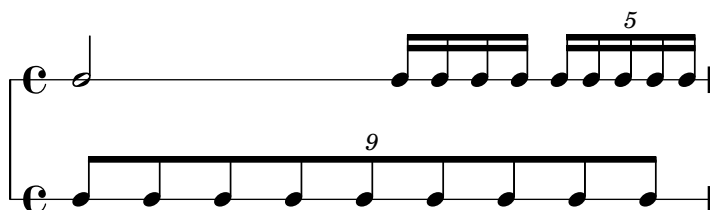
La spaziatura è pessima perché le note del rigo inferiore spaziate a distanze uguali non si allungano in modo uniforme. Le incisioni classiche contengono pochissime terzine complesse e quindi le regole di incisione classica possono generare questo tipo di risultato. Impostando `proportionalNotationDuration` ciò viene corretto.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #1/20
    }
  }
}
```



Ma se osserviamo con attenzione possiamo vedere che le note della seconda metà della nonina hanno una spaziatura leggermente più larga delle note della prima parte della nonina. Per assicurare un allungamento uniforme, attiviamo `uniform-stretching`, una proprietà di `SpacingSpanner`.

```
\score {
  <<
    \new RhythmicStaff {
      c2 16 16 16 16 \tuplet 5/4 { 16 16 16 16 16 }
    }
    \new RhythmicStaff {
      \tuplet 9/8 { c8 8 8 8 8 8 8 8 8 }
    }
  >>
  \layout {
    \context {
      \Score
      \proportionalNotationDuration = #1/20
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```



Il nostro esempio di due righe ora ha una spaziatura esatta, le relazioni ritmiche sono visivamente chiare e possiamo includere una linea del tempo con tacche o altro elemento grafico, se lo vogliamo.

Nota che la notazione proporzionale di LilyPond si aspetta che tutte le partiture proporzionali impostino l'attributo `'uniform-stretching` di `SpacingSpanner` su `##t`. Impostare `proportionalNotationDuration` senza impostare anche l'attributo `'uniform-stretching` di `SpacingSpanner` su `##t` farà sì, per esempio, che le pause invisibili occupino una quantità scorretta di spazio orizzontale.

`SpacingSpanner` è un grob astratto che si trova nel contesto `Score`. Come per le impostazioni di `proportionalNotationDuration`, le modifiche `\override` a `SpacingSpanner` si possono trovare in uno dei tre diversi punti del file di input – nel blocco `\with` del brano, nel blocco `\context` del brano o direttamente in mezzo alle note.

Per impostazione predefinita, esiste un solo `SpacingSpanner` per `Score`. Ciò significa che `uniform-stretching` è attivato o disattivato per l'intera partitura. Possiamo tuttavia modificare tale comportamento e attivare diverse funzionalità di spaziatura in punti diversi del brano. Per farlo si usa il comando `\newSpacingSection`. Maggiori informazioni in Sezione 4.5.2 [Nuova spaziatura nel corso di un brano], pagina 592.

Ora esaminiamo gli effetti dell'incisore `Separating_line_group_engraver` e vediamo perché le partiture proporzionali solitamente tolgano questo incisore. L'esempio seguente mostra che c'è un piccolo spazio "introduttivo" proprio prima della prima nota di ogni sistema.

```
\paper {
  indent = #0
}
```

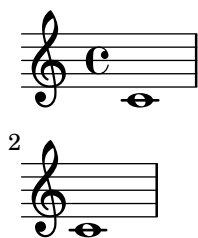
```
\new Staff {
  c'1
  \break
  c'1
}
```



L'ampiezza di questo spazio introduttivo è la stessa sia dopo un'indicazione di tempo che dopo un'armatura di chiave o una chiave. È l'incisore `Separating_line_group_engraver` a causare questo spazio. Togliendolo lo spazio diventa zero.

```
\paper {
  indent = #0
}

\new Staff \with {
  \remove Separating_line_group_engraver
} {
  c'1
  \break
  c'1
}
```



Nella notazione proporzionale gli elementi non musicali come le indicazioni di tempo, le armature di chiave, le chiavi e le alterazioni sono problematiche. Nessuna di queste infatti ha una durata ritmica, ma tutte occupano spazio orizzontale. Questi problemi sono affrontati diversamente dalle varie partiture proporzionali.

È possibile evitare i problemi di spaziatura dovuti alle armature di chiave semplicemente omettendole. Questa è un'opzione valida dato che la maggior parte delle partiture proporzionali sono di musica contemporanea. Lo stesso potrebbe valere per le indicazioni di tempo, specialmente per quelle partiture che includono una linea del tempo o altri elementi grafici. Ma queste partiture sono un'eccezione e la maggior parte delle partiture proporzionali hanno almeno qualche indicazione di tempo. Le chiavi e le alterazioni sono ancora più fondamentali.

Dunque quali strategie adottare per spaziare elementi non musicali nel contesto di musica proporzionale? Una valida opzione è la proprietà `strict-note-spacing` di `SpacingSpanner`. Confrontiamo i seguenti due righe:

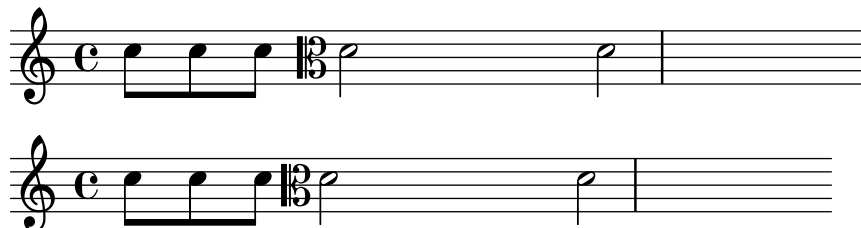
```
\new Staff {
  \set Score.proportionalNotationDuration = #1/16
```

```

c''8 8 8 \clef alto d'2 2
}

\new Staff {
  \set Score.proportionalNotationDuration = #1/16
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  c''8 8 8 \clef alto d'2 2
}

```



Entrambi sono proporzionali, ma la spaziatura del primo è troppo larga a causa del cambio di chiave. La spaziatura del secondo resta invece stretta, perché è attivato `strict-note-spacing`. Attivando `strict-note-spacing`, la larghezza di indicazioni di tempo, armature di chiave, chiavi e alterazioni non ha alcun ruolo nell'algoritmo di spaziatura.

Oltre alle impostazioni che abbiamo visto, ce ne sono altre che appaiono frequentemente nelle partiture proporzionali, tra cui:

- `\override SpacingSpanner.strict-grace-spacing = ##t`
- `\set tupletFullLength = ##t`
- `\override Beam.breakable = ##t`
- `\override Glissando.breakable = ##t`
- `\override TextSpanner.breakable = ##t`
- `\remove Forbid_line_break_engraver` nel contesto `Voice`

Queste impostazioni spaziano in modo conciso gli abbellimenti, estendono le parentesi dei gruppi irregolari per contrassegnare i punti di inizio e di fine del ritmo, e permettono agli elementi che si estendono orizzontalmente di andare oltre i sistemi e le pagine. Consultare le sezioni del manuale per queste impostazioni.

Vedi anche

Guida alla notazione: Sezione 4.5.2 [Nuova spaziatura nel corso di un brano], pagina 592.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.6 Riduzione del numero di pagine di una partitura

Talvolta può capitare di avere uno o due rigli su una seconda (o terza, o quarta. . .) pagina. Ciò è fastidioso, specialmente se c'è molto spazio nelle pagine precedenti.

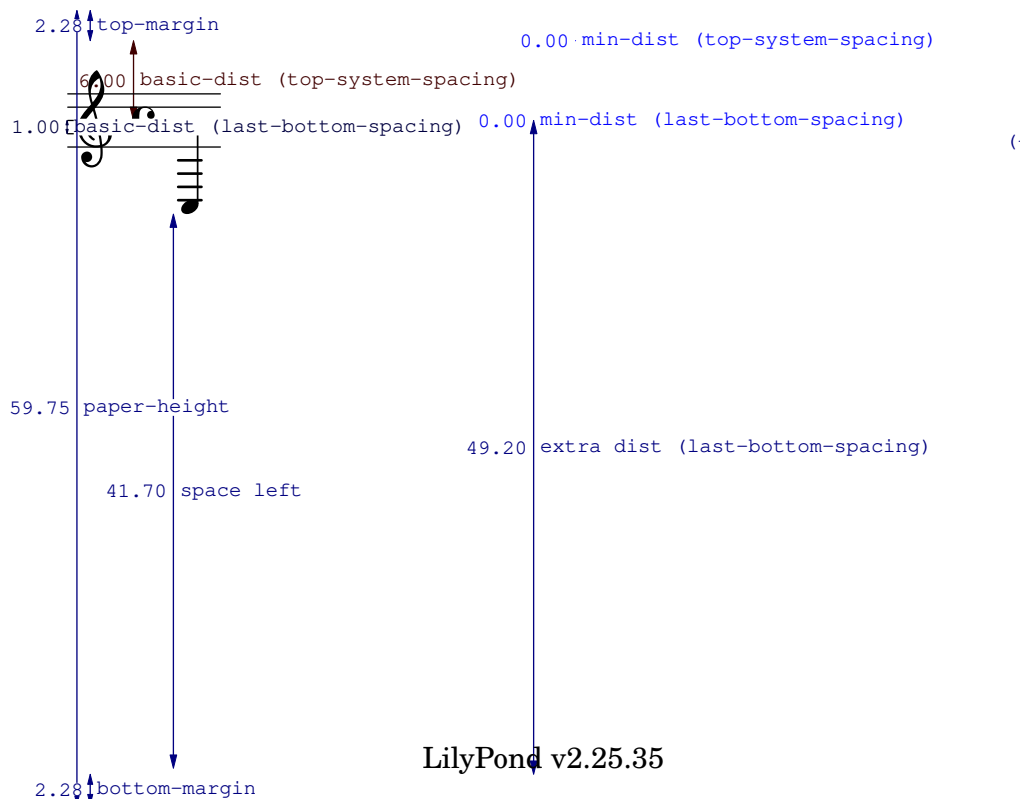
Quando si studiano i problemi di formattazione, uno strumento irrinunciabile è `annotate-spacing`, un comando che mostra i valori delle diverse variabili di spaziatura. Maggiori dettagli nella prossima sezione, Sezione 4.6.1 [Visualizzare la spaziatura], pagina 601.

4.6.1 Visualizzare la spaziatura

Per visualizzare graficamente le dimensioni delle variabili della formattazione verticale che possono essere modificate per formattare la pagina, impostare `annotate-spacing` nel blocco `\paper`:

```
\book {
```

```
\score { { c4 } }
\paper { annotate-spacing = ##t }
}
```



Tutte le dimensioni della formattazione sono visualizzate in spazi rigo, indipendentemente dalle unità di misura specificate nei blocchi `\paper` o `\layout`. Nell'esempio qui sopra, l'altezza del foglio (`paper-height`) ha un valore di 59.75 spazi rigo (`staff-space`), e la dimensione del rigo (`staff-size`) è pari a 20 punti (il valore predefinito). Nota che:

$$\begin{aligned}
 1 \text{ punto} &= (25.4/72.27) \text{ mm} \\
 1 \text{ staff-space} &= (\text{staff-size})/4 \text{ pts} \\
 &= (\text{staff-size})/4 \cdot (25.4/72.27) \text{ mm}
 \end{aligned}$$

In questo caso, uno `staff-space` è uguale all'incirca a 1.757mm. Dunque i 59.75 `staff-space` di `paper-height` corrispondono a 105 millimetri, pari all'altezza del formato a6 in orientamento orizzontale. Le coppie (a,b) sono intervalli, dove a è l'estremo inferiore e b l'estremo superiore dell'intervallo.

Vedi anche

Guida alla notazione: Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

4.6.2 Modificare la spaziatura

L'output di `annotate-spacing` svela le dimensioni verticali molto dettagliatamente. Maggiori informazioni su come modificare i margini e altre variabili di formattazione si trovano in Sezione 4.1 [Formattazione della pagina], pagina 552.

Oltre ai margini, ci sono altre opzioni utili per salvare spazio:

- Forzare i sistemi perché si avvicinino il più possibile (per far entrare più sistemi possibile in una pagina) mentre sono spaziati in modo da non lasciare spazio bianco in fondo alla pagina.

```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forzare il numero dei sistemi. Ciò può essere utile in due modi. Il semplice impostare un valore, persino lo stesso valore del numero di sistemi disposti senza modificare la variabile, può far sì che più sistemi riescano a entrare in ogni pagina, perché viene saltato il passaggio di valutazione, dando un valore più adatto per ogni pagina. Inoltre, forzare davvero una riduzione nel numero di sistemi può far risparmiare un'ulteriore pagina. Per esempio, se la formattazione predefinita ha 11 sistemi, la seguente impostazione forzerà la formattazione con 10 sistemi.

```
\paper {
  system-count = #10
}
```

- Forzare il numero delle pagine. Per esempio, la seguente impostazione forzerà la formattazione in due pagine.

```
\paper {
  page-count = #2
}
```

- Evitare (o ridurre) gli oggetti che aumentano la dimensione verticale di un sistema. Per esempio, le parentesi delle volte per i finali alternativi richiedono ulteriore spazio. Se questi finali si estendono per due sistemi, occupano più spazio che se fossero sullo stesso sistema. Altro esempio: le dinamiche che “spuntano fuori” da un sistema possono essere avvicinate al rigo:

```
\relative e' {
  e4 c g\ff c
  e4 c g-\tweak X-offset #-2.7 \ff c
}
```



- Modificare la spaziatura orizzontale tramite SpacingSpanner. Maggiori informazioni in Sezione 4.5.3 [Modifica della spaziatura orizzontale], pagina 593. L'esempio seguente mostra la spaziatura predefinita:

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```


}



L'esempio successivo modifica `common-shortest-duration` da un valore di $1/4$ a uno di $1/2$. La nota di un quarto è la durata più comune e più breve in questo esempio, dunque rendendola più lunga si verifica un effetto “compressione”:

```
\score {
  \relative {
    g'4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner.common-shortest-duration =
        \musicLength 2
    }
  }
}
```



La proprietà `common-shortest-duration` non può essere modificata in modo dinamico, quindi deve essere sempre posta in un blocco `\context` così che sia applicata all'intera partitura.

Vedi anche

Guida alla notazione: Sezione 4.1 [Formattazione della pagina], pagina 552, Sezione 4.5.3 [Modifica della spaziatura orizzontale], pagina 593.

Frammenti: Sezione “Spacing” in *Frammenti di codice*.

5 Modifica delle impostazioni predefinite

LilyPond è progettato per fornire la migliore qualità grafica mantenendo le impostazioni predefinite. Tuttavia può capitare che sia necessario modificare la sua formattazione predefinita. La formattazione è regolata da un gran numero di “pulsanti e interruttori” chiamati ‘proprietà’. Prima di proseguire si consiglia di leggere una guida introduttiva su come accedere a queste proprietà e modificarle: vedi Sezione “Modifica dell’output” in *Manuale di Apprendimento*, nel Manuale di apprendimento. Questo capitolo tratta lo stesso argomento, ma in uno stile più adatto a un manuale.

La descrizione completa delle proprietà che si possono ritoccare si trova in un documento separato: Sezione “la Guida al funzionamento interno” in *Guida al Funzionamento Interno*. Questo manuale elenca tutte le variabili, le funzioni e le opzioni disponibili in LilyPond. Viene fornito come un documento HTML, disponibile on-line (<https://lilypond.org/doc/stable/Documentation/internals/>), ed è anche incluso nel pacchetto della documentazione di LilyPond.

Internamente, LilyPond usa il linguaggio Scheme (un dialetto di LISP) per fornire l’infrastruttura. Modificare le decisioni di formattazione prevede in effetti l’accesso alle funzioni interne del programma, cosa che richiede l’input Scheme. In un file .ly gli elementi Scheme sono introdotti col segno cancelletto #.¹

5.1 Contesti di interpretazione

Questa sezione spiega cosa sono i contesti e come modificarli.

Vedi anche

Manuale di apprendimento: Sezione “Contesti e incisori” in *Manuale di Apprendimento*.

File installati: ly/engraver-init.ly, ly/performer-init.ly.

Frammenti: Sezione “Contexts and engravers” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “Contexts” in *Guida al Funzionamento Interno*, Sezione “Engravers and Performers” in *Guida al Funzionamento Interno*.

5.1.1 Tutto sui contesti

I contesti sono organizzati in modo gerarchico:

Definizioni di output – gerarchia dei contesti

Questa sezione spiega la rilevanza delle definizioni di output quando si lavora coi contesti. Esempi di vere definizioni di output sono illustrati dopo (vedi [Modifica di tutti i contesti dello stesso tipo], pagina 615).

Sebbene la musica scritta in un file possa riferirsi a un certo tipo o nome di un contesto, i contesti vengono creati soltanto quando la musica viene interpretata. LilyPond interpreta la musica in base a una “definizione di output” e lo fa per le varie definizioni di output, producendo un output diverso per ciascuna. La definizione di output che crea l’output grafico si specifica con \layout.

Una definizione di output molto più semplice, usata per produrre l’output Midi, si specifica con \midi. Molte altre definizioni di output sono usate da LilyPond internamente, come quando si usa la combinazione automatica delle parti ([Combinazione automatica delle parti], pagina 194) o si creano citazioni musicali ([Citare altre voci], pagina 222).

¹ Sezione “Scheme tutorial” in *Estendere*, contiene una breve guida sull’inserimento di numeri, liste, stringhe e simboli in Scheme.

Le definizioni di output definiscono la relazione tra i contesti e le loro rispettive impostazioni predefinite. Sebbene la maggior parte dei cambiamenti venga fatta solitamente in un blocco `\layout`, le impostazioni relative al Midi avranno effetto solo se inserite in un blocco `\midi`.

Alcune impostazioni interessano vari tipi di output: per esempio, se `autoBeaming` è disattivato in qualche contesto, le travature contano come melismi al fine di abbinare musica e testo vocale, come descritto in [Durate automatiche delle sillabe], pagina 284. Questo abbinamento viene fatto sia per l'output grafico che per il Midi. Se le modifiche fatte a `autoBeaming` in una definizione di contesto di un blocco `\layout` non sono ripetute nel corrispondente blocco `\midi`, il testo e la musica andranno fuori sincrono nel Midi.

Vedi anche

File installati: `ly/engraver-init.ly`. `ly/performer-init.ly`.

Score – il padre di tutti i contesti

Questo è il contesto del livello superiore. Nessun altro contesto può contenere un contesto Score. Per impostazione predefinita, il contesto Score gestisce l'amministrazione delle indicazioni di tempo e garantisce che elementi come le chiavi, le indicazioni di tempo e le armature di chiave siano allineate da rigo a rigo.

Un contesto Score viene istanziato implicitamente quando viene elaborato un blocco `\score {...}`.

Contesti del livello superiore – contenitori di righi

StaffGroup

Raggruppa i righi e aggiunge una parentesi quadra sul lato sinistro, che raggruppa insieme i righi. Le stanghette dei righi in esso contenuti sono connessi verticalmente. *StaffGroup* è semplicemente un insieme di righi, con una parentesi quadra all'inizio e le stanghette che li attraversano.

ChoirStaff

Identico a *StaffGroup* eccetto che le stanghette dei righi in esso contenuti non sono connesse verticalmente.

GrandStaff

Un gruppo di righi, con una parentesi graffa sul lato sinistro, che raggruppa insieme i righi. Le stanghette dei righi in esso contenuti sono connessi verticalmente.

PianoStaff

Identico a *GrandStaff*, ma con il supporto ai nomi degli strumenti a sinistra di ogni sistema.

Contesti del livello intermedio – rigi

Staff

Gestisce chiavi, stanghette, armature di chiave, alterazioni. Può contenere contesti *Voice*.

RhythmicStaff

Come *Staff* ma per le ritmiche. Le altezze vengono ignorate e le note appaiono su una linea. L'output MIDI conserva le altezze inalterate.

TabStaff

Contesto per generare l'intavolatura (o tablatura). La forma predefinita è quella dell'intavolatura per chitarra, con sei linee.

DrumStaff

Contesto per gli strumenti percussivi. Può contenere *DrumVoice*.

VaticanaStaff

Identico a *Staff*, a parte il fatto che è progettato per rappresentare un brano in stile gregoriano.

MensuralStaff

Identico a *Staff*, a parte il fatto che è progettato per rappresentare un brano in stile mensurale.

Contesti del livello inferiore – voci

I contesti al livello della voce inizializzano alcune proprietà e avviano gli incisori adatti. Un contesto del livello inferiore è privo di `defaultchild`. Sebbene sia possibile far sì che accetti/contenga sottocontesti, questi possono essere creati e inseriti solo esplicitamente.

Voice

Corrisponde a una voce di un rigo. Questo contesto gestisce la conversione di segni di dinamica, gambi, travature, apici e pedici, legature di portamento e di valore e pause. Deve essere istanziata esplicitamente se si hanno molteplici voci nello stesso rigo.

VaticanaVoice

Identico a *Voice*, a parte il fatto che è progettato per rappresentare un brano in stile gregoriano.

MensuralVoice

Identico a *Voice*, con delle modifiche per rappresentare un brano in stile mensurale.

Lyrics

Corrisponde a una voce con testo vocale. Gestisce la stampa di una singola linea di testo vocale.

DrumVoice

Il contesto della voce usato in un rigo per percussioni.

FiguredBass

Il contesto in cui sono creati gli oggetti *BassFigure* a partire dall'input inserito in modalità `\figuremode`.

TabVoice

Il contesto della voce usato all'interno di un contesto *TabStaff*. Solitamente omissso così che sia creato implicitamente.

CueVoice

Un contesto della voce usato per rappresentare note a dimensione ridotta, inteso soprattutto per aggiungere citazioni in corpo più piccolo a un rigo, vedi [Formattazione delle notine], pagina 226. Solitamente omissso così che sia creato implicitamente.

ChordNames

Crea e dispone i nomi degli accordi.

5.1.2 Creazione e citazione di un contesto

LilyPond crea i contesti del livello inferiore automaticamente se incontra un'espressione musicale prima di un contesto adatto, ma questo approccio di solito funziona soltanto per partiture semplici o frammenti musicali simili a quelli della documentazione. Per partiture più complesse si consiglia di specificare tutti i contesti esplicitamente coi comandi `\new` o `\context`. La sintassi di questi due comandi è molto simile:

```
[\new | \context] Contesto [ = nome] [espressione-musicale]
```

dove si può usare `\new` o `\context`. *Contesto* è il tipo di contesto che deve essere creato, *nome* è un nome opzionale da assegnare al contesto che si sta creando e *espressione-musicale* è una

singola espressione musicale che deve essere interpretata dagli incisori e dai performer in questo contesto.

Il prefisso `\new` senza un nome viene usato comunemente per creare partiture con molti righi:

```
<<
  \new Staff \relative {
    % lascia che il contesto Voice sia creato implicitamente
    c' '4 c
  }
  \new Staff \relative {
    d' '4 d
  }
>>
```



e per mettere varie voci in un rigo:

```
\new Staff <<
  \new Voice \relative {
    \voiceOne
    c' '8 c c4 c c
  }
  \new Voice \relative {
    \voiceTwo
    g' '4 g g g
  }
>>
```



`\new` deve essere sempre usato per specificare contesti senza nome.

La differenza tra `\new` e `\context` sta nell'azione presa:

- `\new` con o senza un nome creerà sempre un contesto del tutto nuovo e distinto, anche se ne esiste già uno con lo stesso nome:

```
\new Staff <<
  \new Voice = "A" \relative {
    \voiceOne
    c' '8 c c4 c c
  }
  \new Voice = "A" \relative {
    \voiceTwo
    g' '4 g g g
  }
>>
```



- `\context` seguito da un nome creerà un contesto distinto solo se non esiste già un contesto dello stesso tipo con lo stesso nome nella stessa gerarchia di contesto. Altrimenti sarà preso come riferimento per quel contesto precedentemente creato, e la sua espressione musicale verrà passata a quel contesto per la sua interpretazione.

I contesti con nome possono essere utili in casi particolari come il testo vocale o il basso continuo, come è spiegato in [Lavorare con testo e variabili], pagina 293, e Sezione “Modelli per gruppi vocali” in *Manuale di Apprendimento* per il primo e in [Visualizzazione del basso continuo], pagina 448, per il secondo.

In generale, una possibile applicazione dei contesti con nome è la separazione di formattazione della partitura e contenuto musicale. Le seguenti due forme sono entrambe valide:

```
\score {
  <<
    % formattazione della partitura
    \new Staff <<
      \new Voice = "one" {
        \voiceOne
      }
      \new Voice = "two" {
        \voiceTwo
      }
    >>

    % contenuto musicale
    \context Voice = "one" {
      \relative {
        c' '4 c c c
      }
    }
    \context Voice = "two" {
      \relative {
        g'8 g g4 g g
      }
    }
  >>
}
```



```
\score {
  <<
    % formattazione della partitura
    \new Staff <<
      \context Voice = "one" {
        \voiceOne
      }
      \context Voice = "two" {
```

```

        \voiceTwo
      }
    >>

    % contenuto musicale
    \context Voice = "one" {
      \relative {
        c''4 c c c
      }
    }
    \context Voice = "two" {
      \relative {
        g'8 g g4 g g
      }
    }
  >>
}

```



Altrimenti si possono usare le variabili per ottenere un risultato simile. Vedi Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*.

- `\context` senza nome corrisponderà al primo di qualsiasi contesto precedentemente creato dello stesso tipo nella stessa gerarchia di contesto, anche uno a cui è stato assegnato un nome, e la sua espressione musicale sarà passata a quel contesto per la sua interpretazione. Questa forma è raramente utile. Tuttavia, si usa `\context` senza nome e senza espressione musicale per impostare il contesto in cui una procedura Scheme specificata con `\applyContext` viene eseguita:

```

\new Staff \relative {
  c'1
  \context Timing
  \applyContext #(lambda (ctx)
                    (newline)
                    (display (ly:context-current-moment ctx)))
  c1
}

```

Un contesto deve essere nominato se deve essere citato successivamente, per esempio quando il testo vocale è associato alla musica:

```

\new Voice = "tenore" musica
...
\new Lyrics \lyricsto "tenore" testo

```

Maggiori informazioni sull’associazione del testo vocale alla musica in [Durate automatiche delle sillabe], pagina 284.

Le proprietà di tutti i contesti di un certo tipo possono essere modificate in un blocco `\layout` (con una sintassi diversa), vedi [Modifica di tutti i contesti dello stesso tipo], pagina 615. Questo costrutto fornisce anche un mezzo per mantenere le istruzioni di formattazione separate dal contenuto musicale. Se occorre modificare un solo contesto, bisogna usare un blocco `\with`, vedi [Modifica di un solo contesto specifico], pagina 618.

Vedi anche

Manuale di apprendimento: Sezione “Organizzare i brani con le variabili” in *Manuale di Apprendimento*.

Guida alla notazione: [Modifica di un solo contesto specifico], pagina 618, [Durate automatiche delle sillabe], pagina 284.

5.1.3 Conservazione di un contesto

I contesti vengono solitamente terminati nel primo momento musicale in cui non hanno niente da fare. Quindi i contesti Voice muoiono appena non contengono eventi; i contesti Staff muoiono appena tutti i contesti Voice al loro interno non contengono eventi; etc. Ciò può causare difficoltà se contesti precedenti, che sono stati terminati, devono essere richiamati; per esempio, quando si cambia il rigo col comando `\change`, quando si associa il testo a una voce col comando `\lyricsto` o quando si aggiungono ulteriori eventi musicali a un contesto precedente.

C'è un'eccezione a questa regola generale: all'interno di un costrutto `{...}` (musica sequenziale), la nozione di «contesto corrente» scenderà di livello ogni volta che un elemento della sequenza termina in un sottocontesto del precedente contesto corrente. Ciò evita la falsa creazione di contesti impliciti in varie situazioni, ma significa che il primo contesto in cui si scende verrà conservato fino alla fine dell'espressione.

Al contrario, i contesti di un costrutto `<<...>>` (musica simultanea) non sono portati avanti, dunque racchiudendo un comando di creazione di contesto in un'ulteriore coppia di `<<...>>` impedirà che il contesto persista in tutta la sequenza `{...}`.

Un contesto può essere tenuto attivo assicurandosi che abbia qualcosa da fare in ogni momento musicale. I contesti Staff sono tenuti attivi assicurandosi che una delle loro voci sia conservata. Un modo per farlo consiste nell'aggiungere pause spaziatrici a una voce in parallelo con la musica vera. Queste devono essere aggiunte a ogni contesto Voice da tenere attivo. Se si usano sporadicamente varie voci, è più sicuro tenerle attive invece di tentare di affidarsi alle eccezioni menzionate sopra.

Nell'esempio seguente, sia la voce A che la voce B sono mantenute attive in questo modo per la durata del brano:

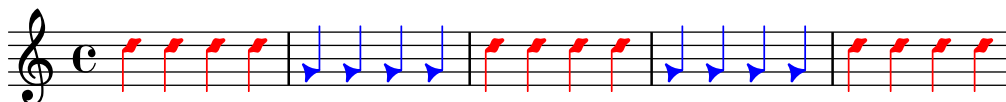
```
musicA = \relative { d'4 d d d }
musicB = \relative { g'4 g g g }
keepVoicesAlive = {
  <<
    \new Voice = "A" { s1*5 } % Tiene la voce "A" attiva per 5 battute
    \new Voice = "B" { s1*5 } % Tiene la voce "B" attiva per 5 battute
  >>
}

music = {
  \context Voice = "A" {
    \voiceOneStyle
    \musicA
  }
  \context Voice = "B" {
    \voiceTwoStyle
    \musicB
  }
  \context Voice = "A" { \musicA }
  \context Voice = "B" { \musicB }
  \context Voice = "A" { \musicA }
```



```
}
```

```
\score {
  \new Staff <<
    \keepVoicesAlive
    \music
  >>
}
```



L'esempio seguente mostra come scrivere una linea melodica sporadica con testo vocale usando questo approccio. Ovviamente, in una situazione reale la melodia e l'accompagnamento consisterebbero di varie e diverse sezioni.

```
melodia = \relative { a'4 a a a }
accompagnamento = \relative { d'4 d d d }
parole = \lyricmode { Queste parole seguo -- no la mel -- o -- dia }
\score {
  <<
    \new Staff = "musica" {
      <<
        \new Voice = "melodia" {
          \voiceOne
          s1*4 % Tiene la voce "melodia" attiva per 4 battute
        }
        {
          \new Voice = "accompagnamento" {
            \voiceTwo
            \accompagnamento
          }
          <<
            \context Voice = "melodia" { \melodia }
            \context Voice = "accompagnamento" { \accompagnamento }
          >>
          \context Voice = "accompagnamento" { \accompagnamento }
          <<
            \context Voice = "melodia" { \melodia }
            \context Voice = "accompagnamento" { \accompagnamento }
          >>
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = "musica" }
    \lyricsto "melodia" { \parole }
  >>
}
```



Un modo alternativo, migliore in molte circostanze, è quello di mantenere la linea melodica attiva semplicemente includendo le note spaziatrici in modo che si allineino correttamente con l'accompagnamento:

```
melodia = \relative {
  s1 % salta una battuta
  a'4 a a a
  s1 % salta una battuta
  a4 a a a
}
accompagnamento = \relative {
  d'4 d d d
  d4 d d d
  d4 d d d
  d4 d d d
}
parole = \lyricmode { Queste parole seguono -- no la mel -- o -- dia }

\score {
  <<
    \new Staff = "musica" {
      <<
        \new Voice = "melodia" {
          \voiceOne
          \melodia
        }
        \new Voice = "accompagnamento" {
          \voiceTwo
          \accompagnamento
        }
      >>
    }
    \new Lyrics \with { alignAboveContext = "musica" }
    \lyricsto "melodia" { \parole }
  >>
}
```



5.1.4 Modifica dei componenti aggiuntivi di un contesto

I contesti della notazione (come `Score` e `Staff`) non contengono solo le proprietà, ma anche dei componenti aggiuntivi chiamati “engraver” (incisori) che creano gli elementi della notazione. Per esempio, il contesto `Voice` contiene l’incisore `Note_heads_engraver` e il contesto `Staff` contiene l’incisore `Key_engraver`.

Una descrizione completa di ogni componente aggiuntivo si trova in Guida al funzionamento interno \mapsto Translation \mapsto Engravers. Ogni contesto descritto in Guida al funzionamento interno \mapsto Translation \mapsto Context, elenca gli incisori usati per quel contesto.

Può essere utile sperimentare questi componenti aggiuntivi. Per farlo si avvia un nuovo contesto con `\new` o `\context` e si modifica:

```

\new contesto \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ...musica...
}

```

dove ogni ... deve essere sostituito dal nome di un incisore. Il comando `\remove` toglie l'incisore, mentre `\consists` lo mantiene. Ecco un semplice esempio che toglie l'incisore dell'indicazione di tempo (`Time_signature_engraver`) e quello della chiave (`Clef_engraver`) da un contesto `Staff`:

```

<<
  \new Staff \relative {
    f'2 g
  }
  \new Staff \with {
    \remove Time_signature_engraver
    \remove Clef_engraver
  } \relative {
    f'2 g2
  }
>>

```



Nel secondo rigo non ci sono indicazioni di tempo né simboli della chiave. Questo è un metodo piuttosto grezzo per far scomparire gli oggetti, dato che ha effetto sull'intero rigo. Questo metodo influenza anche la spaziatura, cosa che potrebbe non essere desiderabile. Metodi più sofisticati per nascondere gli oggetti sono mostrati in Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*.

L'esempio successivo mostra un'applicazione pratica. Le stanghette e le indicazioni di tempo sono normalmente sincronizzate al livello dell'intera partitura. Ciò viene fatto da `Timing_translator` e `Default_bar_line_engraver`. Questo componente aggiuntivo gestisce le indicazioni di tempo, il posizionamento nella misura, etc. Spostando questi incisori dal contesto `Score` al contesto `Staff`, possiamo ottenere una partitura in cui ogni rigo ha la sua indicazione di tempo.

```

\score {
  <<
    \new Staff \with {
      \consists Timing_translator
    }
    \relative {
      \time 3/4
      c''4 c c c c c
    }
  >>
}

```

```

    }
    \new Staff \with {
      \consists Timing_translator
    }
    \relative {
      \time 2/4
      c' '4 c c c c c
    }
  >>
  \layout {
    \context {
      \Score
      \remove Timing_translator
    }
  }
}

```



Problemi noti e avvertimenti

L'ordine in cui vengono specificati gli incisori è l'ordine in cui vengono richiamati per compiere la loro elaborazione. Di solito l'ordine in cui gli incisori sono specificati non conta, ma in pochi casi speciali l'ordine è importante, per esempio quando un incisore scrive una proprietà e un altro la legge o quando un incisore crea un grob e un altro lo deve elaborare.

I seguenti ordini sono importanti:

- l'incisore `Bar_engraver` deve essere primo di solito,
- l'incisore `New_fingering_engraver` deve precedere l'incisore `Script_column_engraver`,
- `Timing_translator` deve precedere l'incisore `Bar_number_engraver`.

Vedi anche

File installati: `ly/engraver-init.ly`.

5.1.5 Modifica delle impostazioni predefinite di un contesto

Le proprietà dei contesti e dei grob possono essere modificate con i comandi `\set` e `\override`, come è spiegato in Sezione 5.3 [Modifica delle proprietà], pagina 627. Questi comandi creano eventi musicali, rendendo effettivi i cambiamenti nel momento temporale in cui è elaborata la musica.

Questa sezione spiega invece come cambiare i valori *predefiniti* delle proprietà dei contesti e dei grob nel momento in cui viene creato il contesto. Esistono due modi per farlo. Uno modifica i valori predefiniti in tutti i contesti di un certo tipo, l'altro modifica i valori predefiniti soltanto in una certa istanza di un contesto.

Modifica di tutti i contesti dello stesso tipo

Le impostazioni di contesto predefinite da usare per l'output grafico nei contesti `Score`, `Staff`, `Voice` e in altri contesti possono essere specificate in un blocco `\context` compreso in un qualsiasi blocco `\layout`.

Le impostazioni per l'output Midi invece devono essere specificate separatamente in blocchi `\midi` (vedi [Definizioni di output – gerarchia dei contesti], pagina 605).

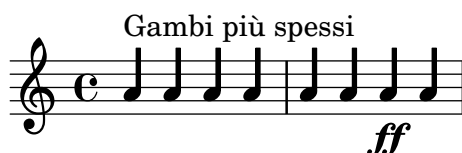
Il blocco `\layout` deve trovarsi all'interno del blocco `\score` al quale si vuole applicare, dopo la musica.

```
\layout {
  \context {
    \Voice
    [impostazioni di contesto per tutti i contesti Voice]
  }
  \context {
    \Staff
    [impostazioni di contesto per tutti i contesti Staff]
  }
}
```

Si possono specificare i seguenti tipi di impostazioni:

- Un comando `\override`, ma col nome del contesto omissso

```
\score {
  \relative {
    a'4~"Gambi più spessi" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      \override Stem.thickness = #4.0
    }
  }
}
```



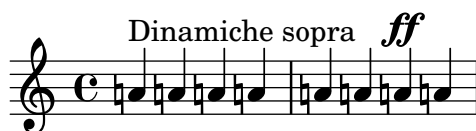
- Impostando direttamente una proprietà di contesto

```
\score {
  \relative {
    a'4~"Tipo di carattere più piccolo" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Staff
      fontSize = #-4
    }
  }
}
```



- Un comando predefinito come `\dynamicUp` o un'espressione musicale come `\accidentalStyle dodecaphonic`

```
\score {
  \relative {
    a'4^"Dinamiche sopra" a a a
    a4 a a\ff a
  }
  \layout {
    \context {
      \Voice
      \dynamicUp
    }
    \context {
      \Staff
      \accidentalStyle dodecaphonic
    }
  }
}
```



- Una variabile definita dall'utente contenente un blocco `\with`; il blocco `\with` è spiegato approfonditamente in [Modifica di un solo contesto specifico], pagina 618.

```
StaffDefaults = \with {
  fontSize = #-4
}

\score {
  \new Staff {
    \relative {
      a'4^"Tipo di carattere più piccolo" a a a
      a4 a a a
    }
  }
  \layout {
    \context {
      \Staff
      \StaffDefaults
    }
  }
}
```



I comandi che impostano una proprietà possono essere posti in un blocco `\layout` senza doverli racchiudere in un blocco `\context`. Così facendo si ottiene lo stesso risultato che si otterrebbe se si includessero gli stessi comandi all'inizio di ogni contesto del tipo specificato. Se non è specificato alcun contesto, avranno effetto su *qualsiasi* contesto di basso livello (vedi

[Contesti del livello inferiore – voci], pagina 607). La sintassi di un comando di impostazione della proprietà in un blocco `\layout` è identico a quello che si userebbe in mezzo alle note.

```
\score {
  \new Staff {
    \relative {
      a'4^"Tipo di carattere più piccolo" a a a
      a4 a a a
    }
  }
  \layout {
    \accidentalStyle dodecaphonic
    \set fontSize = #-4
    \override Voice.Stem.thickness = #4.0
  }
}
```



Modifica di un solo contesto specifico

Le proprietà di contesto di una sola istanza di un contesto specifico possono essere cambiate in un blocco `\with`. Tutte le altre istanze di contesto dello stesso tipo manterranno le impostazioni predefinite di LilyPond e saranno modificate da qualsiasi blocco `\layout` che le riguarda. Il blocco `\with` deve essere posto subito dopo il comando `\new tipo-contesto`:

```
\new Staff \with { [impostazioni di contesto per questa istanza di contesto soltanto] }
{
  ...
}
```

Altrimenti, se la musica viene inserita usando la forma breve dei comandi che specificano la modalità di inserimento, per esempio `\chords` invece di `\chordmode`, il comando `\with` deve essere posto subito dopo il comando che specifica la modalità:

```
\chords \with { [impostazioni di contesto per questo contesto (implicito) soltanto] }
{
  ...
}
```

perché è il contesto implicito creato da queste forme brevi che deve essere modificato. Vale la stessa considerazione per le altre forme brevi che indicano la modalità di inserimento (`\drums`, `\figures`), vedi Sezione 5.4.1 [Modalità di inserimento], pagina 641.

Le ‘modifiche di contesto’, essendo specificate in blocchi `\with` che si trovano all’interno della musica, avranno effetto su *tutti* gli output (quello grafico e il Midi), diversamente da quanto avviene per le modifiche inserite in una definizione di output.

Si possono specificare i seguenti tipi di impostazioni:

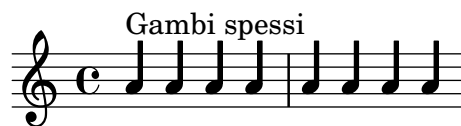
- Un comando `\override`, ma col nome di contesto omissso

```
\score {
  \new Staff {
    \new Voice \with { \override Stem.thickness = #4.0 }
    {
      \relative {
        a'4^"Gambi spessi" a a a
      }
    }
  }
}
```

```

      a4 a a a
    }
  }
}

```



- Impostando direttamente una proprietà di contesto

```

\score {
  <<
    \new Staff {
      \relative {
        a'4~"Tipo di carattere predefinito" a a a
        a4 a a a
      }
    }
    \new Staff \with { fontSize = #-4 }
    {
      \relative {
        a'4~"Tipo di carattere più piccolo" a a a
        a4 a a a
      }
    }
  >>
}

```



- Un comando predefinito come `\dynamicUp`

```

\score {
  <<
    \new Staff {
      \new Voice {
        \relative {
          a'4~"Dinamiche sotto" a a a
          a4 a a\ff a
        }
      }
    }
    \new Staff \with { \accidentalStyle dodecapronic }
    {
      \new Voice \with { \dynamicUp }
      {
        \relative {

```



```

a'4~"Dinamiche sopra" a a a
a4 a a\ff a
}
}
}
>>
}

```



Vedi anche

Guida alla notazione: Sezione 5.4.1 [Modalità di inserimento], pagina 641.

Ordine di precedenza

Il valore di una proprietà da applicare in un certo momento viene determinato nel modo seguente:

- se un comando `\override` o `\set` nel flusso dell'input (le note) è attivo viene usato quel valore,
- altrimenti viene usato il valore predefinito preso da una dichiarazione `\with` nella dichiarazione di iniziazione del contesto,
- altrimenti viene usato il valore predefinito preso dal blocco `\context` appropriato più recente nei blocchi `\layout` o `\midi`,
- altrimenti viene usato il valore predefinito in LilyPond.

Vedi anche

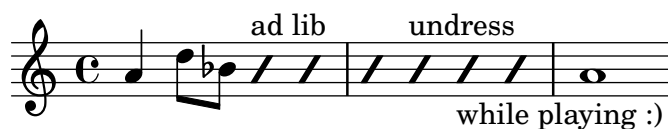
Manuale di apprendimento: Sezione “Modificare le proprietà di contesto” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.1.1 [Tutto sui contesti], pagina 605, [Contesti del livello inferiore – voci], pagina 607, Sezione 5.3.2 [Il comando `\set`], pagina 628, Sezione 5.3.3 [Il comando `\override`], pagina 630, Sezione 4.2.1 [Il blocco `\layout`], pagina 563.

5.1.6 Definizione di nuovi contesti

Contesti specifici, come `Staff` e `Voice`, sono creati a partire da semplici mattoncini. È possibile creare nuovi tipi di contesto con diverse combinazioni di incisori.

Il prossimo esempio mostra come costruire un diverso tipo di contesto `Voice` da zero. Sarà simile a `Voice`, ma stamperà soltanto teste di nota a forma di barra posizionate al centro. Può essere usato per indicare l'improvvisazione nei brani jazz:



Queste impostazioni sono definite in un blocco `\context` compreso in un blocco `\layout`:

```
\layout {
```

```

\context {
  ...
}

```

L'input di esempio che segue sostituisce i ... del frammento precedente.

Per prima cosa occorre definire un nome per il nuovo contesto:

```
\name ImproVoice
```

Essendo simile al contesto `Voice`, ci servono comandi che funzionano in contesti `Voice` esistenti per far sì che continuino a funzionare. Per farlo si assegna al nuovo contesto un alias di `Voice`,

```
\alias Voice
```

Il contesto stamperà note e testi di spiegazione, dunque dobbiamo aggiungere gli incisori che forniscono queste funzionalità, più l'incisore che raggruppa in colonne le note, i gambi e le pause che si trovano nello stesso momento musicale:

```

\consists Note_heads_engraver
\consists Text_engraver
\consists Rhythmic_column_engraver

```

Le teste di nota devono essere poste tutte sulla linea centrale:

```

\consists Pitch_squash_engraver
squashedPosition = #0

```

L'incisore `Pitch_squash_engraver` modifica le teste di nota (create dall'incisore `Note_heads_engraver`) e imposta la loro posizione verticale sul valore di `squashedPosition`, in questo caso 0, la linea centrale.

Le note appaiono come una barra e non hanno gambi:

```

\override NoteHead.style = #'slash
\hide Stem

```

Tutti questi componenti aggiuntivi devono comunicare sotto il controllo del contesto. I meccanismi con cui i contesti comunicano sono stabiliti dichiarando il tipo di contesto con `\type`. In un blocco `\layout`, la maggior parte dei contesti sarà del tipo `Engraver_group`. Alcuni contesti speciali e i contesti nei blocchi `\midi` usano altri tipi di contesto. Copiare e modificare una definizione di contesto esistente comprenderà anche la definizione del tipo. Poiché questo esempio crea una definizione da zero, deve essere specificato in modo esplicito.

```
\type Engraver_group
```

Mettendo tutte queste parti insieme otteniamo:

```

\context {
  \name ImproVoice
  \type Engraver_group
  \consists Note_heads_engraver
  \consists Text_engraver
  \consists Rhythmic_column_engraver
  \consists Pitch_squash_engraver
  squashedPosition = #0
  \override NoteHead.style = #'slash
  \hide Stem
  \alias Voice
}

```

I contesti formano gerarchie. Vogliamo posizionare il contesto ImproVoice all'interno del contesto Staff, proprio come i normali contesti Voice. Cambiamo quindi la definizione di Staff col comando `\accepts`,

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Spesso quando si riutilizza una definizione di contesto esistente, il contesto risultante può essere usato in qualsiasi situazione in cui il contesto originale sarebbe stato utile.

```
\layout {
  ...
  \inherit-acceptability a da
}
```

farà sì che i contesti del tipo *a* siano accettati da tutti i contesti che accettano anche *da*. Per esempio, usando

```
\layout {
  ...
  \inherit-acceptability "ImproVoice" "Voice"
}
```

verrà aggiunto un `\accepts` per ImproVoice a entrambe le definizioni di Staff e RhythmicStaff.

L'opposto di `\accepts` è `\denies`, che è talvolta necessario se si riusano definizioni di contesto esistenti.

Sistemando i pezzi necessari in un blocco `\layout` ci porta a:

```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \inherit-acceptability "ImproVoice" "Voice"
}
```

Quindi l'output all'inizio di questa parte può essere inserito così:

```
\relative {
  a'4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"undress"
    c c_"while playing :)")
  }
  a1
}
```

Per completare questo esempio, le modifiche che alterano la gerarchia dei contesti devono essere ripetute in un blocco `\midi` per far sì che l'output Midi dipenda dalle stesse relazioni di contesto.

Vedi anche

Guida al funzionamento interno: Sezione “Note_heads_engraver” in *Guida al Funzionamento Interno*, Sezione “Text_engraver” in *Guida al Funzionamento Interno*, Sezione “Rhythmic_column_engraver” in *Guida al Funzionamento Interno*, Sezione “Pitch_squash_engraver” in *Guida al Funzionamento Interno*.

5.1.7 Ordine di disposizione dei contesti

Normalmente i contesti in un sistema sono posizionati dall'alto verso il basso nell'ordine in cui sono incontrati nel file di input. Quando i contesti sono annidati, il contesto più esterno includerà i contesti annidati al suo interno come specificato nel file di input solo se i contesti interni sono inclusi nell'elenco “accepts” del contesto più esterno; altrimenti saranno riposizionati sotto il contesto più esterno invece di essere annidati al suo interno.

L'elenco “accepts” di un contesto può essere modificato coi comandi `\accepts` o `\denies`. `\accepts` aggiunge un contesto all'elenco “accepts” mentre `\denies` lo rimuove dall'elenco.

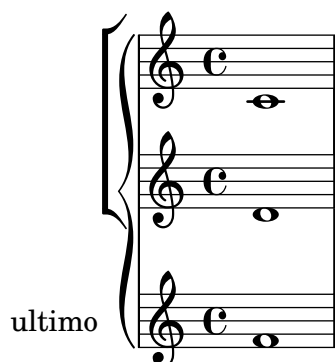
Per esempio, un gruppo di righe racchiusi da una parentesi quadra non si trova solitamente all'interno di un gruppo di righe con stanghette connesse e racchiusi da una graffa, e `GrandStaff` non accetta al suo interno `StaffGroup` per impostazione predefinita.

```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = ultimo f'1 }
  >>
}
```



Tuttavia col comando `\accepts` si può aggiungere `StaffGroup` al contesto `GrandStaff`:

```
\score {
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { c'1 }
      \new Staff { d'1 }
    >>
    \new Staff { \set Staff.instrumentName = ultimo f'1 }
  >>
  \layout {
    \context {
      \GrandStaff
      \accepts StaffGroup
    }
  }
}
```



`\denies` si usa soprattutto quando un nuovo contesto è basato su un altro, ma l’annidamento richiesto differisce. Per esempio, il contesto `VaticanaStaff` è basato sul contesto `Staff`, ma col contesto `VaticanaVoice` che sostituisce il contesto `Voice` nell’elenco “accepts”.

Nota bene: un contesto verrà creato implicitamente e senza avviso se si incontra un comando quando non c’è un contesto adatto disponibile per contenerlo.

Nella definizione di un contesto, il tipo di sottocontesto da creare implicitamente viene specificato con `\defaultchild`. Vari eventi musicali richiedono un contesto “di base”: quando si incontra tale evento, vengono creati ricorsivamente i sottocontesti finché non si raggiunge un contesto privo di impostazione ‘`defaultchild`’.

La creazione implicita di contesti può talvolta causare nuovi righi o nuove partiture non attesi. Per evitare questi problemi è consigliabile usare `\new` per creare i contesti esplicitamente.

Talvolta un contesto deve esistere solo per un breve periodo, come nel caso esemplare di un contesto rigo per un’ossia. Per ottenere ciò, di solito si introduce la definizione di contesto nel punto appropriato in parallelo con la sezione corrispondente della musica principale. Per impostazione predefinita, il contesto temporaneo verrà posizionato sotto tutti i contesti esistenti. Per riposizionarlo sopra il contesto chiamato “principale”, dovrebbe essere definito in questo modo:

```
\new Staff \with { alignAboveContext = "principale" }
```

Una situazione simile si pone quando si posiziona un contesto temporaneo per il testo vocale all’interno di una disposizione multirigo come `ChoirStaff`; per esempio, quando si aggiunge una seconda strofa a una sezione ripetuta. Per impostazione predefinita, il contesto temporaneo per il testo vocale verrà posizionato sotto i righi più bassi. Definendo il contesto temporaneo per il testo vocale con `alignBelowContext`, questo può essere posizionato correttamente sotto i contesti della voce (con nome) che contengono la prima strofa.

Esempi che mostrano questo riposizionamento di contesti temporanei possono essere trovati altrove — vedi Sezione “Annidare le espressioni musicali” in *Manuale di Apprendimento*, Sezione 1.6.2 [Modificare singoli righi], pagina 210, e Sezione 2.1.2 [Tecniche specifiche per il testo vocale], pagina 293.

Vedi anche

Manuale di apprendimento: Sezione “Annidare le espressioni musicali” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.6.2 [Modificare singoli righi], pagina 210, Sezione 2.1.2 [Tecniche specifiche per il testo vocale], pagina 293.

Utilizzo: Sezione “Appare un rigo in più” in *Uso del Programma*.

File installati: `ly/engraver-init.ly`.

5.2 Come funziona la Guida al funzionamento interno

5.2.1 Navigazione nella guida al programma

Supponiamo di voler togliere l'indicazione di diteggiatura nel seguente frammento:

c' '-2



Consultando la documentazione sulle istruzioni di diteggiatura (in [Indicazioni di diteggiatura], pagina 235), si nota:

Vedi anche

Guida al funzionamento interno: Sezione “Fingering” in *Guida al Funzionamento Interno*.

La guida del programmatore è disponibile come documento HTML. È caldamente consigliata la lettura della versione HTML, accessibile online o in locale scaricando la documentazione HTML. Questa sezione sarà molto più difficile da comprendere se si usa il manuale PDF.

Seguire il collegamento a Sezione “Fingering” in *Guida al Funzionamento Interno*. In cima alla pagina si vede

Fingering objects are created by the following engraver(s): Sezione “Fingering_engraver” in *Guida al Funzionamento Interno* and Sezione “New_fingering_engraver” in *Guida al Funzionamento Interno*.

Seguendo i collegamenti correlati nella guida del programma, possiamo seguire il flusso di informazione del programma:

- Sezione “Fingering” in *Guida al Funzionamento Interno*: Sezione “Fingering” in *Guida al Funzionamento Interno* objects are created by the following engraver(s): Sezione “Fingering_engraver” in *Guida al Funzionamento Interno*.
- Sezione “Fingering_engraver” in *Guida al Funzionamento Interno*: Music types accepted: Sezione “fingering-event” in *Guida al Funzionamento Interno*
- Sezione “fingering-event” in *Guida al Funzionamento Interno*: Music event type fingering-event is in Music expressions named Sezione “FingeringEvent” in *Guida al Funzionamento Interno*

Questo percorso va in direzione contraria al flusso dell'informazione nel programma: parte dall'output e termina nell'evento di input. Si può anche partire da un evento di input e leggere seguendo il flusso di informazione arrivando infine all'oggetto di output.

La guida al programma può anche essere sfogliata come un normale documento. Contiene capitoli su Music definitions su Sezione “Translation” in *Guida al Funzionamento Interno*, e sul Sezione “Backend” in *Guida al Funzionamento Interno*. Ogni capitolo elenca tutte le definizioni usate e tutte le proprietà che possono essere ritoccate.

5.2.2 Interfacce di formattazione

La pagina HTML che abbiamo incontrato nella sezione precedente descrive l'oggetto di formattazione chiamato Sezione “Fingering” in *Guida al Funzionamento Interno*. Tale oggetto è un simbolo interno alla partitura. Ha delle proprietà che contengono numeri (come gli spessori e le direzioni), ma anche collegamenti a oggetti correlati. Un oggetto di formattazione viene chiamato anche *Grob*, che è un diminutivo di Graphical Object (oggetto grafico). Maggiori dettagli sui Grob si trovano in Sezione “grob-interface” in *Guida al Funzionamento Interno*.

La pagina relativa al grob *Fingering* elenca le definizioni per l'oggetto *Fingering*. Per esempio, la pagina dice

```
padding (dimension, in staff space):
0.5
```

che significa che il numero verrà tenuto a una distanza di almeno 0.5 spazi rigo dalla testa della nota.

Ogni oggetto di formattazione può avere varie funzioni come elemento della notazione o tipografico. Per esempio, l'oggetto *Fingering* ha i seguenti aspetti

- La sua dimensione è indipendente dalla spaziatura orizzontale, diversamente da legature di portamento o travature.
- È un frammento testuale, solitamente molto breve.
- Questo frammento di testo viene rappresentato con un tipo di carattere, diversamente da legature di portamento o travature.
- Orizzontalmente, il centro del simbolo deve essere allineato al centro della testa della nota.
- Verticalmente, il simbolo viene posto accanto alla nota e sul rigo.
- La posizione verticale viene coordinata anche con altri simboli di apici o pedici.

Ognuno di questi aspetti viene registrato nelle cosiddette interfacce (*interface*), che sono elencate in fondo alla pagina Sezione “*Fingering*” in *Guida al Funzionamento Interno*:

This object supports the following interfaces: Sezione “item-interface” in *Guida al Funzionamento Interno*, Sezione “self-alignment-interface” in *Guida al Funzionamento Interno*, Sezione “side-position-interface” in *Guida al Funzionamento Interno*, Sezione “text-interface” in *Guida al Funzionamento Interno*, Sezione “text-script-interface” in *Guida al Funzionamento Interno*, Sezione “font-interface” in *Guida al Funzionamento Interno*, Sezione “finger-interface” in *Guida al Funzionamento Interno*, and Sezione “grob-interface” in *Guida al Funzionamento Interno*.

Facendo clic su uno di questo collegamenti si arriva alla pagina dell'interfaccia del rispettivo oggetto. Ogni interfaccia ha una serie di proprietà. Alcune non sono a disposizione dell'utente ('Internal properties'), ma altre possono essere modificate.

Abbiamo parlato dell'oggetto *Fingering*, ma in realtà non si tratta di niente di troppo complesso. Il file di inizializzazione (vedi Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*) `scm/define-grobs.scm` mostra l'anima di questo “oggetto”,

```
(Fingering
 . ((padding . 0.5)
    (avoid-slur . around)
    (slur-padding . 0.2)
    (staff-padding . 0.5)
    (self-alignment-X . 0)
    (self-alignment-Y . 0)
    (script-priority . 100)
    (stencil . ,ly:text-interface::print)
    (direction . ,ly:script-interface::calc-direction)
    (font-encoding . fetaText)
    (font-size . -5) ; don't overlap when next to heads.
    (meta . ((class . Item)
              (interfaces . (finger-interface
                             font-interface
                             text-script-interface
                             text-interface
```

```
side-position-interface
self-alignment-interface
item-interface))))))
```

Come si può vedere, l'oggetto `Fingering` non è niente più di un insieme di impostazioni di variabili e la pagina web nella Guida al funzionamento interno è generata direttamente da questa definizione.

5.2.3 Determinazione della proprietà del `grob`

Volevamo cambiare la posizione del **2** in

c' '-2



Dato che il **2** è posizionato verticalmente vicino alla sua nota, dobbiamo fare delle prove con l'interfaccia associata a questo posizionamento, ovvero `side-position-interface`. La pagina relativa a questa interfaccia dice

```
side-position-interface
```

Position a victim object (this one) next to other objects (the support). The property `direction` signifies where to put the victim object relative to the support (left or right, up or down?)

Sotto questa descrizione, la variabile `padding` viene descritta come

```
padding    (dimension, in staff space)
```

Add this much extra space between objects that are next to each other.

Aumentando il valore di `padding`, possiamo spostare la diteggiatura lontana dalla testa di nota. Il seguente comando inserisce “tre spazi rigo” di distanza tra la nota e un segno di diteggiatura:

```
\once \override Voice.Fingering.padding = #3
```

L'inserimento di spazio prima che l'oggetto della diteggiatura venga creato produce il seguente risultato:

```
\once \override Voice.Fingering.padding = #3
```

c' '-2



In questo caso, il contesto per questa modifica è `Voice`. Come è scritto nella pagina sull'incisore Sezione “`Fingering-engraver`” in *Guida al Funzionamento Interno*:

`Fingering-engraver` is part of contexts: ... Sezione “`Voice`” in *Guida al Funzionamento Interno*

5.3 Modifica delle proprietà

5.3.1 Panoramica sulla modifica delle proprietà

Ogni contesto è responsabile della creazione di certi tipi di oggetti grafici. Le impostazioni usate per la rappresentazione grafica di questi oggetti sono anch'esse salvate dal contesto. Cambiando queste impostazioni, si può alterare l'aspetto degli oggetti.

Ci sono due tipi diversi di proprietà salvate nei contesti: le proprietà del contesto e quelle del grob. Le proprietà del contesto sono proprietà che si applicano al contesto nel suo complesso e regolano il modo in cui il contesto stesso viene mostrato. Al contrario, le proprietà del grob si applicano a tipi di grob specifici che verranno mostrati nel contesto.

I comandi `\set` e `\unset` vengono usati per cambiare i valori delle proprietà di contesto. I comandi `\override` e `\revert` servono a cambiare i valori delle proprietà dei grob.

Vedi anche

Guida al funzionamento interno: Sezione “Backend” in *Guida al Funzionamento Interno*, Sezione “All layout objects” in *Guida al Funzionamento Interno*, Sezione “OverrideProperty” in *Guida al Funzionamento Interno*, Sezione “RevertProperty” in *Guida al Funzionamento Interno*, Sezione “PropertySet” in *Guida al Funzionamento Interno*.

Problemi noti e avvertimenti

Il backend non è molto severo nel controllo del tipo delle proprietà di un oggetto. Riferimenti ciclici nei valori Scheme delle proprietà possono causare attese o crash, o entrambi.

5.3.2 Il comando `\set`

Ogni contesto ha un insieme di *proprietà*, variabili contenute in quel contesto. Le proprietà di contesto si cambiano col comando `\set`, che ha la seguente sintassi:

```
\set contesto.proprietà = #valore
```

valore è un oggetto Scheme, ecco perché deve essere preceduto dal carattere `#`.

Le proprietà dei contesti sono solitamente nominate in studlyCaps. Perlopiù regolano la traduzione dalla musica alla notazione, per esempio `localAlterations` (per determinare se stampare le alterazioni), o `measurePosition` (per determinare quando stampare una stanghetta). Le proprietà di contesto possono cambiare nel tempo nel corso dell'interpretazione di un brano musicale; `measurePosition` ne è un ovvio esempio. Le proprietà di contesto si modificano con `\set`.

Per esempio, le pause multiple sono combinate in una singola battuta (come è spiegato in [Compressione delle misure vuote], pagina 231) se la proprietà di contesto `skipBars` viene impostata su `#t`:

```
R1*2
\set Score.skipBars = ##t
R1*2
```



Se l'argomento *contesto* non viene specificato, la proprietà sarà impostata nel contesto di base corrente (solitamente `ChordNames`, `Voice`, `TabVoice` o `Lyrics`).

```
\set Score.autoBeaming = ##f
\relative {
  e' '8 e e e
  \set autoBeaming = ##t
  e8 e e e
```

```

} \
\relative {
  c' '8 c c c c8 c c c
}

```



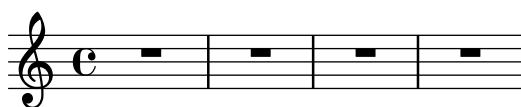
La modifica viene applicata ‘al volo’, durante l’interpretazione della musica, in modo che l’impostazione abbia effetto soltanto sul secondo gruppo di note da un ottavo.

Nota che il contesto più inferiore non contiene sempre la proprietà che si desidera cambiare. Per esempio, il tentativo di impostare la proprietà `skipBars` del contesto di base predefinito, in questo caso `Voice`, non produrrà alcun risultato, perché `skipBars` è una proprietà del contesto `Score`.

```

R1*2
\set skipBars = ##t
R1*2

```



I contesti sono gerarchici, per cui se viene specificato un contesto che ne racchiude altri, per esempio `Staff`, il cambiamento viene applicato anche a tutti i contesti `Voice` del rigo corrente.

Il comando `\unset`:

```
\unset contesto.proprietà
```

viene usato per togliere la definizione di *proprietà* dal *contesto*. Questo comando rimuove la definizione solo se impostata nel *contesto*. Le proprietà impostate nei contesti più esterni non saranno modificate da un `\unset` in un contesto più interno:

```

\set Score.autoBeaming = ##t
\relative {
  \unset autoBeaming
  e' '8 e e e
  \unset Score.autoBeaming
  e8 e e e
} \
\relative {
  c' '8 c c c c8 c c c
}

```



Come in `\set`, non è necessario specificare l’argomento *contesto* per un contesto di fondo, quindi le due dichiarazioni

```

\set Voice.autoBeaming = ##t
\set autoBeaming = ##t

```

sono equivalenti se l’attuale contesto di fondo è `Voice`.

Anteponendo `\once` ai comandi `\set` o `\unset`, l'impostazione verrà applicata soltanto a un singolo momento temporale:

```
c''4
\once \set fontSize = #4.7
c''4
c''4
```



Una descrizione completa di tutte le proprietà di contesto disponibili si trova nella Guida al funzionamento interno, vedi Translation \mapsto Tunable context properties.

Vedi anche

Guida al funzionamento interno: Sezione “Tunable context properties” in *Guida al Funzionamento Interno*.

5.3.3 Il comando `\override`

Esiste un tipo particolare di proprietà di contesto: la descrizione del grob. Le descrizioni dei grob vengono nominate in StudlyCaps (iniziando con lettere maiuscole). Contengono le “impostazioni predefinite” per un certo tipo di grob, come una lista associativa. Consultare `scm/define-grobs.scm` per vedere le impostazioni per ogni descrizione di grob. Le descrizioni di grob si modificano con `\override`.

La sintassi del comando `\override` è

```
\override [contesto.]NomeGrob.proprietà = #valore
```

Per esempio, possiamo aumentare lo spessore del gambo di una nota modificando la proprietà `thickness` dell'oggetto `Stem`:

```
c''4 c''
\override Voice.Stem.thickness = #3.0
c''4 c''
```



Se non viene specificato il contesto in un `\override`, viene usato il contesto di base:

```
\override Staff.Stem.thickness = #3.0
<<
  \relative {
    e''4 e
    \override Stem.thickness = #0.5
    e4 e
  } \
  \relative {
    c''4 c c c
  }
>>
```



Alcune opzioni modificabili vengono chiamate ‘sottoproprietà’ e stanno dentro le proprietà. Per cambiarle, usare i comandi nella forma

```
\override Stem.details.beamed-lengths = #'(4 4 3)
```

o, per modificare le estremità degli estensori, usare una forma simile alle seguenti

```
\override TextSpanner.bound-details.left.text = "left text"
```

```
\override TextSpanner.bound-details.right.text = "right text"
```

Gli effetti di `\override` possono essere annullati con `\revert`.

La sintassi del comando `\revert` è

```
\revert [contesto.]NomeGrob.proprietà
```

Per esempio,

```
\relative {
  c' '4
  \override Voice.Stem.thickness = #3.0
  c4 c
  \revert Voice.Stem.thickness
  c4
}
```



Gli effetti di `\override` e `\revert` si applicano a tutti i grob nel contesto interessato da quel momento in poi:

```
<<
\relative {
  e' '4
  \override Staff.Stem.thickness = #3.0
  e4 e e
} \\\
\relative {
  c' '4 c c
  \revert Staff.Stem.thickness
  c4
}
>>
```



Si può usare `\once` insieme a `\override` o `\revert` per agire solo sul momento corrente:

```
<<
\relative c {
  \override Stem.thickness = #3.0
  e' '4 e e e
} \\\
\relative {
  c' '4
  \once \override Stem.thickness = #3.0
  c4 c c
}
```

```
}
>>
```



Vedi anche

Guida al funzionamento interno: Sezione “Backend” in *Guida al Funzionamento Interno*.

5.3.4 Il comando `\tweak`

Modificare le proprietà dei grob con `\override` fa sì che i cambiamenti siano applicati a tutti i grob in questione in quel contesto nel momento in cui la modifica viene applicata. Talvolta, tuttavia, si può voler applicare le modifiche a un solo grob invece che a tutti i grob del contesto interessato. Per farlo si usa il comando `\tweak`, che ha la seguente sintassi:

```
\tweak [oggetto-formattazione.]proprietà-grob valore
```

oggetto-formattazione è opzionale. Il comando `\tweak` viene applicato all’oggetto musicale che segue immediatamente *valore* nel flusso musicale.

Per un’introduzione alla sintassi e agli usi del comando `tweak` leggere Sezione “Metodi di modifica” in *Manuale di Apprendimento*.

Quando molti elementi simili sono disposti nello stesso momento musicale, non è possibile usare il comando `\override` per modificarne soltanto uno. È in casi come questi che occorre usare il comando `\tweak`. Ecco alcuni elementi che possono apparire più di una volta nello stesso momento musicale:

- teste di note di un accordo
- segni di articolazione su una singola nota
- legature di valore tra note di un accordo
- parentesi di gruppi irregolari che iniziano nello stesso momento

In questo esempio, il colore di una testa di nota e il tipo di testa di un’altra nota sono modificati all’interno di un accordo:

```
< c '
  \tweak color #red
  d '
  g '
  \tweak duration-log #1
  a '
> 4
```



`\tweak` può essere usato per modificare le legature di portamento:

```
\relative { c'-\tweak thickness #5 ( d e f ) }
```



Perché il comando `\tweak` funzioni, deve trovarsi proprio accanto all’oggetto al quale deve essere applicato dopo che il file di input è stato convertito in un flusso musicale. Modificare un

intero accordo non cambia niente perché il suo evento musicale agisce solo come contenitore, mentre tutti gli oggetti della formattazione sono creati dagli eventi interni all' EventChord:

```
\tweak color #red c''4
\tweak color #red <c'' e''>4
<\tweak color #red c'' e''>4
```



Il semplice comando `\tweak` non può essere usato per modificare un oggetto che non sia creato direttamente dall'input. In particolare, non agirà su gambi, travature automatiche o alterazioni, dato che questi sono generati successivamente dagli oggetti della formattazione di NoteHead invece che da elementi musicali nel flusso dell'input.

Tali oggetti della formattazione creati indirettamente possono essere tuttavia modificati usando una forma del comando `\tweak` in cui il nome del grob è indicato esplicitamente:

```
\tweak Stem.color #red
\tweak Beam.color #green c''8 e''
<c'' e'' \tweak Accidental.font-size #-3 ges''>4
```



`\tweak` non può essere usato per modificare le chiavi o le indicazioni di tempo, perché queste vengono separate da qualsiasi comando `\tweak` precedente nel flusso dell'input a causa dell'inserimento automatico di ulteriori elementi richiesti per specificare il contesto.

Si possono usare vari comandi `\tweak` prima di un elemento della notazione e tutti avranno effetto su di esso:

```
c'
-\tweak style #'dashed-line
-\tweak dash-fraction #0.2
-\tweak thickness #3
-\tweak color #red
\glissando
f''
```



Il flusso musicale generato da una sezione di un file di input, compresi gli elementi inseriti automaticamente, può essere esaminato, vedi Sezione “Displaying music expressions” in *Estendere*. Ciò può essere utile nel determinare cosa può essere modificato da un comando `\tweak` o nel valutare come aggiustare l'input per far sì che un `\tweak` sia applicato.

Vedi anche

Manuale di apprendimento: Sezione “Metodi di modifica” in *Manuale di Apprendimento*.

Estendere LilyPond: Sezione “Displaying music expressions” in *Estendere*.

Problemi noti e avvertimenti

Il comando `\tweak` non può essere usato per modificare i punti di controllo di una sola legatura di valore tra tante in un accordo, se non quelli della prima legatura incontrata nel file di input.

5.3.5 `\set` vs. `\override`

I comandi `\set` e `\override` manipolano le proprietà associate ai contesti. In entrambi i casi, le proprietà seguono una *gerarchia dei contesti*; proprietà non impostate in un contesto mostreranno i valori del rispettivo contesto padre.

La durata e il valore di una proprietà di contesto sono dinamici e disponibili soltanto mentre la musica viene interpretata (ovvero ‘reiterata’). Nel momento della creazione del contesto, le proprietà sono inizializzate a partire dalle definizioni corrispondenti (insieme a altre modifiche) di quel contesto. Qualsiasi modifica successiva viene fatta con dei comandi di impostazione della proprietà presenti nella musica stessa.

Le definizioni degli oggetti grafici (o “grob”) sono una categoria *speciale* di proprietà di contesto, perché la loro struttura e uso sono diversi da quelli delle normali proprietà di contesto. Diversamente da quest’ultime, le definizioni dei grob sono suddivise in *proprietà dei grob*.

Inoltre, diversamente dalle normali proprietà di contesto, le definizioni dei grob hanno una loro ‘contabilità’ interna usata per tenere traccia delle loro individuali proprietà e sottoproprietà. Ciò significa che è possibile definire queste parti in contesti diversi e avere sempre a disposizione la definizione complessiva del grob nel momento della sua creazione, assemblata da tutti i pezzi compresi tra il contesto corrente e i suoi contesti superiori.

Un grob viene solitamente creato da un incisore nel momento in cui l’espressione musicale viene interpretata e riceve le sue proprietà iniziali dalla definizione del grob corrente del contesto dell’incisore. L’incisore (o altre parti del ‘backend’ di LilyPond) può successivamente cambiare (o aggiungere) le proprietà iniziali del grob, ma ciò non ha effetto sulla definizione di grob del contesto.

Ciò che LilyPond chiama ‘proprietà del grob’ nel contesto delle modifiche a livello di utente sono in realtà le proprietà della definizione di grob di un *contesto*.

Le definizioni di grob sono manipolate con `\override` e `\revert` e hanno un nome che inizia con una lettera maiuscola (come ‘NoteHead’), mentre le normali proprietà di contesto sono manipolate con `\set` e `\unset` e il loro nome inizia con una lettera minuscola.

I comandi `\tweak` e `\overrideProperty` modificano le proprietà dei grob bypassando completamente tutte le proprietà di contesto. Catturano i grob mentre vengono creati, impostando le loro proprietà per un evento musicale (`\tweak`) o, nel caso di `\overrideProperty`, per una sovrascrittura specifica.

5.3.6 Il comando `\offset`

Sebbene sia possibile impostare le proprietà dei grob su nuovi valori coi comandi `\override`, `\tweak` e `\overrideProperty`, è spesso più opportuno modificare tali proprietà in modo relativo a un valore predefinito. A questo fine è stato creato il comando `\offset`.

La sintassi di `\offset` è

```
[-]\offset proprietà offset elemento
```

Il comando funziona aggiungendo i contenuti di *offset* all’impostazione predefinita della proprietà *proprietà* del grob indicato da *elemento*.

A seconda di come è formulato il comando, `\offset` può comportarsi come un `\tweak` o come un `\override`. Le variazioni d’uso sono trattate dopo aver considerato le proprietà del grob che possono essere usate con `\offset`.

Proprietà che possono essere spostate con `offset`

Molte, ma non tutte, le proprietà dei grob possono essere spostate con `offset`. Se la *proprietà* non può essere spostata con `offset`, l’oggetto resterà invariato e apparirà un avviso. In questi casi bisogna usare `\override` o `\tweak` per modificare l’oggetto.

Si potrebbe procedere per prova e errore e basarsi sugli avvisi per capire cosa può essere spostato con `offset` e cosa no. Tuttavia si può usare un approccio più sistematico.

I seguenti criteri determinano se una proprietà possa essere modificata con `\offset`:

- La proprietà ha un'‘impostazione predefinita’ nella descrizione del grob. Queste proprietà sono elencate, per ogni grob, in Sezione “All layout objects” in *Guida al Funzionamento Interno* (si trovano anche in `scm/define-grobs.scm`).
- La proprietà accetta un valore numerico. I valori numerici comprendono `number`, una lista di `number`, `number-pair` e `number-pair-list`. Le pagine in Sezione “All layout objects” in *Guida al Funzionamento Interno* elencano il tipo di dati tipici di ciascuna proprietà. È irrilevante che l'impostazione predefinita sia una funzione.
- La proprietà non può essere una ‘sottoproprietà’, ovvero una proprietà che risiede all'interno di un'altra proprietà.
- Le proprietà impostate su valori infiniti non possono essere spostate con `offset`, perché non esiste un modo sensato per spostare un infinito positivo o negativo.

Gli esempi seguenti prendono in considerazione varie proprietà grob in relazione ai criteri appena delineati.

- Proprietà che possono essere spostate con `offset`

`Hairpin.height`

Questa proprietà non è una sottoproprietà e è elencata in Sezione “Hairpin” in *Guida al Funzionamento Interno*. Come valore prende ‘dimensione, in spazi rigo’ impostata su 0.6666, ovviamente non un numero infinito.

`Arpeggio.positions`

La pagina Sezione “Arpeggio” in *Guida al Funzionamento Interno* elenca una proprietà `positions` che accetta una ‘coppia di numeri’. Il suo valore predefinito è `ly:arpeggio::positions`, una funzione di callback che verrà elaborata durante la fase di formattazione tipografica per emettere una coppia di numeri per ogni oggetto Arpeggio.

- Proprietà che non possono essere spostate con `offset`

`Hairpin.color`

`color` non è presente in Sezione “Hairpin” in *Guida al Funzionamento Interno*.

`Hairpin.circled-tip`

L'elenco per `Hairpin.circled-tip` in Sezione “Hairpin” in *Guida al Funzionamento Interno* mostra che prende un valore booleano. I booleani sono non numerici.

`Stem.details.lengths`

Benché sia elencato in Sezione “Stem” in *Guida al Funzionamento Interno* e il suo valore predefinito sia un elenco di `number`, si tratta di una ‘sottoproprietà’. Attualmente non sono supportate le ‘proprietà annidate’.

\offset come override

Se *elemento* è un nome di grob come Arpeggio o `Staff.OttavaBracket`, il risultato è un `\override` del tipo di grob specificato.

```
\offset proprietà offset [contesto.]NomeGrob
```

Notare che il trattino iniziale non è *mai* usato nella forma ‘override’, proprio come non è mai usato col comando `\override` stesso.

L'esempio seguente usa la forma ‘override’ per allungare gli arpeggi predefiniti mostrati nella prima misura in modo che coprano del tutto l'ensione degli accordi. Gli arpeggi sono allungati

di mezzo spazio rigo verso l'alto e verso il basso. Viene mostrata anche la stessa operazione fatta sul primo accordo con un normale override della proprietà *positions*. Questo metodo non rispecchia affatto il compito di 'allungare di mezzo spazio rigo', perché le estremità devono essere specificate con coordinate assolute invece che relative. Inoltre, sarebbero necessari override individuali per ciascuno degli altri accordi, dato che variano in dimensione e posizione.

```
arpeggioMusic = {
  <c' e' g'>\arpeggio <a' c'' e''>\arpeggio
  <d' f' a' c''>\arpeggio <c' e' g' b' d'' f'' a''>\arpeggio
}

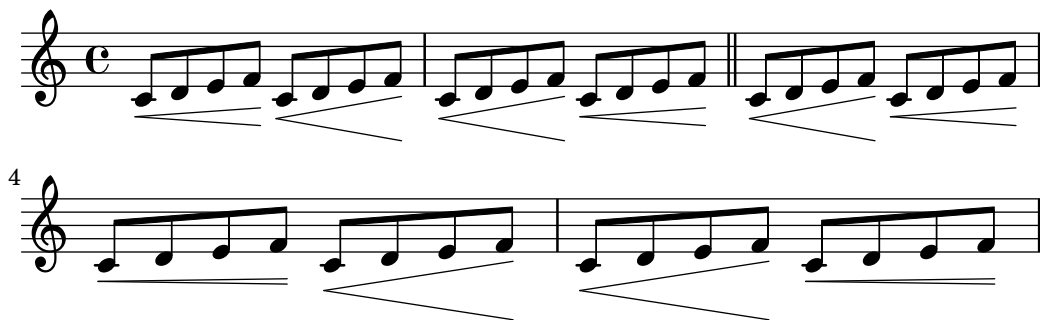
{
  \arpeggioMusic
  \bar "||"
  \offset positions #'(-0.5 . 0.5) Arpeggio
  \arpeggioMusic
  \bar "||"
  \once \override Arpeggio.positions = #'(-3.5 . -0.5)
  <c' e' g'>1\arpeggio
  \bar "||"
}
```



Nel suo uso come 'override', `\offset` può essere preceduto da `\once` o `\temporary` e annullato con `\revert` e la proprietà (vedi Sezione "Intermediate substitution functions" in *Estendere*). Ciò deriva dal fatto che `\offset` in realtà crea un `\override` della proprietà.

```
music = { c'8\< d' e' f'\! }

{
  \music
  \offset height 1 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
  \once \offset height 1 Hairpin
  \music \music
  \bar "||"
  \override Hairpin.height = 0.2
  \music
  \temporary \offset height 2 Hairpin
  \music
  \music
  \revert Hairpin.height
  \music
  \bar "||"
}
```



Proprio come `\override`, la forma ‘override’ di `\offset` può essere usata con `\undo` e `\single`.

```
longStem = \offset length 6 Stem

{
  \longStem c'4 c''' c' c''
  \bar "||"
  \undo \longStem c'4 c''' c' c''
  \bar "||"
  \single \longStem c'4 c''' c' c''
  \bar "||"
}
```



\offset come tweak

Se *elemento* è un’espressione musicale come (o `\arpeggio`, il risultato è la stessa espressione musicale con una modifica di tipo *tweak* applicata.

```
[-]\offset [NomeGrob.]proprietà offset espressione-musicale
```

La sintassi di `\offset` nella sua forma ‘tweak’ è analoga a quella dello stesso comando `\tweak`, sia nell’ordine sia in presenza o assenza del trattino iniziale.

L’esempio seguente usa la forma ‘tweak’ per regolare la posizione verticale dell’oggetto `BreathingSign`. Si confronti questo col normale comando `\tweak` che lo segue. La sintassi è equivalente, ma l’output di `\tweak` è meno intuitivo, perché `BreathingSign.Y-offset` è calcolato dalla linea centrale del pentagramma. Al contrario, non è necessario conoscere come è calcolato `Y-offset` quando si usa `\offset`.

```
{
  c''4
  \breathe
  c''4
  \offset Y-offset 2 \breathe
  c''2
  \tweak Y-offset 3 \breathe
}
```



Nell'esempio precedente, gli oggetti modificati sono stati creati direttamente dall'input dell'utente: il comando `\breathe` è un'istruzione esplicita che restituisce un oggetto `BreathingSign`. Dato che l'ambito del comando era non ambiguo, non è stato necessario specificare il nome dell'oggetto. Ma quando un oggetto viene creato *indirettamente*, bisogna includere il nome del grob. Ciò è valido anche per il comando `\tweak`.

Nell'esempio seguente, l'oggetto Beam viene abbassato di due spazi rigo applicando `\offset` alla proprietà `positions`.

Il primo impiego di `\offset` richiede la presenza del nome del grob, perché nessun elemento dell'input crea esplicitamente la travatura. Nel secondo impiego la travatura viene creata manualmente con l'espressione musicale `[;` dunque il nome del grob non è necessario. L'esempio mostra anche una scorciatoia: un singolo numero viene applicato a entrambi i membri di una coppia di numeri.)

```
{
  c''8 g'' e'' d''
  \offset Beam.positions #'(-2 . -2)
  c''8 g'' e'' d''
  c''8 g'' e'' d''
  c''8-\offset positions #-2 [ g'' e'' d'']
}
```



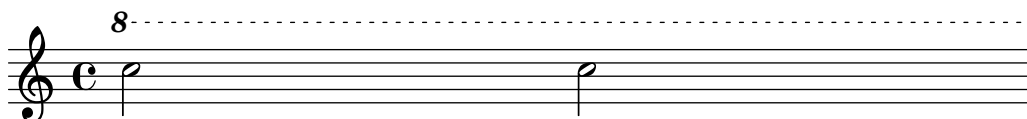
\offset con estensori spezzati

È anche possibile modificare in modo indipendente i segmenti di un estensore che va oltre una o più interruzioni di linea. In questo caso, *offset* prende una lista di valori del tipo di dati richiesto dalla proprietà.

Il comando `\offset` usato in questo modo è simile al comando `\alterBroken`, vedi Sezione 5.5.5 [Modifica degli estensori spezzati], pagina 671. Tuttavia, diversamente da `\alterBroken`, i valori assegnati a `\offset` non sono assoluti, ma relativi.

L'esempio seguente sposta l'oggetto 'spezzato' OttavaBracket attraverso la sua proprietà staff-padding. Dato che la proprietà prende un numero, a *offset* viene fornita una lista di numeri per rendere conto dei due segmenti creati dall'interruzione di linea. Il pezzo di parentesi dell'ottava sulla prima linea non viene modificato perché viene aggiunto 0 al suo valore predefinito di staff-padding. Il segmento sulla seconda linea viene alzato di tre spazi rigo dalla sua altezza predefinita. L'altezza predefinita è 2, anche se non è necessario saperlo per ottenere la posizione desiderata.

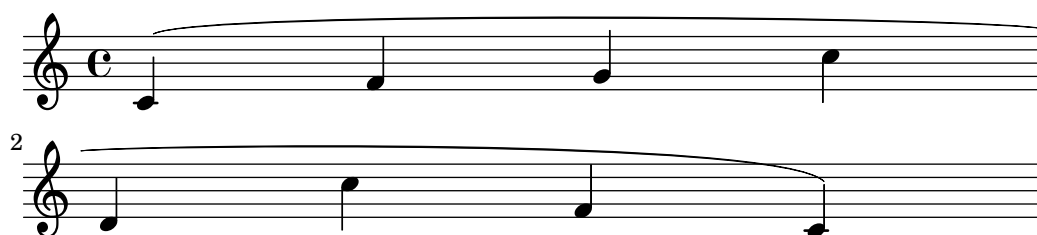
```
{
  \offset staff-padding #'(0 3) Staff.OttavaBracket
  \ottava #1
  c'''2 c'''
  \break
  c'''2 c'''
}
```





L'esempio seguente imita l'effetto del comando `\shape` spostando la proprietà `control-points` dell'oggetto `Slur`. Qui `offset` è una lista di `number-pair-list`, una per ciascun segmento della legatura di portamento. Questo esempio realizza un risultato identico a quello corrispondente in Sezione 5.5.4 [Modifica delle forme], pagina 666.

```
{
  c'4-\offset control-points #'(
    ((0 . 0) (0 . 0) (0 . 0) (0 . 1))
    ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
  ) ( f'4 g' c'
  \break
  d'4 c' f' c')
}
```



5.3.7 Modifica delle liste associative

Alcune proprietà configurabili dall'utente sono rappresentate internamente come *alist* (liste associative), che contengono coppie di *chiavi* e *valori*. La struttura di una lista associativa è:

```
'((chiave1 . valore1)
  (chiave2 . valore2)
  (chiave3 . valore3)
  ...)
```

Se una lista associativa è una proprietà di un grob o una variabile `\paper`, le sue chiavi possono essere modificate individualmente senza influenzare altre chiavi.

Per esempio, per ridurre lo spazio tra righe adiacenti in un gruppo di righe, si usa la proprietà `staff-staff-spacing` del grob `StaffGrouper`. La proprietà è una lista associativa con quattro chiavi: `basic-distance`, `minimum-distance`, `padding` e `stretchability`. Le impostazioni predefinite per questa proprietà sono elencate nella sezione “Backend” della Guida al funzionamento interno (vedi Sezione “StaffGrouper” in *Guida al Funzionamento Interno*):

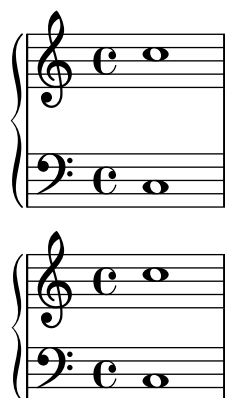
```
'((basic-distance . 9)
  (minimum-distance . 7)
  (padding . 1)
  (stretchability . 5))
```

Un modo per avvicinare i righe è ridurre il valore della chiave `basic-distance` (9) perché corrisponda al valore di `minimum-distance` (7). Per modificare una chiave singola individualmente, usare una *dichiarazione annidata*:

```
% spazio predefinito tra i righe
\new PianoStaff <<
  \new Staff { \clef treble c''1 }
  \new Staff { \clef bass c1 }
```

>>

```
% spazio ridotto tra i righi
\new PianoStaff \with {
  % questa è la dichiarazione annidata
  \override StaffGrouper.staff-staff-spacing.basic-distance = #7
} <<
\new Staff { \clef treble c'1 }
\new Staff { \clef bass c1 }
>>
```



Con una dichiarazione annidata si aggiornerà la chiave specificata (come `basic-distance` nell'esempio precedente) senza modificare alcuna altra chiave già impostata per la stessa proprietà.

Ora immaginiamo di volere che i righi siano più vicini possibile ma senza sovrapporsi. Il modo più semplice per fare ciò è impostare tutte e quattro le chiavi della lista su zero. Tuttavia, non è necessario inserire quattro dichiarazioni annidate, una per ogni chiave. Si può invece ridefinire completamente la proprietà con una sola dichiarazione, attraverso una lista associativa:

```
\new PianoStaff \with {
  \override StaffGrouper.staff-staff-spacing =
    #'((basic-distance . 0)
      (minimum-distance . 0)
      (padding . 0)
      (stretchability . 0))
} <<
\new Staff { \clef treble c'1 }
\new Staff { \clef bass c1 }
>>
```



Nota bene che qualsiasi chiave non elencata esplicitamente nella definizione della lista associativa sarà ripristinata al suo valore *predefinito-quando-non-impostato*. Nel caso di `staff-staff-spacing`, qualsiasi chiave-valore non impostata viene ripristinata su zero (eccetto `stretchability`, che prende il valore di `basic-distance` quando non è impostata). Dunque le due seguenti dichiarazioni sono equivalenti:

```
\override StaffGrouper.staff-staff-spacing =
```

```
#'((basic-distance . 7))

\override StaffGrouper.staff-staff-spacing =
  #'((basic-distance . 7)
      (minimum-distance . 0)
      (padding . 0)
      (stretchability . 7))
```

Una conseguenza (possibilmente non voluta) di questo approccio è l'eliminazione di eventuali impostazioni predefinite impostate in un file di inizializzazione e caricate ogni volta che un file di input viene compilato. Nell'esempio precedente le impostazioni predefinite per padding e minimum-distance (definite in scm/define-grobs.scm) sono ripristinate ai loro valori predefiniti-quando-non-impostati (zero per entrambe le chiavi). La definizione di una proprietà o di una variabile come una lista associativa (di qualsiasi dimensione) ripristinerà sempre tutte le chiavi-valori ai loro valori predefiniti-quando-non-impostati. A meno che questo non sia il risultato voluto, è più sicuro aggiornare le chiavi-valori individualmente con una dichiarazione annidata.

Nota: Le dichiarazioni annidate non funzionano per le liste associative delle proprietà di contesto (come beamExceptions, keyAlterations, timeSignatureSettings, etc.). Queste proprietà possono essere modificate soltanto ridefinendole completamente come liste associative.

5.4 Proprietà e concetti utili

5.4.1 Modalità di inserimento

Il modo in cui la notazione contenuta in un file di input è interpretata è determinato dalla modalità di inserimento corrente. In generale, esistono due modi di specificare la modalità: una forma lunga, come `\chordmode`, e una breve, come `\chords`. La forma lunga viene usata tipicamente quando si passa l'input a una variabile o quando si inserisce l'input direttamente in un contesto creato esplicitamente. La forma breve crea implicitamente un contesto del tipo corretto per quell'input e passa l'input a questo direttamente. È utile in casi semplici in cui non è necessario creare esplicitamente il contesto.

Modalità accordo

Viene attivata col comando `\chordmode` e fa sì che l'input sia interpretato con la sintassi della notazione degli accordi, vedi Sezione 2.7 [Notazione per accordi], pagina 431. La musica in modalità accordo viene elaborata come accordi su un rigo se inserita in un contesto `Staff`, come nomi di accordo se inserita in un contesto `ChordNames` oppure come diagrammi della tastiera se inserita in un contesto `FretBoards`.

La modalità accordo viene attivata anche col comando `\chords`. Anche questo fa sì che l'input che segue sia interpretato con la sintassi della notazione degli accordi, ma in aggiunta crea implicitamente un nuovo contesto `ChordNames` e elabora l'input al suo interno come nomi di accordo, vedi [Stampa dei nomi degli accordi], pagina 437.

Modalità percussioni

Viene attivata col comando `\drummode` e fa sì che l'input sia interpretato con la sintassi della notazione delle percussioni, vedi [Notazione di base per percussioni], pagina 412. La musica in modalità percussioni è elaborata come note percussive se inserita in un contesto `DrumStaff`.

La modalità percussioni viene attivata anche col comando `\drums`. Anche questo fa sì che l'input che segue sia interpretato con la sintassi della notazione delle percussioni, ma in aggiunta

crea implicitamente un nuovo contesto `DrumStaff` e elabora l'input come note percussive, vedi [Notazione di base per percussioni], pagina 412.

Modalità basso continuo

Viene attivata col comando `\figuremode` e fa sì che l'input che segue sia interpretato con la sintassi del basso continuo, vedi [Inserimento del basso continuo], pagina 446. La musica in modalità basso continuo viene elaborata come basso continuo se inserita in un contesto `FiguredBass` o in un contesto `Staff`.

La modalità basso continuo viene attivata anche col comando `\figures`. Anche questo fa sì che l'input che segue sia interpretato con la sintassi del basso continuo, ma in aggiunta crea implicitamente un nuovo contesto `FiguredBass` e elabora l'input al suo interno come basso figurato, vedi [Introduzione al basso continuo], pagina 445.

Modalità tastiera e intavolatura

Non esistono speciali modalità di inserimento per i simboli della tastiera e dell'intavolatura (o tablatura).

Per creare diagrammi in intavolatura, inserire note o accordi nella modalità nota e visualizzarli in un contesto `TabStaff`, vedi [Intavolature predefinite], pagina 360.

Per creare diagrammi della tastiera sopra un rigo, inserire le note o gli accordi in modalità nota o in modalità accordo e elaborarli in un contesto `FretBoards`, vedi [Diagrammi dei tasti automatici], pagina 403. Altrimenti i diagrammi possono essere inseriti come testo (markup) sopra le note usando il comando `\fret-diagram`, vedi [Diagrammi dei tasti], pagina 382.

Modalità testo vocale

Viene attivata col comando `\lyricmode` e fa sì che l'input sia interpretato come sillabe del testo vocale con durate opzionali e modificatori del testo associato, vedi Sezione 2.1 [Musica vocale], pagina 280. L'input in modalità testo vocale viene elaborato come sillabe del testo se inserite in un contesto `Lyrics`.

La modalità testo vocale viene attivata anche col comando `\lyrics`. Anche questo fa sì che l'input che segue sia interpretato come sillabe del testo ma in aggiunta crea implicitamente un nuovo contesto `Lyrics` e elabora l'input al suo interno come sillabe.

La modalità testo vocale viene attivata anche col comando `\addlyrics`. Anche questo crea implicitamente un nuovo contesto `Lyrics` e aggiunge anche un comando implicito `\lyricsto` che associa il testo vocale che segue con la musica che lo precede, vedi [Durate automatiche delle sillabe], pagina 284..

Modalità testo (markup)

Viene attivata col comando `\markup` e fa sì che l'input sia interpretato con la sintassi di markup, vedi Sezione A.12 [Comandi per *markup*], pagina 713.

Modalità nota

Questa è la modalità predefinita e può essere attivata esplicitamente col comando `\notemode`. L'input viene interpretato come altezze, durate, markup, etc. e rappresentato come notazione musicale su un rigo.

Normalmente non è necessario specificare la modalità nota esplicitamente, ma potrebbe essere utile farlo in alcune situazioni, per esempio se si è in modalità testo vocale o accordo o qualsiasi altra modalità e si vuole inserire qualcosa che può essere fatto solo con la sintassi della modalità nota.

5.4.2 Direzione e posizionamento

Nella composizione tipografica musicale la direzione e il posizionamento di molti elementi è una questione di gusto. Per esempio, i gambi delle note possono essere rivolti in su o in giù; testi, dinamiche e altri segni espressivi possono essere posti sopra o sotto il rigo; il testo può essere allineato a sinistra, destra o al centro; etc. La maggior parte di queste scelte possono essere lasciate alla decisione di LilyPond, ma in alcuni casi si può voler forzare una direzione o un posizionamento particolari.

Indicatori di direzione delle articolazioni

Per impostazione predefinita alcune direzioni sono sempre in su o sempre in giù (es: le dinamiche o la corona), mentre altri elementi possono alternare tra su e giù in base alla direzione del gambo (come le legature di portamento o gli accenti).

L'azione predefinita può essere modificata antepoendo all'articolazione un *indicatore di direzione*. Sono disponibili tre indicatori di direzione: `^` (ovvero “su”), `_` (ovvero “giù”) e `-` (ovvero “usa la direzione predefinita”). L'indicatore di direzione solitamente può essere omesso, nel qual caso viene supposto `-`, ma un indicatore di direzione è **sempre** richiesto prima di

- comandi `\tweak`
- comandi `\markup`
- comandi `\tag`
- stringhe di testo, ovvero `-"stringa"`
- istruzioni di diteggiatura, come `-1`
- scorciatoie delle articolazioni, come `-. , -> , --`

Gli indicatori di direzione hanno effetto soltanto sulla nota vicina:

```
\relative {
  c' '2( c)
  c2_( c)
  c2( c)
  c2^( c)
}
```



La proprietà `direction`

La posizione della direzione di molti oggetti della formattazione è regolata dalla proprietà `direction`.

Il valore della proprietà `direction` può essere impostata su 1, ovvero “su” o “sopra”, oppure su -1, ovvero “giù” o “sotto”. I simboli UP e DOWN possono essere usati al posto di 1 e -1. La direzione predefinita può essere specificata impostando `direction` su 0 o CENTER. Altrimenti, in molti casi esistono comandi predefiniti per specificare la direzione. Questo hanno la forma

```
\xxxUp, \xxxDown o \xxxNeutral
```

dove `\xxxNeutral` significa “usa la direzione predefinita”. Vedi Sezione “Oggetti interni al rigo” in *Manuale di Apprendimento*.

In pochi casi, per esempio l'arpeggio, il valore della proprietà `direction` può specificare se l'oggetto debba essere posizionato a destra o a sinistra dell'oggetto genitore. In questo caso -1 o LEFT significano “a sinistra” e 1 o RIGHT significano “a destra”. 0 o CENTER significano “usa la direzione predefinita”.

Queste indicazioni hanno effetto su tutte le note finché non vengono annullate.

```
\relative {
  c' '2( c)
  \slurDown
  c2( c)
  c2( c)
  \slurNeutral
  c2( c)
}
```



Nella musica polifonica, generalmente è meglio specificare una voce esplicita invece di cambiare la direzione di un oggetto. Maggiori informazioni in Sezione 1.5.2 [Più voci], pagina 184.

Vedi anche

Manuale di apprendimento: Sezione “Oggetti interni al rigo” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 1.5.2 [Più voci], pagina 184.

5.4.3 Distanze e misurazioni

Le distanze in LilyPond sono di due tipi: assolute e proporzionali.

Le distanze assolute si usano per specificare i margini, le indentazioni e altri dettagli della formattazione di pagina; per impostazione predefinita sono specificate in millimetri. Le distanze possono essere specificate in altre unità di misura appendendo alla quantità `\mm`, `\cm`, `\in` (inch, ovvero pollici) o `\pt` (punti, 1/72.27 di un pollice). Le distanze della formattazione di pagina possono essere specificate anche in unità di misura proporzionali (vedi paragrafo seguente) appendendo `\staff-space` alla quantità. La formattazione di pagina è trattata dettagliatamente in Sezione 4.1 [Formattazione della pagina], pagina 552.

Le distanze proporzionali sono sempre specificate in unità di spazi rigo o, raramente, mezzo spazio rigo. Lo spazio rigo è la distanza tra due linee del rigo adiacenti. Il valore predefinito può essere modificato globalmente impostando la dimensione globale del rigo oppure può essere sovrascritto localmente cambiando la proprietà `staff-space` di `StaffSymbol`. Le distanze proporzionali vengono ridimensionate automaticamente insieme a qualsiasi cambiamento nella dimensione globale del rigo o nella proprietà `staff-space` di `StaffSymbol`, mentre i tipi di carattere vengono ridimensionati automaticamente soltanto con i cambiamenti alla dimensione globale del rigo. Questa permette quindi di variare facilmente la dimensione complessiva di una partitura. I metodi per impostare la dimensione globale del rigo sono descritti in Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

Se soltanto una sezione di una partitura deve essere elaborata su una scala diversa, per esempio una sezione ossia o una nota a piè di pagina, non si può cambiare la dimensione globale del rigo perché ciò avrebbe effetto sull'intera partitura. In questi casi bisogna modificare sia la proprietà `staff-space` di `StaffSymbol` sia la dimensione dei tipi di carattere. Esiste una funzione `Scheme`, `magstep`, che permette di convertire una modifica della dimensione del tipo di carattere nella modifica equivalente in `staff-space`. Questa funzione è spiegata e esemplificata in Sezione “Lunghezza e spessore degli oggetti” in *Manuale di Apprendimento*.

Vedi anche

Manuale di apprendimento: Sezione “Lunghezza e spessore degli oggetti” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 4.1 [Formattazione della pagina], pagina 552, Sezione 4.2.2 [Impostare la dimensione del rigo], pagina 565.

5.4.4 Dimensioni

Le dimensioni di un oggetto grafico specificano le posizioni delle estremità sinistra e destra e di quelle inferiore e superiore del riquadro in cui sono iscritti gli oggetti come distanze dal punto di riferimento degli oggetti in unità di spazi rigo. Queste posizioni sono solitamente scritte come due coppie Scheme. Per esempio, il comando di testo `\with-dimensions` prende tre argomenti: i primi due sono una coppia Scheme che indica le posizioni delle estremità sinistra e destra e un'altra coppia Scheme che indica le posizioni delle estremità inferiore e superiore:

```
\with-dimensions #'(-5 . 10) #'(-3 . 15) arg
```

Questo comando specifica un riquadro per *arg* il cui margine sinistro si trova a -5, il margine destro a 10, il margine inferiore a -3 e quello superiore a 15, tutti misurati a partire dal punto di riferimento degli oggetti in unità di spazi rigo.

Vedi anche

Guida alla notazione: Sezione 5.4.3 [Distanze e misurazioni], pagina 644.

5.4.5 Proprietà del simbolo del rigo

La posizione verticale e il numero delle linee del rigo possono essere definiti contemporaneamente. Come mostra l'esempio seguente, le posizioni delle note non sono influenzate dalle posizioni delle linee del rigo.

Nota: La proprietà `'line-positions` sovrascrive la proprietà `'line-count`. Il numero di linee del rigo è definito implicitamente dal numero di elementi nella lista di valori per `'line-positions`.

```
\new Staff \with {
  \override StaffSymbol.line-positions = #'(7 3 0 -4 -6 -7)
}
\relative { a4 e' f b | d1 }
```



La larghezza di un rigo può essere modificata. Le unità sono spazi rigo. La spaziatura degli oggetti dentro il rigo non è influenzata da questa impostazione.

```
\new Staff \with {
  \override StaffSymbol.width = #23
}
\relative { a4 e' f b | d1 }
```



5.4.6 Estensori

Molti oggetti della notazione musicale si estendono per varie note o addirittura per molte battute. Ne sono un esempio le legature di portamento, le travature, le parentesi dei gruppi irregolari,

quelle delle volte delle ripetizioni, i crescendo, i trilli e i glissandi. Tali oggetti in LilyPond si chiamano “spanner”, ovvero estensori, e hanno delle proprietà speciali per regolare il loro aspetto e comportamento. Alcune di queste proprietà sono comuni a tutti gli estensori; altre sono limitate a un sottoinsieme di estensori.

Tutti gli estensori supportano l'interfaccia `spanner-interface`. Ma pochi, essenzialmente quelli che disegnano una linea dritta tra due oggetti, supportano anche l'interfaccia `line-spanner-interface`.

Uso di `spanner-interface`

Questa interfaccia fornisce tre proprietà che si applicano a vari estensori.

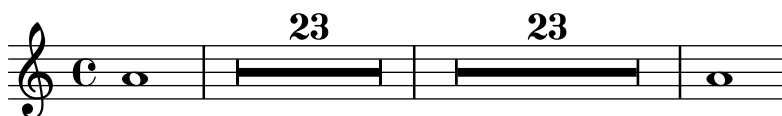
La proprietà `minimum-length`

La lunghezza minima dell'estensore è specificata dalla proprietà `minimum-length`. Aumentando questa si aumenta necessariamente anche la spaziatura delle note comprese tra le due estremità. Tuttavia questa modifica non ha effetto su molti estensori, perché la loro lunghezza è determinata da altre considerazioni. Di seguito vengono mostrati alcuni esempi in cui è in funzione.

```
a' ~ a'
a'
% aumenta la lunghezza della legatura di valore
-\tweak minimum-length #5
~ a'
```



```
\relative \compressMMRests {
  a'1
  R1*23
  % aumenta la lunghezza della misura con pausa
  \once \override MultiMeasureRest.minimum-length = #20
  R1*23
  a1
}
```



```
\relative {
  a' \< a a a \!
  % aumenta la lunghezza della forcilla
  \override Hairpin.minimum-length = #20
  a \< a a a \!
}
```



Questa modifica può essere usata anche per aumentare la lunghezza delle legature di portamento e di frase:

```
\relative {
```

```

a' ( g)
a
-\tweak minimum-length #5
( g)

a\ ( g\ )
a
-\tweak minimum-length #5
\ ( g\ )
}

```



Per alcuni oggetti della formattazione, la proprietà `minimum-length` diventa effettiva solo se viene richiamata esplicitamente la procedura `set-spacing-rods`. Per farlo, la proprietà `springs-and-rods` deve essere impostata su `ly:spanner::set-spacing-rods`. Per esempio, la lunghezza minima di un glissando cambia solo quando si imposta la proprietà `springs-and-rods`:

```

% default
e' \glissando c''

% non funziona da sola
\once \override Glissando.minimum-length = #20
e' \glissando c''

% funziona solo quando entrambe le modifiche sono presenti
\once \override Glissando.minimum-length = #20
\once \override Glissando.springs-and-rods = #ly:spanner::set-spacing-rods
e' \glissando c''

```



Lo stesso vale per l'oggetto `Beam`:

```

% non funziona da sola
\once \override Beam.minimum-length = #20
e'8 e' e' e'

% funziona solo quando entrambe le modifiche sono presenti
\once \override Beam.minimum-length = #20
\once \override Beam.springs-and-rods = #ly:spanner::set-spacing-rods
e'8 e' e' e'

```



La proprietà `minimum-length-after-break`

La proprietà `minimum-length-after-break` può essere usata per allungare stensori spezzati dopo un a capo. Come per la proprietà `minimum-length`, viene spesso usata per impostare la proprietà `springs-and-rods` su `ly:spanner::set-spacing-rods`.

```

{
  \once \override Tie.minimum-length-after-break = 20
  a1~
  \break
  a1

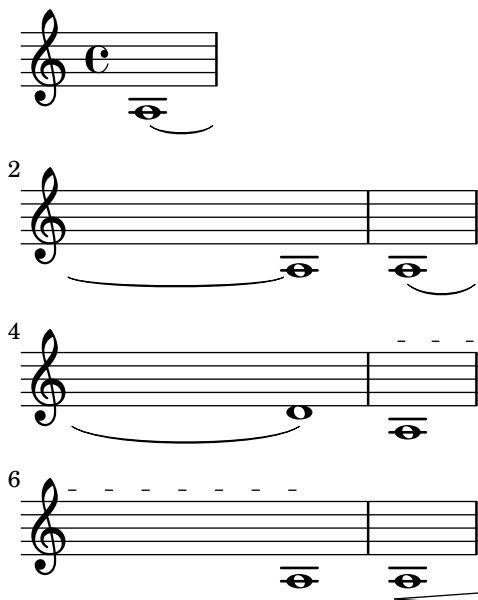
  \once \override Slur.minimum-length-after-break = 20
  a1(
  \break
  d'1)

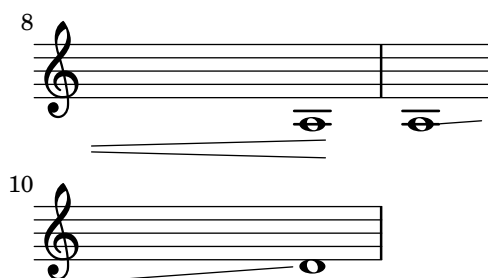
  \once \override TextSpanner.springs-and-rods =
    #ly:spanner::set-spacing-rods
  \once \override TextSpanner.minimum-length-after-break = 20
  a1\startTextSpan
  \break
  a1\stopTextSpan

  \once \override Hairpin.after-line-breaking = ##t
  \once \override Hairpin.to-barline = ##f
  \once \override Hairpin.minimum-length-after-break = 20
  a1\<
  \break
  a1\!

  \once \override Glissando.springs-and-rods =
    #ly:spanner::set-spacing-rods
  \once \override Glissando.breakable = ##t
  \once \override Glissando.after-line-breaking = ##t
  \once \override Glissando.minimum-length-after-break = 20
  a1\glissando
  \break
  d'1
}

```

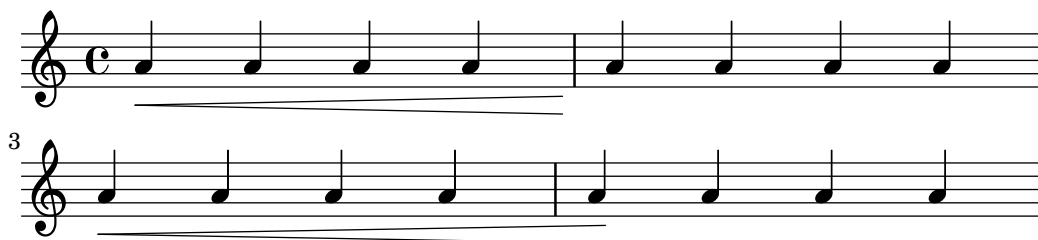




La proprietà to-barline

La terza proprietà utile di `spanner-interface` è `to-barline`. Il valore predefinito (vero) fa sì che le forcelle e gli altri estensori che terminano sulla prima nota di una misura si estendano invece fino alla stanghetta immediatamente precedente. Se impostata su falso, l'estensore andrà oltre la stanghetta e terminerà sulla nota stessa:

```
\relative {
  a' \< a a a a \! a a a \break
  \override Hairpin.to-barline = ##f
  a \< a a a a \! a a a
}
```



Questa proprietà non funziona con tutti gli estensori. Per esempio, impostandola su `#t` non si produce alcun effetto su legature di portamento o di frase, o su altri estensori per i quali terminare sulla stanghetta non avrebbe senso.

Uso di `line-spanner-interface`

Gli oggetti che supportano l'interfaccia `line-spanner-interface` comprendono:

- `DynamicTextSpanner`
- `Glissando`
- `TextSpanner`
- `TrillSpanner`
- `VoiceFollower`

La routine responsabile del disegno di questi estensori è `ly:line-spanner::print`. Essa determina la posizione esatta delle due estremità e disegna una linea tra di esse, nello stile richiesto. Le posizioni delle due estremità dell'estensore sono calcolate al volo, ma è possibile sovrascrivere le loro coordinate Y. Le proprietà da specificare sono annidate di due livelli nella gerarchia della proprietà, ma la sintassi del comando `\override` è piuttosto semplice:

```
e''2 \glissando b'
\once \override Glissando.bound-details.left.Y = #3
\once \override Glissando.bound-details.right.Y = #-2
e''2 \glissando b'
```



Le unità di misura della proprietà *Y* sono gli spazi rigo (*staff-space*), con la linea centrale del rigo che è il punto zero. Per il *glissando*, questo è il valore per *Y* alla coordinata *X* che corrisponde al punto centrale della testa di ogni nota, se si immagina che la linea si estenda fino a là.

Se *Y* non è impostato, il valore viene calcolato dalla posizione verticale del punto di attacco corrispondente dell'estensore.

In caso di un'interruzione di linea, i valori per le estremità sono specificati dalle sottoliste *left-broken* e *right-broken* di *bound-details*. Per esempio:

```
\override Glissando.breakable = ##t
\override Glissando.bound-details.right-broken.Y = #-3
c''1 \glissando \break
f''1
```



Altre proprietà delle sottoliste *left* e *right* della proprietà *bound-details* possono essere modificate nello stesso modo di *Y*:

Y Questa imposta la coordinata *Y* dell'estremità, che si sposta di un certo numero di *staff-space* dalla linea centrale del rigo. Per impostazione predefinita, è il centro dell'oggetto a cui è associato, dunque un *glissando* punta al centro verticale della testa di nota.

Nel caso di estensori orizzontali, come quelli del testo e del trillo, il suo valore è fisso su 0.

attach-dir

Questa determina dove la linea inizia e termina nella direzione *X*, relativa all'oggetto a cui è associato. Quindi un valore di -1 (o *LEFT*) fa sì che la linea inizi/termini sul lato sinistro della testa della nota a cui è attaccata.

X

Questa è la coordinata *X* assoluta dell'estremità. Viene solitamente calcolata al volo e sovrascriverla produce solitamente un effetto poco utile.

stencil

Gli estensori della linea possono avere dei simboli all'inizio o alla fine, contenuti in questa sottoproprietà. Questa proprietà è per uso interno; si raccomanda di usare *text* al suo posto.

text

Questo è un testo markup che viene valutato per produrre lo stampo (*stencil*, in inglese). Viene usato per mettere *cresc.*, *tr* e altro testo su estensori orizzontali.

```
\override TextSpanner.bound-details.left.text
= \markup { \small \bold Rall. }
\relative { c''2\startTextSpan b c a\stopTextSpan }
```



stencil-align-dir-y
stencil-offset

Se non si imposta una di queste proprietà, lo stampo viene messo semplicemente all'estremità, centrato sulla linea, come definito nelle sottoproprietà X e Y. Impostando `stencil-align-dir-y` o `stencil-offset` il simbolo si sposterà verticalmente sul margine rispetto all'estremità della linea:

```
\override TextSpanner.bound-details.left.stencil-align-dir-y = #-2
\override TextSpanner.bound-details.right.stencil-align-dir-y = #UP

\override TextSpanner.bound-details.left.text = "ggg"
\override TextSpanner.bound-details.right.text = "hhh"

\relative { c'4^\startTextSpan c c c \stopTextSpan }
```



Nota bene che valori negativi spostano il testo in *su*, contrariamente a quanto si potrebbe pensare, perché un valore di `-1` o `DOWN` fa sì che si allinei il margine *inferiore* del testo con la linea dell'estensore. Un valore di `1` o `UP` allinea il margine superiore del testo con la linea dell'estensore.

- arrow** Impostando questa sottoproprietà su `#t` viene generata la punta di una freccia a un'estremità della linea.
- padding** Questa sottoproprietà regola lo spazio tra l'estremità della linea specificata e la fine reale. Senza padding, un glissando inizia e termina nel centro della testa di ogni nota.

La funzione musicale `\endSpanners` termina prematuramente l'estensore che inizia nella nota immediatamente seguente. Viene terminato esattamente dopo una nota o alla stanghetta seguente se `to-barline` è impostato su `vero` e c'è una stanghetta prima della nota successiva.

```
\relative c'' {
  \endSpanners
  c2 \startTextSpan c2 c2
  \endSpanners
  c2 \< c2 c2
}
```



Quando si usa `\endSpanners` non è necessario chiudere `\startTextSpan` con `\stopTextSpan`, né è necessario chiudere le forcelle con `\!`.

Vedi anche

Guida al funzionamento interno: Sezione “TextSpanner” in *Guida al Funzionamento Interno*, Sezione “Glissando” in *Guida al Funzionamento Interno*, Sezione “VoiceFollower” in *Guida al Funzionamento Interno*, Sezione “TrillSpanner” in *Guida al Funzionamento Interno*, Sezione “line-spanner-interface” in *Guida al Funzionamento Interno*.

5.4.7 Visibilità degli oggetti

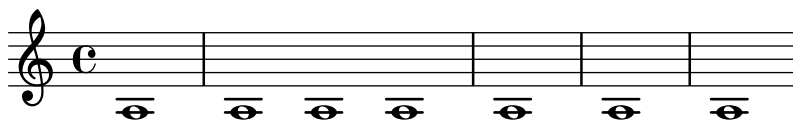
Esistono quattro modi principali per regolare la visibilità degli oggetti della formattazione: si può togliere il loro stampo, possono essere resi trasparenti, possono essere colorati di bianco o si può sovrascrivere la loro proprietà `break-visibility`. Le prime tre sono valide per tutti gli oggetti della formattazione; l'ultimo solo per alcuni, gli oggetti che possono essere *interrotti*, o spezzati, (`break`, in inglese). Il Manuale di apprendimento contiene un'introduzione a queste quattro tecniche in Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*.

Esistono anche altre tecniche che sono specifiche di alcuni oggetti della formattazione. Queste sono trattate in [Considerazioni speciali], pagina 655.

Soppressione dello stampo

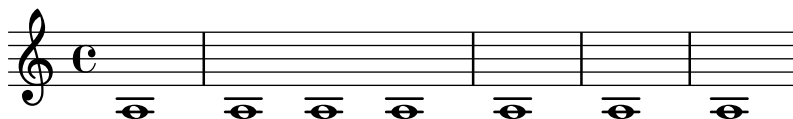
Tutti gli oggetti della formattazione hanno una proprietà `stencil` (in italiano, stampo). Per impostazione predefinita questa è impostata sulla funzione specifica che disegna quell'oggetto. Se questa proprietà viene sovrascritta e impostata su `#f`, non verrà richiamata alcuna funzione e l'oggetto non verrà disegnato. L'azione predefinita può essere ripristinata con `\revert`.

```
a1 a
\override Score.BarLine.stencil = #f
a a
\revert Score.BarLine.stencil
a a a
```



Questa operazione piuttosto comune ha una scorciatoia, `\omit`:

```
a1 a
\omit Score.BarLine
a a
\undo \omit Score.BarLine
a a a
```



Rendere gli oggetti trasparenti

Tutti gli oggetti della formattazione hanno una proprietà `transparent` la cui impostazione predefinita è `#f` (falso). Se impostata su `#t` (vero), l'oggetto occupa sempre lo spazio ma è reso invisibile.

```
a'4 a'
\once \override NoteHead.transparent = #t
a' a'
```



Questa operazione piuttosto comune ha una scorciatoia, `\hide`:

```
a'4 a'
\once \hide NoteHead
```

a' a'



Dipingere gli oggetti di bianco

Tutti gli oggetti della formattazione hanno una proprietà del colore il cui valore predefinito è `black` (nero). Se questa viene sovrascritta con `white` (bianco) l'oggetto non sarà più distinguibile dallo sfondo. Tuttavia se l'oggetto attraversa altri oggetti, il colore dei punti di contatto saranno determinati dall'ordine in cui sono disegnati, e ciò potrebbe lasciare un'immagine "spettrale" dell'oggetto bianco, come mostrato qui:

```
\override Staff.Clef.color = #white
a'1
```



Ciò può essere evitato cambiando l'ordine di stampa degli oggetti. Tutti gli oggetti della formattazione hanno una proprietà `layer` (strato) che deve essere impostata su un numero intero. Gli oggetti col valore più basso di `layer` sono disegnati per primi, poi vengono disegnati quelli con valori via via più alti, dunque gli oggetti con valori più alti stanno sopra gli oggetti con valori più bassi. Per impostazione predefinita alla maggior parte degli oggetti viene assegnato un valore di `layer` pari a 1, sebbene ad alcuni oggetti, tra cui `StaffSymbol` e `BarLine`, sia assegnato un valore di 0. L'ordine con cui vengono stampati oggetti con lo stesso valore di `layer` è indeterminato.

Nell'esempio precedente la chiave bianca, con un valore predefinito di `layer` di 1, viene disegnato dopo le linee del rigo (il cui valore predefinito di `layer` è 0), dunque sta sopra di esse. Per cambiare tale comportamento, all'oggetto `Clef` deve essere assegnato un valore di `layer` inferiore, per esempio -1, in modo che venga disegnato prima:

```
\override Staff.Clef.color = #white
\override Staff.Clef.layer = #-1
a'1
```



Frammenti di codice selezionati

Uso della proprietà *whiteout*

Qualsiasi oggetto grafico può essere posizionato sopra uno sfondo bianco per mascherare parti degli oggetti che si trovano sotto. Ciò può essere utile per migliorare l'aspetto delle collisioni in situazioni complesse in cui il riposizionamento degli oggetti è troppo difficile. Bisogna impostare esplicitamente la proprietà `layer` (livello) per controllare quali oggetti debbano essere mascherati dallo sfondo bianco.

In questo esempio la collisione della legatura di valore con l'indicazione di tempo viene migliorata mascherando la parte della legatura che incrocia l'indicazione di tempo impostando la proprietà `whiteout` di `TimeSignature`. Per farlo si sposta `TimeSignature` su un livello superiore a `Tie`, che viene lasciato al livello predefinito 1; e `StaffSymbol` viene spostato su un livello superiore a `TimeSignature` in modo che non venga mascherato.

```
{
```

```

\override Score.StaffSymbol.layer = 4
\override Staff.TimeSignature.layer = 3
b'2 b'~
\once \override Staff.TimeSignature.whiteout = ##t
\time 3/4
b' r4
}

```



Uso di break-visibility

La maggior parte degli oggetti della formattazione sono stampati una volta sola, ma alcuni come le stanghette, le chiavi, le indicazioni di tempo e le armature di chiave possono dover essere stampate due volte quando si verifica un'interruzione di linea – una volta al termine della linea e di nuovo all'inizio della linea successiva. Tali oggetti, chiamati *spezzabili*, (dall'inglese breakable) hanno una proprietà, *break-visibility*, per regolare la loro visibilità nelle tre posizioni in cui potrebbero apparire: all'inizio di una linea, in mezzo a una linea se vengono modificati e al termine di una linea se un cambiamento ha luogo lì.

Per esempio, l'indicazione di tempo viene stampata all'inizio della prima linea, ma da nessuna altra parte a meno che non cambi e allora verrà stampata nel punto in cui il cambiamento ha luogo. Se tale cambiamento si verifica al termine di una linea, la nuova indicazione di tempo verrà stampata all'inizio della linea successiva e un'indicazione di tempo di precauzione verrà stampata anche al termine della linea precedente.

Tale comportamento è regolato dalla proprietà *break-visibility*, che è spiegata in Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*. Questa proprietà prende un vettore di tre valori booleani che, in ordine, determinano se l'oggetto è stampato al termine, in mezzo o all'inizio di una linea. O, per essere più precisi, prima di un'interruzione di linea, dove non c'è un'interruzione di linea oppure dopo un'interruzione di linea.

Altrimenti, queste otto combinazioni possono essere specificate da funzioni predefinite, definite in `scm/output-lib.scm`, dove le ultime tre colonne indicano se gli oggetti della formattazione saranno visibili nelle posizioni mostrate nell'intestazione delle colonne:

Funzione	Forma vettoriale	Prima di interruzione	Senza interruzione	Dopo interruzione
<code>all-visible</code>	<code>##t ##t ##t</code>	sì	sì	sì
<code>begin-of-line-visible</code>	<code>##f ##f ##t</code>	no	no	sì
<code>center-visible</code>	<code>##f ##t ##f</code>	no	sì	no
<code>end-of-line-visible</code>	<code>##t ##f ##f</code>	sì	no	no
<code>begin-of-line-invisible</code>	<code>##t ##t ##f</code>	sì	sì	no
<code>center-invisible</code>	<code>##t ##f ##t</code>	sì	no	sì
<code>end-of-line-invisible</code>	<code>##f ##t ##t</code>	no	sì	sì
<code>all-invisible</code>	<code>##f ##f ##f</code>	no	no	no

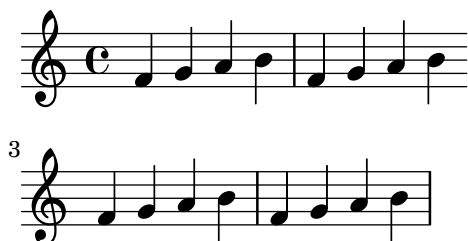
Le impostazioni predefinite di *break-visibility* dipendono dall'oggetto della formattazione. La tabella seguente mostra tutti gli oggetti rilevanti che sono influenzati da *break-visibility* e l'impostazione predefinita di questa proprietà:

Oggetto formattazione	Contesto abituale	Impostazione predefinita
<code>BarLine</code>	<code>Score</code>	calcolato

BarNumber	Score	begin-of-line-visible
BreathingSign	Voice	begin-of-line-invisible
Clef	Staff	begin-of-line-visible
Custos	Staff	end-of-line-visible
DoublePercentRepeat	Voice	begin-of-line-invisible
KeyCancellation	Staff	begin-of-line-invisible
KeySignature	Staff	begin-of-line-visible
ClefModifier	Staff	begin-of-line-visible
RehearsalMark	Score	end-of-line-invisible
TimeSignature	Staff	all-visible

L'esempio successivo mostra l'uso della forma vettoriale per regolare la visibilità delle stanghette:

```
\relative {
  f'4 g a b
  f4 g a b
  % Toglie la stanghetta al termine della linea corrente
  \once \override Score.BarLine.break-visibility = ##(f #t #t)
  \break
  f4 g a b
  f4 g a b
}
```



Sebbene tutti i componenti del vettore usati per sovrascrivere `break-visibility` debbano essere presenti, non tutti sono funzionanti con qualsiasi oggetto della formattazione, e alcune combinazioni potrebbero perfino dare errore. Esistono le seguenti limitazioni:

- Non è possibile stampare le stanghette all'inizio di una linea.
- Un numero di battuta non può essere stampato all'inizio della *prima* linea a meno che non sia impostato per essere diverso da 1.
- Clef – vedi la sezione successiva.
- Le ripetizioni percentuali doppie sono o *tutte stampate* o *tutte soppresse*. Usare `begin-of-line-invisible` per stampare e `all-invisible` per sopprimere.
- Key signature – vedi la sezione successiva.
- ClefModifier – vedi la sezione successiva.

Considerazioni speciali

Visibilità dopo un cambio esplicito

La proprietà `break-visibility` controlla la visibilità delle armature di chiave e dei cambi di chiave solo all'inizio delle linee, ovvero dopo un'interruzione di linea. Non ha effetto sulla visibilità dell'armatura di chiave o della chiave che seguono un cambio esplicito dell'armatura o della chiave all'interno o alla fine di una linea. Nell'esempio seguente l'armatura di chiave che

segue il cambio esplicito a Si bemolle maggiore è ancora visibile, anche se all-invisible è impostata.

```
\relative {
  \key g \major
  f'4 g a b
  % Tentativo di eliminazione di tutte le armature di chiave
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b
  \break
  f4 g a b
  f4 g a b
}
```



La visibilità di questi cambi espliciti di armatura e di chiave è controllata dalle proprietà `explicitKeySignatureVisibility` e `explicitClefVisibility`. Sono equivalenti alla proprietà `break-visibility` e prendono entrambe come argomento un vettore di tre booleani o le funzioni predefinite elencate prima, proprio come `break-visibility`. Entrambe le proprietà appartengono al contesto `Staff`, non agli oggetti della formattazione stessi, dunque si impostano col comando `\set`. La loro impostazione predefinita è `all-visible`. Queste proprietà regolano solo la visibilità di armature di chiave e chiavi risultanti da cambi espliciti e non quelli all’inizio delle linee; per eliminare questi ultimi bisogna sempre sovrascrivere `break-visibility` nell’oggetto appropriato.

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Visibilità dei bequadri

Per eliminare i bequadri stampati su un cambio di chiave esplicito, impostare la proprietà del contesto Staff `printKeyCancellation` su `#f`:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = ##f
  \override Staff.KeySignature.break-visibility = #all-invisible
  \key bes \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Con queste modifiche restano solo le alterazioni accanto alle note per indicare il cambio di armatura.

Nota bene che quando si cambia l'armatura su Do maggiore o La minore i bequadri sono l'unica indicazione del cambio di chiave. In questo caso impostando `printKeyCancellation` su `#f` non si ottiene alcun effetto:

```
\relative {
  \key g \major
  f'4 g a b
  \set Staff.explicitKeySignatureVisibility = #all-invisible
  \set Staff.printKeyCancellation = ##f
  \key c \major
  f4 g a b \break
  f4 g a b
  f4 g a b
}
```



Per sopprimere i bequadri anche quando la tonalità passa a Do maggiore o La minore, sovrascrivere la visibilità del grob `KeyCancellation`:

```
\relative {
  \key g \major
  f'4 g a b
```

```

\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Staff.KeyCancellation.break-visibility = #all-invisible
\key c \major
f4 g a b \break
f4 g a b
f4 g a b
}

```



Battute automatiche

In un caso particolare, la stampa delle stanghette può essere disattivata impostando la proprietà `automaticBars` nel contesto `Score`. Se impostata su `#f`, le stanghette non verranno stampate automaticamente; devono invece essere create esplicitamente con un comando `\bar`. Diversamente dal comando predefinito `\cadenzaOn`, le misure vengono comunque contate. La generazione delle battute riprenderà in base a quel conteggio se questa proprietà viene poi impostata su `#t`. Se impostata su `#f`, le interruzioni di linea possono trovarsi solo su comandi `\bar` espliciti.

Chiavi trasposte

Il piccolo simbolo di trasposizione sulle chiavi trasposte è prodotto dall'oggetto di formattazione `ClefModifier`. La sua visibilità è ereditata automaticamente dall'oggetto `Clef`, dunque non è necessario applicare una modifica di `break-visibility` agli oggetti `ClefModifier` per sopprimere i simboli di trasposizione per le chiavi invisibili.

Per cambi di chiave espliciti, la proprietà `explicitClefVisibility` regola sia il simbolo della chiave che qualsiasi simbolo di trasposizione ad esso associato.

Vedi anche

Manuale di apprendimento: Sezione “Visibilità e colore degli oggetti” in *Manuale di Apprendimento*.

5.4.8 Stili della linea

Alcune indicazioni esecutive, come *rallentando*, *accelerando* e *trilli* sono scritte in forma testuale e sono estese per varie misure tramite delle linee, talvolta puntate o ondulate.

Queste usano tutte le stesse routine del glissando per disegnare i testi e le linee, dunque il loro comportamento viene modificato nello stesso modo. Viene fatto con un estensore, e la routine responsabile del disegno degli estensori è `ly:line-spanner::print`. Questa routine determina la posizione esatta dei due *punti di estensione* e disegna una linea tra loro, nello stile richiesto.

Ecco un esempio che mostra i vari stili di linea disponibili, e come impostarli..

```

\relative {
  d''2 \glissando d'2
  \once \override Glissando.style = #'dashed-line
  d,2 \glissando d'2
  \override Glissando.style = #'dotted-line
  d,2 \glissando d'2
  \override Glissando.style = #'zigzag
}

```

```
d,2 \glissando d'2
\override Glissando.style = #'trill
d,2 \glissando d'2
}
```



I punti delle estremità dell'estensore sono calcolati al volo per ogni oggetto grafico, ma è possibile sovrascriverli:

```
\relative {
  e''2 \glissando f
  \once \override Glissando.bound-details.right.Y = #-2
  e2 \glissando f
}
```



Il valore di Y è impostato su -2 per l'estremità destra. Quella sinistra può essere aggiustata ugualmente specificando left invece di right.

Se Y non è impostata, il valore è calcolato a partire dalla posizione verticale dei punti di attacco sinistro e destro dell'estensore.

Sono possibili altre modifiche degli estensori; maggiori dettagli in Sezione 5.4.6 [Estensori], pagina 645.

5.4.9 Rotazione degli oggetti

Sia gli oggetti della formattazione che il testo compreso in un blocco markup possono essere ruotati di qualsiasi angolo in quasi qualsiasi punto, ma il metodo per farlo cambia.

Rotazione degli oggetti della formattazione

Tutti gli oggetti della formattazione che supportano l'interfaccia grob-interface possono essere ruotati impostando la loro proprietà rotation. Questa prende come argomento una lista di tre elementi: l'angolo di rotazione in senso antiorario, e le coordinate x e y del punto relativo al punto di riferimento dell'oggetto intorno al quale si deve eseguire la rotazione. L'angolo di rotazione è specificato in gradi e le coordinate in spazi rigo.

L'angolo di rotazione e le coordinate del punto di rotazione devono essere determinate per prova e errore.

Esistono solo poche situazioni in cui la rotazione degli oggetti della formattazione è utile; l'esempio seguente ne mostra uno:

```
g4\< e' d'' f''\!
\override Hairpin.rotation = #'(15 -1 0)
g4\< e' d'' f''\!
```



Rotazione del testo

Tutto il testo in un blocco `\markup` può essere ruotato di qualsiasi angolo facendolo precedere dal comando `\rotate`. Il comando prende due argomenti: l'angolo di rotazione in gradi in senso antiorario e il testo da ruotare. Le estensioni del testo non sono ruotate: prendono il loro valore dagli estremi delle coordinate *x* e *y* del testo ruotato. Nell'esempio seguente la proprietà `outside-staff-priority` del testo è impostata su `#f` per disabilitare l'elusione automatica delle collisioni, che spingerebbe parte del testo troppo in alto.

```
\override TextScript.outside-staff-priority = ##f
g4~\markup { \rotate #30 "un Sol" }
b~\markup { \rotate #30 "un Si" }
des'~\markup { \rotate #30 "un Re bemolle" }
fis'~\markup { \rotate #30 "un Fa diesis" }
```



5.5 Ritocchi avanzati

Questa sezione discute i vari approcci che permettono di migliorare l'aspetto delle partiture.

Vedi anche

Manuale di apprendimento: Sezione “Modifica dell’output” in *Manuale di Apprendimento*, Sezione “Altre fonti di informazione” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione 5.2 [Come funziona la Guida al funzionamento interno], pagina 625, Sezione 5.3 [Modifica delle proprietà], pagina 627.

Estendere LilyPond: Sezione “Interfaces for programmers” in *Estendere*.

File installati: `scm/define-grobs.scm`.

Frammenti: Sezione “Tweaks and overrides” in *Frammenti di codice*.

Guida al funzionamento interno: Sezione “All layout objects” in *Guida al Funzionamento Interno*.

5.5.1 Allineamento degli oggetti

Gli oggetti grafici che supportano l’interfaccia `self-alignment-interface` e/o l’interfaccia `side-position-interface` possono essere allineati a un oggetto disposto precedentemente in vari modi. L’elenco di questi oggetti è consultabile in Sezione “`self-alignment-interface`” in *Guida al Funzionamento Interno* e Sezione “`side-position-interface`” in *Guida al Funzionamento Interno*.

Tutti gli oggetti grafici hanno un punto di riferimento, un’estensione orizzontale e una verticale. L’estensione orizzontale è una coppia di numeri che determina gli spostamenti dal punto di riferimento dei margini sinistro e destro (gli spostamenti a sinistra sono negativi). L’estensione verticale è una coppia di numeri che determina lo spostamento dal punto di riferimento ai margini inferiore e superiore (gli spostamenti in giù sono negativi).

La posizione di un oggetto su un rigo è dato dai valori delle proprietà `X-offset` e `Y-offset`. Il valore di `X-offset` dà lo spostamento dalla coordinata *X* del punto di riferimento dell’oggetto genitore, mentre il valore di `Y-offset` dà lo spostamento dalla linea centrale del rigo. I valori di

X-offset e Y-offset possono essere impostati direttamente oppure possono essere impostati per essere calcolati dalle procedure in modo da consentire un allineamento con l'oggetto genitore.

Nota: Molti oggetti hanno speciali questioni di posizionamento che fanno sì che qualsiasi impostazione di X-offset o Y-offset sia ignorata o modificata, anche se l'oggetto supporta l'interfaccia self-alignment-interface. Sovrascrivendo le proprietà X-offset o Y-offset su un valore fisso la rispettiva proprietà self-alignment verrà ignorata.

Per esempio, un'alterazione può essere riposizionata verticalmente impostando Y-offset ma qualsiasi modifica di X-offset non avrà effetto.

I segni di chiamata possono essere allineati con oggetti spezzabili come stanghette, simboli di chiave, simboli di indicazione di tempo e armature di chiave. Ci sono proprietà speciali elencate nell'interfaccia break-aligned-interface per posizionare i segni di chiamata su tali oggetti.

Vedi anche

Guida alla notazione: [Uso di break-alignable-interface], pagina 663.

Estendere LilyPond: Sezione "Callback functions" in *Estendere*.

Impostazione diretta di X-offset e Y-offset

Si possono assegnare valori numerici alle proprietà X-offset e Y-offset di molti oggetti. L'esempio seguente mostra tre note con la posizione predefinita della diteggiatura e le posizioni con X-offset e Y-offset modificate.

```
a'-3
a'
-\tweak X-offset #0
-\tweak Y-offset #0
-3
a'
-\tweak X-offset #-1
-\tweak Y-offset #1
-3
```



Uso di side-position-interface

Un oggetto che supporta l'interfaccia side-position-interface può essere posizionato accanto al suo oggetto genitore in modo che i margini indicati dei due oggetti si tocchino. L'oggetto può essere posizionato sopra, sotto, a destra o a sinistra dell'oggetto genitore. Quest'ultimo non può essere specificato; è invece determinato dall'ordine degli elementi nel flusso di input. La maggior parte degli oggetti ha la testa di nota associata come oggetto genitore.

I valori delle proprietà side-axis e direction determinano dove l'oggetto deve essere posizionato, in questo modo:

Proprietà side-axis	Proprietà direction	Posizionamento
0	-1	sinistra
0	1	destra

1	-1	sotto
1	1	sopra

Quando `side-axis` è 0, `X-offset` deve essere impostato sulla procedura `ly:side-position-interface::x-aligned-side`. Questa procedura restituirà il valore corretto di `X-offset` per porre l'oggetto al lato sinistro o destro dell'oggetto genitore in base al valore di `direction`.

Quando `side-axis` è 1, `Y-offset` deve essere impostato sulla procedura `ly:side-position-interface::y-aligned-side`. Questa procedura restituirà il valore corretto di `Y-offset` per porre l'oggetto sopra o sotto l'oggetto genitore in base al valore di `direction`.

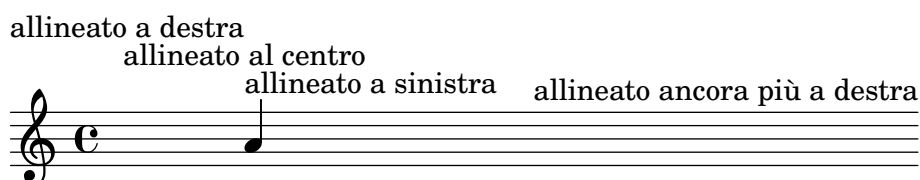
Uso di `self-alignment-interface`

Autoallineamento orizzontale degli oggetti

L'allineamento orizzontale di un oggetto che supporta l'interfaccia `self-alignment-interface` è regolato dal valore della proprietà `self-alignment-X`, purché la proprietà `X-offset` dell'oggetto sia impostata su `ly:self-alignment-interface::x-aligned-on-self`. A `self-alignment-X` può essere assegnato qualsiasi numero reale, in unità di metà del totale dell'estensione `X` dell'oggetto. Valori negativi spostano l'oggetto a destra, valori positivi lo spostano a sinistra. Un valore di 0 centra l'oggetto sul punto di riferimento del suo oggetto genitore, un valore di -1 allinea il margine sinistro dell'oggetto al punto di riferimento del suo genitore, e un valore di 1 allinea il margine destro dell'oggetto al punto di riferimento del suo genitore. I simboli `LEFT`, `CENTER` e `RIGHT` possono essere usati al posto dei valori -1, 0 e 1, rispettivamente.

Normalmente si usa il comando `\override` per modificare il valore di `self-alignment-X`, ma il comando `\tweak` può essere utile per allineare separatamente varie annotazioni su una singola nota:

```
a'
-\tweak self-alignment-X #-1
^"allineato a sinistra"
-\tweak self-alignment-X #0
^"allineato al centro"
-\tweak self-alignment-X #RIGHT
^"allineato a destra"
-\tweak self-alignment-X #-2.5
^"allineato ancora più a destra"
```



Autoallineamento verticale degli oggetti

Gli oggetti possono essere allineati verticalmente in un modo analogo al loro allineamento orizzontale se la proprietà `Y-offset` è impostata su `ly:self-alignment-interface::y-aligned-on-self`. Tuttavia altri meccanismi sono spesso implicati nell'allineamento verticale: il valore di `Y-offset` è solo una delle variabili da prendere in considerazione. Ciò potrebbe rendere complicato l'aggiustamento del valore di alcuni oggetti. Le unità sono soltanto la metà dell'estensione verticale dell'oggetto, che è di solito piuttosto piccola, dunque potrebbero essere necessari grossi numeri. Un valore di -1 allinea il margine inferiore dell'oggetto al punto di riferimento dell'oggetto genitore, un valore

di 0 allinea il centro dell'oggetto al punto di riferimento del genitore, e un valore di 1 allinea il margine superiore dell'oggetto al punto di riferimento del genitore. I simboli DOWN, CENTER e UP possono essere usati al posto di -1, 0 e 1, rispettivamente.

Autoallineamento degli oggetti in entrambe le direzioni

Impostando sia X-offset che Y-offset, un oggetto può essere allineato in entrambe le direzioni simultaneamente.

L'esempio seguente mostra come aggiustare un segno di diteggiatura in modo che stia vicino alla testa di nota.

```
a'
-\tweak self-alignment-X #0.5 % sposta orizzontalmente a sinistra
-\tweak Y-offset #ly:self-alignment-interface:y-aligned-on-self
-\tweak self-alignment-Y #-1 % sposta verticalmente in su
-3 % terzo dito
```

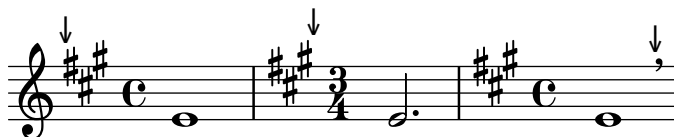


Uso di break-alignable-interface

I segni di chiamata e i numeri di battuta possono essere allineati con oggetti della notazione diversi dalle stanghette. Questi oggetti comprendono ambitus, breathing-sign, clef, custos, staff-bar, left-edge, key-cancellation, key-signature e time-signature.

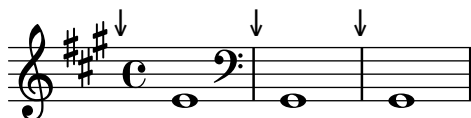
Ogni tipo di oggetto ha il suo punto di riferimento predefinito, al quale i segni di chiamata sono allineati:

```
% Il segno di chiamata sarà allineato al margine destro della chiave
\override Score.RehearsalMark.break-align-symbols = #'(clef)
\key a \major
\clef treble
\mark "↓"
e'1
% Il segno di chiamata sarà allineato al margine sinistro dell'indicazione di tempo
\override Score.RehearsalMark.break-align-symbols = #'(time-signature)
\key a \major
\clef treble
\time 3/4
\mark "↓"
e'2.
% Il segno di chiamata sarà centrato sul segno di respiro
\override Score.RehearsalMark.break-align-symbols = #'(breathing-sign)
\key a \major
\clef treble
\time 4/4
e'1
\breathe
\mark "↓"
```



Si può specificare un elenco dei possibili oggetti a cui allineare. Se alcuni oggetti sono invisibili in quel punto a causa dell'impostazione di `break-visibility` o di impostazioni esplicite della visibilità di tonalità e chiavi, il segno di chiamata o il numero di battuta viene allineato al primo oggetto dell'elenco che è visibile. Se nessun oggetto dell'elenco è visibile, l'oggetto è allineato alla stanghetta. Se la stanghetta è invisibile, l'oggetto viene allineato al punto in cui si dovrebbe trovare la stanghetta.

```
% Il segno di chiamata sarà allineato al margine destro dell'armatura di chiave
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
e'1
% Il segno di chiamata sarà allineato al margine destro della chiave
\set Staff.explicitKeySignatureVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef bass
\mark "↓"
gis,1
% Il segno di chiamata sarà centrato sopra la stanghetta
\set Staff.explicitKeySignatureVisibility = #all-invisible
\set Staff.explicitClefVisibility = #all-invisible
\override Score.RehearsalMark.break-align-symbols = #'(key-signature clef)
\key a \major
\clef treble
\mark "↓"
e'1
```



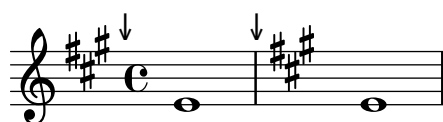
L'allineamento del segno di chiamata relativo all'oggetto della notazione può essere modificato, come evidenziato nell'esempio seguente. In una partitura con molteplici righe, questa impostazione deve essere fatta per tutti i righi.

```
% Il segno di chiamata sarà allineato al margine destro dell'armatura di chiave
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\key a \major
\clef treble
\time 4/4
\mark "↓"
e'1
% Il segno di chiamata sarà centrato sopra l'armatura di chiave
\once \override Score.KeySignature.break-align-anchor-alignment = #CENTER
\mark "↓"
\key a \major
e'1
% Il segno di chiamata sarà allineato al margine sinistro dell'armatura di chiave
\once \override Score.KeySignature.break-align-anchor-alignment = #LEFT
\key a \major
\mark "↓"
e'1
```



Il segno di chiamata può anche essere spostato a destra o a sinistra del margine sinistro di una quantità arbitraria. Le unità sono gli spazi rigo:

```
% Il segno di chiamata sarà allineato al margine sinistro dell'armatura di chiave
% e poi spostato a destra di 3.5 spazi rigo
\override Score.RehearsalMark.break-align-symbols = #'(key-signature)
\once \override Score.KeySignature.break-align-anchor = #3.5
\key a \major
\mark "↓"
e'1
% Il segno di chiamata sarà allineato al margine sinistro dell'armatura di chiave
% e poi spostato a sinistra di 2 spazi rigo
\once \override Score.KeySignature.break-align-anchor = #-2
\key a \major
\mark "↓"
e'1
```



5.5.2 Raggruppamento verticale dei grob

I grob `VerticalAlignment` e `VerticalAxisGroup` lavorano insieme. `VerticalAxisGroup` raggruppa insieme diversi grob come `Staff`, `Lyrics`, etc. `VerticalAlignment` quindi allinea verticalmente i diversi grob raggruppati insieme da `VerticalAxisGroup`. Di solito è presente un solo `VerticalAlignment` per partitura ma ogni `Staff`, `Lyrics`, etc. ha il suo `VerticalAxisGroup`.

5.5.3 Modifica degli stampi

Tutti gli oggetti della formattazione hanno una proprietà `stencil` che fa parte dell'interfaccia `grob-interface`. Per impostazione predefinita, questa proprietà è solitamente impostata su una funzione specifica per l'oggetto per cui è concepita in modo da creare il simbolo che lo rappresenta nell'output. Per esempio, l'impostazione standard della proprietà `stencil` dell'oggetto `MultiMeasureRest` è `ly:multi-measure-rest::print`.

Il simbolo standard di qualsiasi oggetto può essere sostituito modificando la proprietà `stencil` perché faccia riferimento a una procedura diversa e scritta appositamente. Ciò richiede una profonda conoscenza del funzionamento interno di LilyPond, ma esiste un modo più semplice che spesso è in grado di produrre risultati accettabili.

Si tratta di impostare la proprietà `stencil` sulla procedura che stampa testo – `ly:text-interface::print` – e aggiungere una proprietà `text` all'oggetto impostato per contenere il testo markup che produce il simbolo richiesto. Grazie alla flessibilità di markup, si può ottenere molto – vedi in particolare [Notazione grafica nel blocco markup], pagina 268.

L'esempio seguente illustra questa procedura cambiando il simbolo della testa di nota con una croce iscritta in un cerchio.

```
Xin0 = {
  \once \override NoteHead.stencil = #ly:text-interface::print
  \once \override NoteHead.text = \markup {
    \combine
```

```

\halign #-0.7 \draw-circle #0.85 #0.2 ##f
\musicglyph "noteheads.s2cross"
}
}
\relative {
a' a \Xin0 a a
}

```



Qualsiasi glifo *Feta* usato nel font Emmentaler può essere passato al comando markup `\musicglyph` – vedi Sezione A.8 [Il font Emmentaler], pagina 693.

I file EPS e i comandi Postscript possono essere inseriti entrambi nel corpo del testo con i comandi markup `\epsfile` e `\postscript` – vedi Sezione A.12.3 [Graphical markup], pagina 744.

Vedi anche

Guida alla notazione: [Notazione grafica nel blocco markup], pagina 268, Sezione 1.8.2 [Formattazione del testo], pagina 260, Sezione A.12 [Comandi per *markup*], pagina 713, Sezione A.8 [Il font Emmentaler], pagina 693, Sezione A.12.3 [Graphical markup], pagina 744.

5.5.4 Modifica delle forme

Modifica di legature di valore e di portamento

Le legature, siano esse di valore (Tie), di portamento (Slur), di frase (PhrasingSlur), di *laissez-vibrer* (LaissezVibrerTie) o di valore ripetute (RepeatTie), sono tutte disegnate come curve Bézier di terzo ordine. Se la forma della legatura di valore o di frase che viene calcolata automaticamente non è ottimale, può essere modificata manualmente in due modi:

- specificando di quanto spostare i punti di controllo della curva Bézier calcolata automaticamente, o
- specificando esplicitamente le posizioni dei quattro punti di controllo richiesti per definire la curva desiderata.

Entrambi i metodi sono spiegati di seguito. Il primo metodo è più adatto se sono necessari solo piccoli aggiustamenti alla curva; il secondo può essere migliore per creare curve che riguardano soltanto una singola nota.

Curve Bézier cubiche

Le curve Bézier di terzo ordine o cubiche sono definite da quattro punti di controllo. Il primo e il quarto punto di controllo sono esattamente i punti di inizio e fine della curva. I due punti intermedi definiscono la forma. Sul web si trovano delle animazioni che mostrano come viene disegnata la curva, ma la seguente descrizione e immagine può comunque essere utile. La curva inizia dal primo punto di controllo e si dirige verso il secondo, piegandosi gradualmente per arrivare al terzo e poi al quarto, arrivando lì direttamente dal terzo punto di controllo. La curva è contenuta interamente nel quadrilatero definito dai quattro punti di controllo.



Traslazioni, rotazioni e ridimensionamenti dei punti di controllo risultano tutti esattamente nelle stesse operazioni sulla curva.

Specificare gli spostamenti dai punti di controllo correnti

In questo esempio la disposizione automatica della legatura di valore non è ottimale, e `\tieDown` non sarebbe di aiuto.

```
<<
  { e'1~ 1 }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
>>
```



Cambiando i punti di controllo della legatura con `\shape` consente di evitare le collisioni.

La sintassi di `\shape` è

```
[ - ] \shape spostamenti elemento
```

Questo comando riposiziona i punti di controllo di *elemento* di quanto indicato da *spostamenti*. L'argomento *spostamenti* è una lista di coppie di numeri o una lista di tali liste. Ogni elemento di una coppia rappresenta lo spostamento di una delle coordinate di un punto di controllo. Se *elemento* è una stringa, il risultato è `\once\override` per il tipo di grob specificato. Se *elemento* è un'espressione musicale, il risultato è la stessa espressione musicale con un'appropriata modifica.

In altre parole, la funzione `\shape` può comportarsi come un comando `\once\override` o un comando `\tweak` a seconda che l'argomento *elemento* sia il nome di un grob, come "Slur", o un'espressione musicale, come "(". L'argomento *spostamenti* specifica gli sfasamenti dei quattro punti di controllo nella forma di una lista di quattro coppie di valori (dx . dy) in unità di spazi rigo (o una lista di tali liste se la curva ha più di un segmento).

Il trattino iniziale è richiesto se e solo se si usa la forma con `\tweak`.

Dunque usando lo stesso esempio precedente e la forma `\once\override` di `\shape`, il seguente comando alzerà la legatura di valore di metà spazio rigo:

```
<<
  {
    \shape #'((0 . 0.5) (0 . 0.5) (0 . 0.5) (0 . 0.5)) Tie
    e'1~ 1
  }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
>>
```



Questo posizionamento è migliore, ma forse dovrebbe essere alzato di più al centro, come viene fatto nel prossimo esempio, dove stavolta si usa la forma alternativa `\tweak`:

```
<<
  {
    e'1~\shape #'((0 . 0.5) (0 . 1) (0 . 1) (0 . 0.5)) ~ e'
  }
\\
  \relative { r4 <g' c,> <g c,> <g c,> }
>>
```



```

\\
\relative { r4 <g' c,> <g c,> <g c,> }
>>

```



Per facilitare il processo di modifica viene fornita la funzione `\vshape`. Il suo nome vuol dire *visual shape* (forma visiva): si comporta esattamente come `\shape`, con la differenza che vengono mostrati anche i punti di controllo e il poligono.

```

\relative {
  c''8(\( a) e4 gis a\
  \vshape #'((0 . -0.3) (0.5 . -0.2)
             (0.5 . -0.3) (0 . -0.7)) PhrasingSlur
  c8(\( a) e4 gis a\
}

```



Si consiglia di cominciare con `\vshape` e aggiustare i valori finché non si ottiene una curva soddisfacente, quindi si può semplicemente togliere la lettera “v” in `\vshape`.

Si possono ridefinire anche due curve diverse che iniziano nello stesso momento musicale:

```

\relative {
  c''8(\( a) a'4 e c\
  \shape #'((0.7 . -0.4) (0.5 . -0.4)
            (0.3 . -0.3) (0 . -0.2)) Slur
  \shape #'((0 . 0) (0 . 0.5)
            (0 . 0.5) (0 . 0)) PhrasingSlur
  c8(\( a) a'4 e c\
}

```



La funzione `\shape` può sostituire anche i punti di controllo di curve che si estendono oltre le interruzioni di linea. A ogni pezzo della curva spezzata può essere assegnata la sua lista di spostamenti. Se non sono necessarie delle modifiche a un segmento in particolare, la lista vuota può servire come segnaposto. In questo esempio l'interruzione di linea fa sembrare doppia la legatura di portamento singola:

```

\relative {
  c'4( f g c
  \break
  d,4 c' f, c)
}

```





Cambiando le forme delle due metà della legatura di portamento diventa più chiaro che la legatura continua oltre l'interruzione di linea:

```
% () può essere usato come una scorciatoia per ((0 . 0) (0 . 0) (0 . 0) (0 . 0))
% nel caso in cui uno dei segmenti non necessiti di modifiche
\relative c' {
  \shape #'(
    (( 0 . 0) (0 . 0) (0 . 0) (0 . 1))
    ((0.5 . 1.5) (1 . 0) (0 . 0) (0 . -1.5))
  ) Slur
  c4( f g c
  \break
  d,4 c' f, c)
}
```



Se è richiesta una curva a forma di S, bisogna sempre definire manualmente i punti di controllo — LilyPond non sceglierà mai tali forme automaticamente.

```
\relative c'' {
  c8( e b-> f d' a e-> g)
  \shape #'((0 . -1) (5.5 . -0.5) (-5.5 . -10.5) (0 . -5.5)) PhrasingSlur
  c8\ ( e b-> f d' a e-> g\ )
}
```



Specificare i punti di controllo esplicitamente

Le coordinate dei punti di controllo della curva di Bézier sono specificate in unità di spazi rigo. La coordinata X è relativa al punto di riferimento della nota a cui la legatura si attacca, mentre la coordinata Y è relativa alla linea centrale del rigo. Le coordinate sono specificate come una lista di quattro coppie di numeri decimali (reali). Un possibile approccio consiste nel valutare prima le coordinate delle due estremità e poi indovinare i due punti intermedi. I valori ottimali vengono quindi trovati per tentativi. Tenere conto del fatto che, in caso di ulteriori modifiche alla musica o alla formattazione, questi valori potrebbero dover essere nuovamente cambiati manualmente.

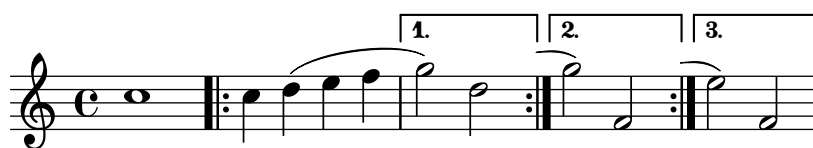
Una situazione in cui è preferibile specificare i punti di controllo esplicitamente invece di specificarne gli spostamenti è quella in cui devono essere indicati per una singola nota. Il prossimo esempio mostra un modo in cui rappresentare una legatura di portamento che si estende nelle sezioni alternative di una volta.

```
\relative {
```

```

c''1
\repeat volta 3 { c4 d( e f }
\alternative {
  \volta 1 { g2) d }
  \volta 2 {
    g2
    % crea una legatura di portamento e spostala in una nuova posizione
    % il <> è solamente un accordo vuoto che serve a terminare la legatura
    -\tweak control-points #'((-2 . 3.8) (-1 . 3.9) (0 . 4) (1 . 3.4)) ( <> )
    f,
  }
  \volta 3 {
    e'2
    % crea una legatura di portamento e spostala in una nuova posizione
    -\tweak control-points #'((-2 . 3) (-1 . 3.1) (0 . 3.2) (1 . 2.4)) ( <> )
    f,
  }
}
}
}

```



Problemi noti e avvertimenti

Non è possibile modificare le forme di legature di valore e di portamento cambiando la proprietà `control-points` se ci sono molteplici legature nello stesso momento musicale – anche il comando `\tweak` non funzionerà in questo caso. Tuttavia si può sovrascrivere la proprietà `tie-configuration` di `TieColumn` per impostare l’inizio della linea e la direzione come desiderato.

Vedi anche

Guida al funzionamento interno: Sezione “`TieColumn`” in *Guida al Funzionamento Interno*.

5.5.5 Modifica degli estensori spezzati

Uso di `\alterBroken`

Quando un estensore va a capo o si spezza, ciascun pezzo eredita gli attributi dell'estensore originario. Dunque la normale modifica di un estensore spezzato applica le stesse modifiche a ciascuno dei suoi segmenti. Nell'esempio successivo, la sovrascrittura di `thickness` agisce sulla legatura di portamento su entrambi i lati rispetto all'interruzione di linea.

```
\relative c' ' {
  r2
  \once\override Slur.thickness = 10
  c8( d e f
  \break
  g8 f e d) r2
}
```



È possibile modificare indipendentemente l'aspetto di singoli pezzi di un estensore spezzato col comando `\alterBroken`. Questo può generare sia un `\override` che un `\tweak` della proprietà di un estensore.

La sintassi di `\alterBroken` è:

```
[-]\alterBroken proprietà valori elemento
```

L'argomento *valori* è una lista di valori, uno per ogni pezzo spezzato. Se *elemento* è il nome di un grob come `Slur` o `Staff.PianoPedalBracket`, il risultato è un `\override` del tipo di grob specificato. Se *elemento* è un'espressione musicale come `"` o `"["`, il risultato è la stessa espressione musicale modificata da un comando `\tweak`.

Il trattino iniziale è richiesto nella forma `\tweak`. Non aggiungerlo se `\alterBroken` viene usato come un `\override`.

Nel suo uso come `\override`, `\alterBroken` può essere preceduto da `\once` o `\temporary` e ripristinato con `\revert` seguito da *proprietà* (vedi Sezione "Intermediate substitution functions" in *Estendere*).

Il codice seguente applica un `\override` indipendente a ciascun segmento della legatura di portamento dell'esempio precedente:

```
\relative c' ' {
  r2
  \alterBroken thickness #'(10 1) Slur
  c8( d e f
  \break
  g8 f e d) r2
}
```





Il comando `\alterBroken` può essere usato con qualsiasi oggetto estensore, inclusi `Tie`, `PhrasingSlur`, `Beam` e `TextSpanner`. Per esempio, un editor che stia preparando un'edizione colta potrebbe voler indicare l'assenza di parte di una legatura di frase in un manoscritto tratteggiando solo il segmento che è stato aggiunto. L'esempio seguente mostra come farlo, in questo caso usando la forma `\tweak` del comando:

```
% Si usa per convenienza la lista vuota, perché è l'impostazione
% predefinita di dash-definition, producendo così una curva continua.
\relative {
  c'2-\alterBroken dash-definition #'(( ) ((0 1.0 0.4 0.75))) \e
  \break
  g2 e\
}
```



È importante comprendere che `\alterBroken` imposterà ogni pezzo di un estensore spezzato sul valore corrispondente specificato in *valori*. Quando il numero di valori è inferiore a quello dei pezzi, a qualsiasi pezzo ulteriore sarà assegnata una lista vuota. Ciò potrebbe portare a risultati indesiderati se la proprietà di formattazione non è impostata su una lista vuota per impostazione predefinita. In tali casi, a ogni segmento deve essere assegnato un valore appropriato.

Problemi noti e avvertimenti

Le interruzioni di linea possono verificarsi in punti diversi in seguito a modifiche della formattazione. Le impostazioni scelte per `\alterBroken` potrebbero non essere più adatte per un estensore che non va più a capo o è diviso in più segmenti di prima. Per proteggersi da questa situazione conviene fare un uso esplicito di `\break`.

Il comando `\alterBroken` non funziona nel caso di proprietà dell'estensore a cui si accede prima di line-breaking, come `direction`.

Vedi anche

Estendere LilyPond: Sezione “Difficult tweaks” in *Estendere*.

5.5.6 Contenitori unpure-pure

Nota: Sebbene questa sottosezione rifletta lo stato corrente, l'esempio dato è discutibile e non mostra alcun effetto.

I contenitori unpure-pure (impuri-puri) sono utili per sovrascrivere i calcoli di spaziatura dell'asse *Y* – in particolare *Y-offset* e *Y-extent* – tramite una funzione Scheme invece che con un numero o una coppia di numeri.

Per alcuni grob, il cui *Y-extent* è basato sulla proprietà `stencil`, sovrascrivere tale proprietà renderà necessario un'ulteriore sovrascrittura di *Y-extent* tramite un contenitore unpure-pure.

Quando una funzione sovrascrive Y-offset e/o Y-extent, si presume che ciò innescherà il calcolo delle interruzioni di linea troppo presto durante la compilazione. Dunque la funzione non viene affatto calcolata (e di solito restituirà un valore di '0' o '(0 . 0)') e ciò può causare delle collisioni. Una funzione 'pure' (pura) non avrà effetto su proprietà, oggetti o "suicidi" dei grob e quindi tutto ciò che è collegato al suo asse Y sarà sempre calcolato correttamente.

Attualmente, ci sono circa trenta funzioni che sono già considerate 'pure' e i contenitori unpure-pure sono un modo per impostare le funzioni non ancora presenti in questa lista come 'pure'. La funzione 'pure' viene calcolata *prima* di qualsiasi interruzione di linea e quindi la spaziatura orizzontale può essere regolata "in tempo". La funzione 'unpure' viene invece calcolata *dopo* le interruzioni di linea.

Nota: Dato che è difficile sapere sempre quali funzioni si trovano in questa lista, consigliamo a chi crea funzioni 'pure' di non usare i grob Beam o VerticalAlignment.

Un contenitore unpure-pure viene costruito nel modo seguente:

```
(ly:make-unpure-pure-container f0 f1)
```

dove *f0* è una funzione che prende *n* argomenti ($n \geq 1$) e il primo argomento deve sempre essere il grob. Questa è la funzione che restituisce il vero risultato. *f1* è la funzione etichettata come 'pure' che prende $n + 2$ argomenti. Di nuovo, il primo argomento deve sempre essere il grob, ma il secondo e il terzo sono il punto di partenza, 'start', e quello di arrivo, 'end'.

start e *end* sono, per tutti gli intenti e scopi, valori fittizi che contano solo per gli estensori, o Spanners, (ovvero Hairpin o Beam), che possono restituire calcoli diversi dell'altezza in base a una colonna di inizio e di fine.

Il resto sono altri argomenti della prima funzione (che potrebbero essere nessuno se $n = 1$).

I risultati della seconda funzione sono usati come un'approssimazione del valore necessario che viene poi usato dalla prima funzione per ottenere il valore reale, che viene infine usato per la messa a punto, molto più tardi nel corso del processo di spaziatura.

```
#(define (square-line-circle-space grob)
  (let* ((pitch (ly:event-property (ly:grob-property grob 'cause) 'pitch))
        (notename (ly:pitch-notename pitch)))
    (if (= 0 (modulo notename 2))
        (make-circle-stencil 0.5 0.0 #t)
        (make-filled-box-stencil '(0 . 1.0)
                                   '(-0.5 . 0.5)))))
```

```
squareLineCircleSpace = {
  \override NoteHead.stencil = #square-line-circle-space
}
```

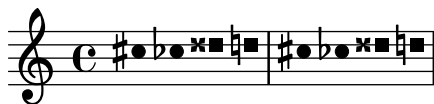
```
smartSquareLineCircleSpace = {
  \squareLineCircleSpace
  \override NoteHead.Y-extent =
    #(ly:make-unpure-pure-container
      ly:grob::stencil-height
      (lambda (grob start end) (ly:grob::stencil-height grob)))
}
```

```
\new Voice \with { \remove Stem_engraver }
\relative c'' {
```

```

\squareLineCircleSpace
cis4 ces disis d
\smartSquareLineCircleSpace
cis4 ces disis d
}

```



Nella prima misura, senza il contenitore `unpure-pure`, il motore della spaziatura non conosce la larghezza della testa di nota e lascia che collida con le alterazioni. Nella seconda misura, che usa i contenitori `unpure-pure`, il motore della spaziatura conosce la larghezza della testa di nota e evita le collisioni allungando la linea dello spazio necessario.

Di solito per semplici calcoli si possono usare funzioni quasi identiche per entrambe le parti, ‘`unpure`’ e ‘`pure`’, cambiando soltanto il numero di argomenti passati e la portata della funzione. Questo caso d’uso è così frequente che `ly:make-unpure-pure-container` costruisce tale seconda funzione automaticamente quando è richiamata con un solo argomento.

Nota: Se una funzione viene etichettata come ‘`pure`’ ma viene fuori che non lo è, i risultati possono essere imprevedibili.

5.6 Uso delle funzioni musicali

Quando le modifiche devono essere riusate con diverse espressioni musicali, è spesso conveniente fare in modo che la modifica faccia parte di una *funzione musicale*. In questa sezione, trattiamo soltanto le funzioni di *sostituzione*, dove l’obiettivo è sostituire una variabile in un punto del codice di input di LilyPond. Altre funzioni più complesse sono descritte in Sezione “Music functions” in *Estendere*.

5.6.1 Sintassi della funzione di sostituzione

Creare una funzione che sostituisca una variabile presente nel codice LilyPond è semplice. La forma generica di queste funzioni è:

```

funzione =
#(define-music-function
  (arg1 arg2 ...)
  (tipo1? tipo2? ...)
  #{
    ...musica...
  #})

```

dove

<code>argN</code>	L’ <i>nesimo</i> argomento.
<code>tipoN?</code>	Un <i>tipo di predicato</i> Scheme per il quale <code>argN</code> deve ritornare <code>#t</code> .
<code>...musica...</code>	Il normale input LilyPond, usando <code>\$</code> (nei punti in cui sono consentiti solo i costrutti LilyPond) o <code>#</code> (per usarlo come un valore Scheme o come argomento di una funzione musicale o come musica all’interno di liste di musica) per riferirsi agli argomenti (es: ‘ <code>#arg1</code> ’).

La lista del tipo di predicati è obbligatoria. Alcuni dei tipi di predicati più comuni usati nelle funzioni musicali sono:

```
boolean?
cheap-list?  (usarlo al posto di 'list?' per un'elaborazione più veloce)
ly:duration?
ly:music?
ly:pitch?
markup?
number?
pair?
string?
symbol?
```

Un elenco dei tipi di predicati disponibili si trova in Sezione A.23 [Tipi di predicati predefiniti], pagina 822. Sono consentiti anche tipi di predicati definiti dagli utenti.

Vedi anche

Guida alla notazione: Sezione A.23 [Tipi di predicati predefiniti], pagina 822.

Estendere LilyPond: Sezione “Music functions” in *Estendere*.

File installati: `lily/music-scheme.cc`, `scm/c++.scm`, `scm/lily.scm`.

5.6.2 Esempi della funzione di sostituzione

Questa sezione presenta alcuni esempi di funzione di sostituzione. L'obiettivo non è fornire un elenco esaustivo, ma mostrare soltanto alcune delle possibilità di semplici funzioni di sostituzione.

Nel primo esempio, viene definita una funzione che semplifica il padding di un oggetto TextScript:

```
padText =
#(define-music-function
  (padding)
  (number?)
  #{
    \once \override TextScript.padding = #padding
  #})

\relative {
  c' '4^"più mosso" b a b
  \padText #1.8
  c4^"più mosso" b a b
  \padText #2.6
  c4^"più mosso" b a b
}
```



Oltre ai numeri, si possono usare come argomenti delle funzioni musicali anche le espressioni musicali, come le note:

```
custosNote =
#(define-music-function
  (note)
```



```

(ly:music?)
#{
  \tweak NoteHead.stencil #ly:text-interface::print
  \tweak NoteHead.text
    \markup \musicglyph "custodes.mensural.u0"
  \tweak Stem.stencil ##f
  #note
#})

\relative { c'4 d e f \custosNote g }

```



Entrambe queste funzioni sono semplici espressioni singole dove manca soltanto l'ultimo elemento di una chiamata di una funzione o di una sovrascrittura. Per queste particolari definizioni di funzione, esiste una sintassi alternativa più semplice, che consiste nello scrivere interamente la parte costante dell'espressione e sostituire il suo elemento finale mancante con `\etc`:

```

padText =
  \once \override TextScript.padding = \etc

\relative {
  c'4^"più mosso" b a b
  \padText #1.8
  c4^"più mosso" b a b
  \padText #2.6
  c4^"più mosso" b a b
}

```



```

custosNote =
  \tweak NoteHead.stencil #ly:text-interface::print
  \tweak NoteHead.text
    \markup \musicglyph "custodes.mensural.u0"
  \tweak Stem.stencil ##f
  \etc

\relative { c'4 d e f \custosNote g }

```



Si possono definire funzioni di sostituzione con molteplici argomenti:

```

tempoPadded =
#(define-music-function
  (padding tempotext)
  (number? markup?)
  #{

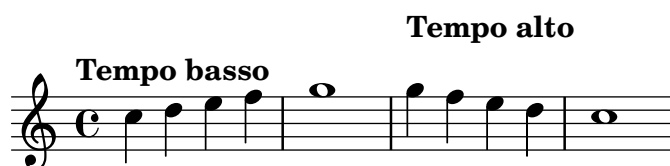
```

```

\once \override Score.MetronomeMark.padding = #padding
\tempo \markup { \bold #tempotext }
#})

\relative {
  \tempo \markup { "Tempo basso" }
  c' '4 d e f g1
  \tempoPadded #4.0 "Tempo alto"
  g4 f e d c1
}

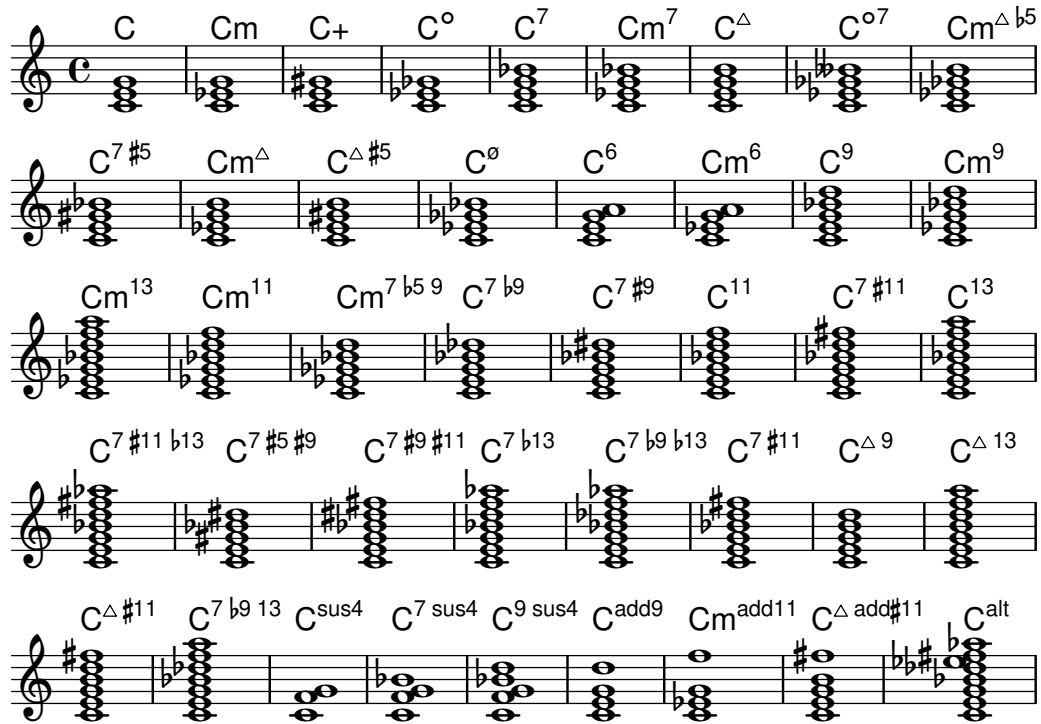
```



Appendice A Tabelle del manuale della notazione

A.1 Grafico dei nomi degli accordi

Il grafico seguente mostra il sistema standard di LilyPond di rappresentazione dei nomi degli accordi, insieme alle altezze che li compongono. Ulteriori sistemi di nomi (non supportati) sono illustrati nel frammento “Chord names alternative” in Sezione “Chord notation” in *Frammenti di codice*, compresa la notazione ispirata da Harald Banter (1982) che fu usata come predefinita nelle prime versioni di LilyPond (fino alla versione 1.7).

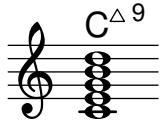




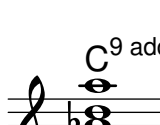
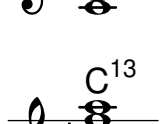






A.2 Modificatori degli accordi comuni

La tabella seguente mostra i modificatori degli accordi che possono essere usati per generare gli accordi comuni.

Tipo	Intervallo	Modificatore	Esempio	Output
Maggiore	Terza maggiore, quinta perfetta	5 o niente	c1:5	
Minore	Terza minore, quinta perfetta	m o m5	c1:m	
Aumentato	Terza maggiore, quinta aumentata	aug	c1:aug	

Diminuito	Terza minore, quinta diminuita	dim	c1:dim	
Settima dominante	Triade maggiore, settima minore	7	c1:7	
Settima maggiore	Triade maggiore, settima maggiore	maj7 o maj	c1:maj7	
Settima minore	Triade minore, settima minore	m7	c1:m7	
Settima diminuita	Triade diminuita, settima diminuita	dim7	c1:dim7	
Settima aumentata	Triade aumentata, settima minore	aug7	c1:aug7	
Settima semidiminuita	Triade diminuita, settima minore	m7.5-	c1:m7.5-	
Minore di settima maggiore	Triade minore, settima maggiore	m7+	c1:m7+	
Sesta maggiore	Triade maggiore, sesta	6	c1:6	
Sesta minore	Triade minore, sesta	m6	c1:m6	
Nona dominante	Settima dominante, nona maggiore	9	c1:9	

Nona maggiore	Settima maggiore, nona maggiore	maj9	c1:maj9	
Nona minore	Settima minore, nona maggiore	m9	c1:m9	
Undicesima dominante	Nona dominante, undicesima perfetta	11	c1:11	
Undicesima maggiore	Nona maggiore, undicesima perfetta	maj11	c1:maj11	
Undicesima minore	Nona minore, undicesima perfetta	m11	c1:m11	
Tredicesima dominante	Nona dominante, tredicesima maggiore	13	c1:13	
Tredicesima dominante	Undicesima dominante, tredicesima maggiore	13.11	c1:13.11	
Tredicesima maggiore	Undicesima maggiore, tredicesima maggiore	maj13.11	c1:maj13.11	
Tredicesima minore	Undicesima minore, tredicesima maggiore	m13.11	c1:m13.11	
Sospeso di seconda	Seconda maggiore, quinta perfetta	sus2	c1:sus2	
Sospeso di quarta	Quarta perfetta, quinta perfetta	sus4	c1:sus4	

Power chord
(bicordo)

Quinta perfetta

1.5

c1:5



Power chord
(tricordo)

Quinta perfetta,
ottava

1.5.8

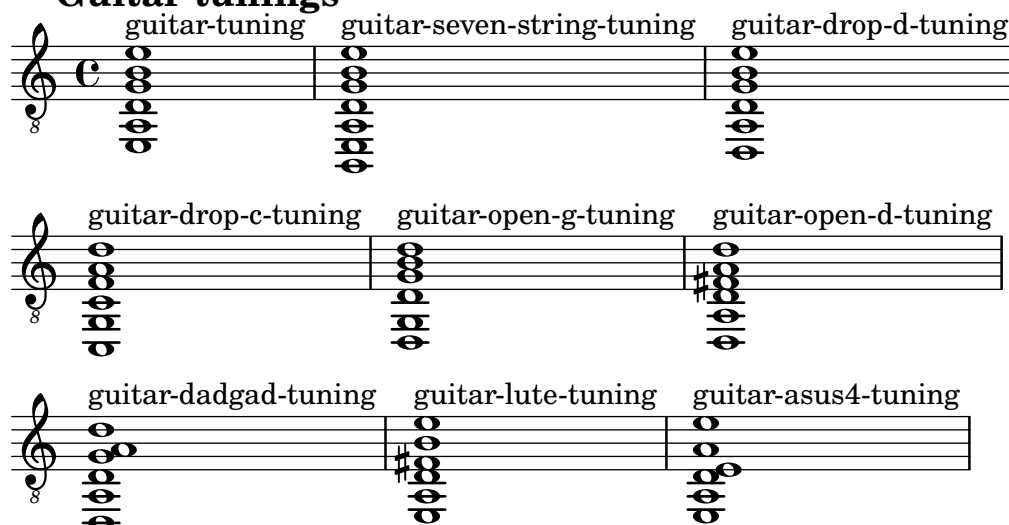
c1:5.8



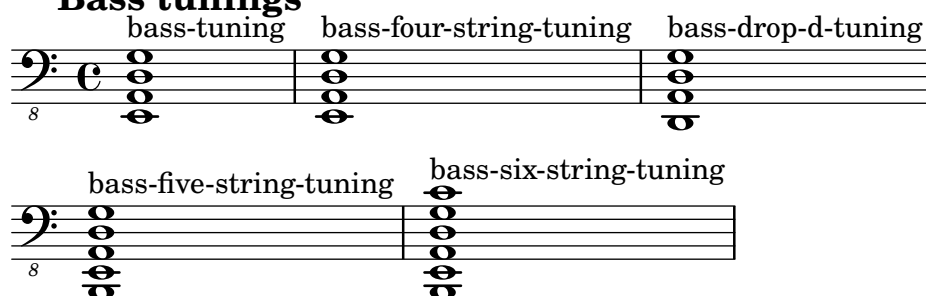
A.3 Accordature predefinite

Il grafico seguente mostra le accordature predefinite degli strumenti a corda.

Guitar tunings



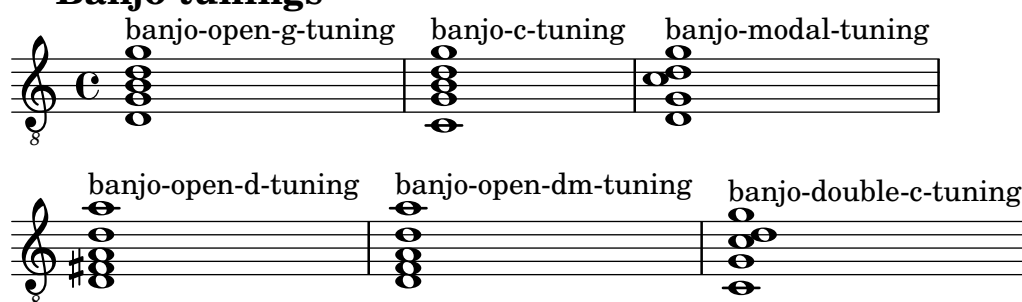
Bass tunings



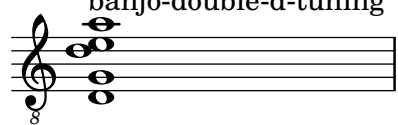
Mandolin tunings



Banjo tunings

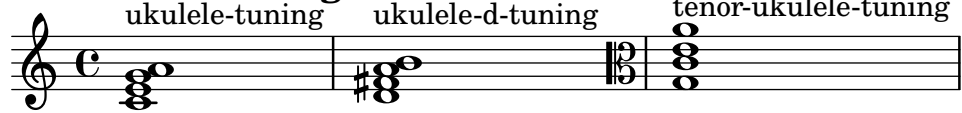


banjo-double-d-tuning

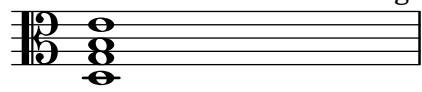


Ukulele tunings

ukulele-tuning ukulele-d-tuning tenor-ukulele-tuning




baritone-ukulele-tuning



Orchestral string tunings

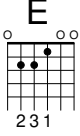
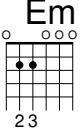
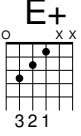
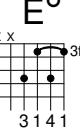
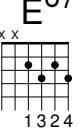
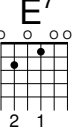
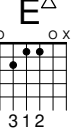
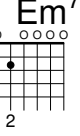
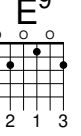
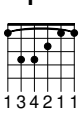
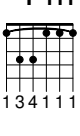
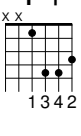
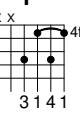
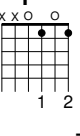
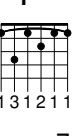
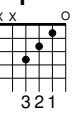
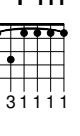

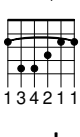
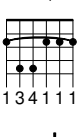
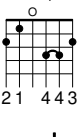
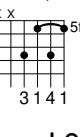
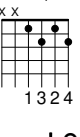
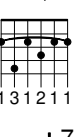



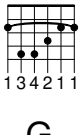
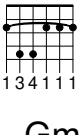
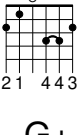
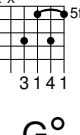
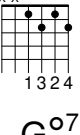
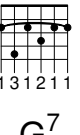

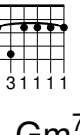
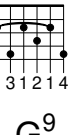
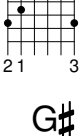

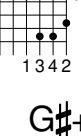
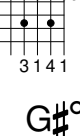

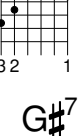
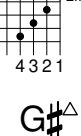
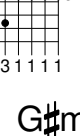
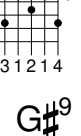
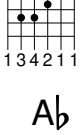
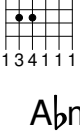
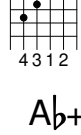




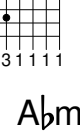
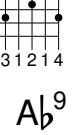
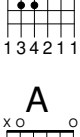
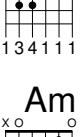
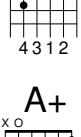
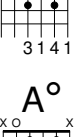
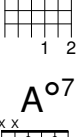
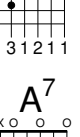

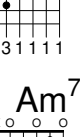
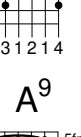
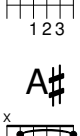
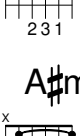
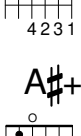
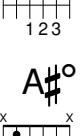
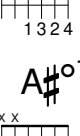
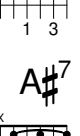
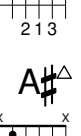
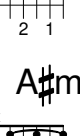
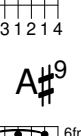
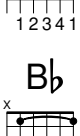
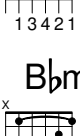
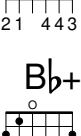
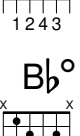
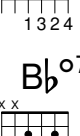
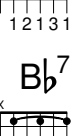
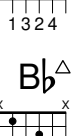
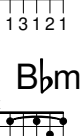
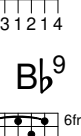
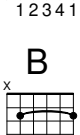
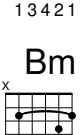
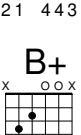
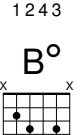
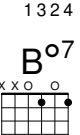
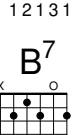
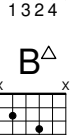
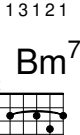
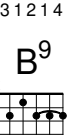









violin-tuning viola-tuning cello-tuning double-bass-tuning



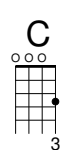
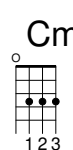
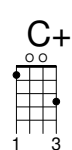
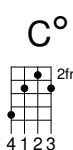
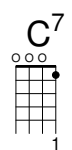
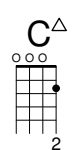

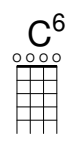
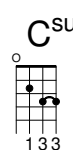
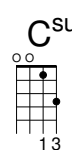
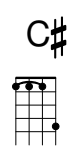
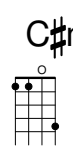
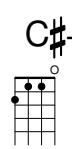
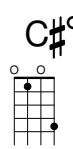
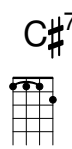
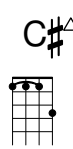
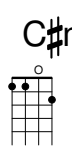

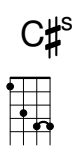
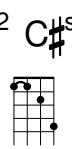
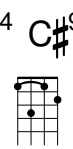

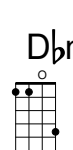
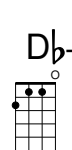
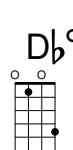


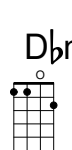

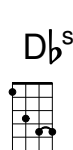
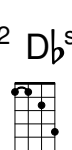

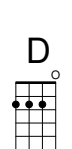
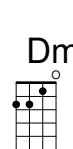
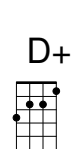
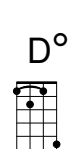


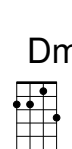
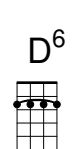
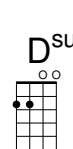
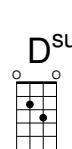

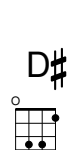
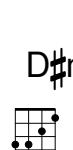
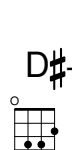
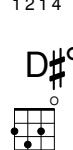
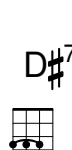
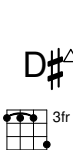
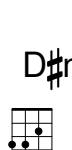
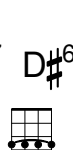
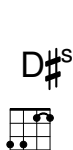
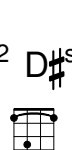
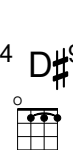
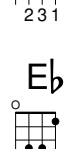
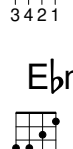
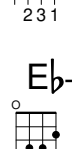
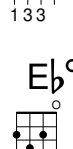
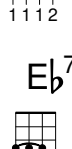
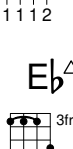
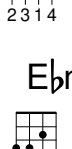
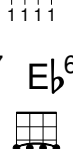
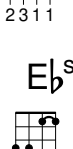
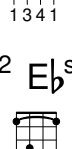
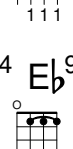


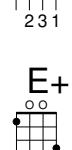
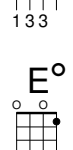
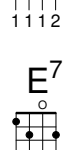
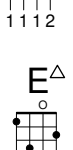

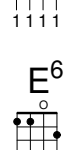
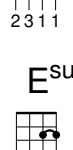
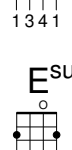
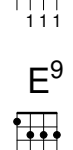

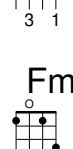
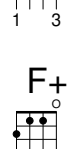
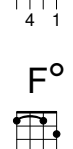
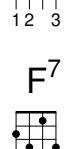

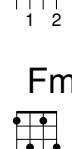
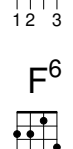
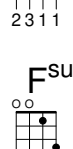
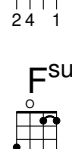
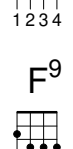


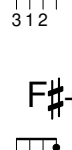
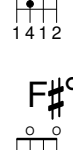
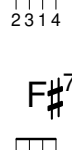
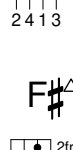

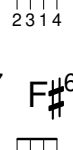
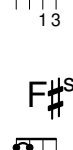
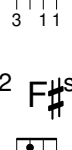
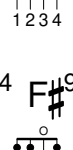
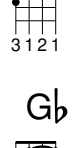

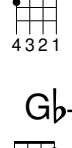
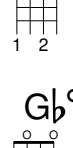
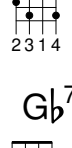


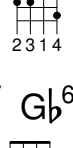

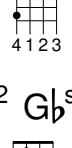
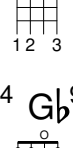
A.4 Diagrammi degli accordi predefiniti

Diagrammi per chitarra

C 3 2 1	Cm 1 3 4 2 1	C+ 2 1 1 4	C° 1 2 4 3	C°7 1 3 2 4	C7 3 2 4 1	C△ 3 2	Cm7 1 3 1 2 1	C9 2 1 3 3 3
C# 3 1 2 1	C#m 2 1 3	C#+ 4 3 1 2	C#° 3 4	C#°7 1 3 2 4	C#7 2 3 1 4	C#△ 4 3 1 1 1	C#m7 4 2 1	C#9 2 1 3 3 3
D♭ 3 1 2 1	D♭m 2 1 3	D♭+ 4 3 1 2	D♭° 3 4	D♭°7 1 3 2 4	D♭7 2 3 1 4	D♭△ 4 3 1 1 1	D♭m7 4 2 1	D♭9 2 1 3 3 3
D 1 3 2	Dm 2 3 1	D+ 2 3 1	D° 1 3 1	D°7 1 2	D7 2 1 3	D△ 1 2 3	Dm7 2 1 1	D9 2 1 3 3 3
D# 3 1 2 1	D#m 3 2 4 1	D#+ 3 2 1 4	D#° 3 1 4 1	D#°7 1 3 2 4	D#7 1 3 2 4	D#△ 1 2 3 4	D#m7 1 2 3 4	D#9 2 1 3 3 3
E♭ 3 1 2 1	E♭m 3 2 4 1	E♭+ 3 2 1 4	E♭° 3 1 4 1	E♭°7 1 3 2 4	E♭7 1 3 2 4	E♭△ 1 2 3 4	E♭m7 1 2 3 4	E♭9 2 1 3 3 3

 E 2 3 1	 Em 2 3	 E+ 3 2 1	 E° 3 1 4 1	 E°7 1 3 2 4	 E7 2 1	 EΔ 3 1 2	 Em7 2	 E9 2 1 3
 F 1 3 4 2 1 1	 Fm 1 3 4 1 1 1	 F+ 1 3 4 2	 F° 3 1 4 1	 F°7 1 2	 F7 1 3 1 2 1 1	 FΔ 3 2 1	 Fm7 1 3 1 1 1 1	 F9 1 3 1 2 1 4
 F# 1 3 4 2 1 1	 F#m 1 3 4 1 1 1	 F#+ 2 1 4 4 3	 F#° 3 1 4 1	 F#°7 1 3 2 4	 F#7 1 3 1 2 1 1	 F#Δ 4 3 2 1	 F#m7 1 3 1 1 1 1	 F#9 1 3 1 2 1 4
 Gb 1 3 4 2 1 1	 Gbm 1 3 4 1 1 1	 Gb+ 2 1 4 4 3	 Gb° 3 1 4 1	 Gb°7 1 3 2 4	 Gb7 1 3 1 2 1 1	 GbΔ 4 3 2 1	 Gbm7 1 3 1 1 1 1	 Gb9 1 3 1 2 1 4
 G 2 1 3	 Gm 1 3 4 1 1 1	 G+ 1 3 4 2	 G° 3 1 4 1	 G°7 1 3 2 4	 G7 3 2 1	 GΔ 4 3 2 1	 Gm7 1 3 1 1 1 1	 G9 1 3 1 2 1 4
 G# 1 3 4 2 1 1	 G#m 1 3 4 1 1 1	 G#+ 4 3 1 2	 G#° 3 1 4 1	 G#°7 1 2	 G#7 1 3 1 2 1 1	 G#Δ 1 1 1 3	 G#m7 1 3 1 1 1 1	 G#9 1 3 1 2 1 4
 Ab 1 3 4 2 1 1	 Abm 1 3 4 1 1 1	 Ab+ 4 3 1 2	 Ab° 3 1 4 1	 Ab°7 1 2	 Ab7 1 3 1 2 1 1	 AbΔ 1 1 1 3	 Abm7 1 3 1 1 1 1	 Ab9 1 3 1 2 1 4
 A 1 2 3	 Am 2 3 1	 A+ 4 2 3 1	 A° 1 2 3	 A°7 1 3 2 4	 A7 1 3	 AΔ 2 1 3	 Am7 2 1	 A9 1 3 1 2 1 4
 A# 1 2 3 4 1	 A#m 1 3 4 2 1	 A#+ 2 1 4 4 3	 A#° 1 2 4 3	 A#°7 1 3 2 4	 A#7 1 2 1 3 1	 A#Δ 1 3 2 4	 A#m7 1 3 1 2 1	 A#9 1 3 1 2 1 4
 Bb 1 2 3 4 1	 Bbm 1 3 4 2 1	 Bb+ 2 1 4 4 3	 Bb° 1 2 4 3	 Bb°7 1 3 2 4	 Bb7 1 2 1 3 1	 BbΔ 1 3 2 4	 Bbm7 1 3 1 2 1	 Bb9 1 3 1 2 1 4
 B 1 2 3 4 1	 Bm 1 3 4 2 1	 B+ 2 1	 B° 1 2 4 3	 B°7 1 2	 B7 2 1 3 4	 BΔ 1 3 2 4	 Bm7 1 3 1 2 1	 B9 2 1 3 3 3

Diagrammi per ukulele

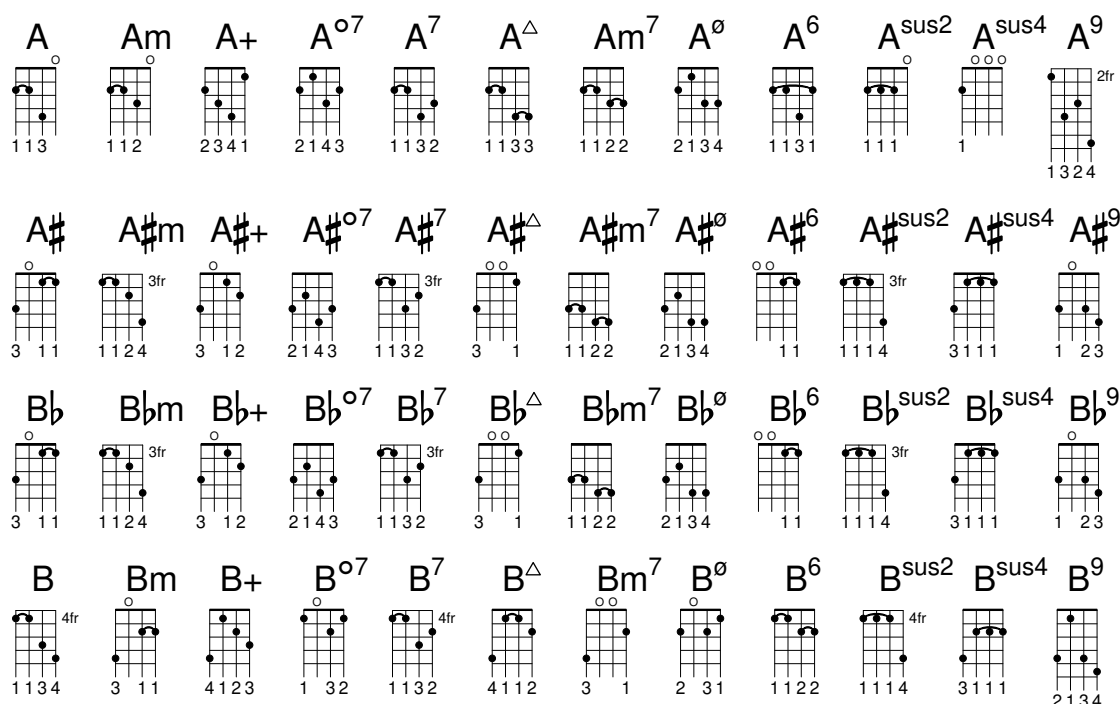
 C	 Cm	 C+	 C°	 C ⁷	 C ^Δ	 Cm ⁷	 C ⁶	 C ^{sus2}	 C ^{sus4}	 C ⁹
 C [#]	 C [#] m	 C ^{#+}	 C [#] °	 C ^{#7}	 C ^{#Δ}	 C [#] m ⁷	 C ^{#6}	 C ^{#sus2}	 C ^{#sus4}	 C ^{#9}
 D ^b	 D ^b m	 D ^{b+}	 D ^b °	 D ^{b7}	 D ^{bΔ}	 D ^b m ⁷	 D ^{b6}	 D ^{b sus2}	 D ^{b sus4}	 D ^{b9}
 D	 Dm	 D+	 D°	 D ⁷	 D ^Δ	 Dm ⁷	 D ⁶	 D ^{sus2}	 D ^{sus4}	 D ⁹
 D [#]	 D [#] m	 D ^{#+}	 D [#] °	 D ^{#7}	 D ^{#Δ}	 D [#] m ⁷	 D ^{#6}	 D ^{# sus2}	 D ^{# sus4}	 D ^{#9}
 E ^b	 E ^b m	 E ^{b+}	 E ^b °	 E ^{b7}	 E ^{bΔ}	 E ^b m ⁷	 E ^{b6}	 E ^{b sus2}	 E ^{b sus4}	 E ^{b9}
 E	 Em	 E+	 E°	 E ⁷	 E ^Δ	 Em ⁷	 E ⁶	 E ^{sus2}	 E ^{sus4}	 E ⁹
 F	 Fm	 F+	 F°	 F ⁷	 F ^Δ	 Fm ⁷	 F ⁶	 F ^{sus2}	 F ^{sus4}	 F ⁹
 F [#]	 F [#] m	 F ^{#+}	 F [#] °	 F ^{#7}	 F ^{#Δ}	 F [#] m ⁷	 F ^{#6}	 F ^{# sus2}	 F ^{# sus4}	 F ^{#9}
 G ^b	 G ^b m	 G ^{b+}	 G ^b °	 G ^{b7}	 G ^{bΔ}	 G ^b m ⁷	 G ^{b6}	 G ^{b sus2}	 G ^{b sus4}	 G ^{b9}

 1 3 2	 2 3 1	 2 3 1	 1 4 2	 2 1 3	 1 2 3	 2 1 1	 1 2	 1 2	 1 2 3	 2 3 1 4
 4 1 3 2	 1 3 4 2	 1 3	 1 2 4 3	 1 3 2 4	 1 2 3 4	 1 4 2 3	 1 3 2 4	 1 3 4 1	 1 3 4 4	 2 3 1 4
 4 1 3 2	 1 3 4 2	 1 3	 1 2 4 3	 1 3 2 4	 1 2 3 4	 1 4 2 3	 1 3 2 4	 1 3 4 1	 1 3 4 4	 2 3 1 4
 2 1	 2	 3 1 2	 1 2 4 4	 1	 1 2	 1 3 2 4	 1 3 2 4	 1 3 2	 1 2	 1 2
 3 2 1 1	 3 1 1 1	 4 3 2 1	 3 1 2	 1 2 1 1	 2 2 1 1	 1 1 1 1	 2 1 1	 3 1 1	 2 3 1 1	 1 3 2 4
 3 2 1 1	 3 1 1 1	 4 3 2 1	 3 1 2	 1 2 1 1	 2 2 1 1	 1 1 1 1	 2 1 1	 3 1 1	 2 3 1 1	 1 3 2 4
 3 2 1 1	 3 1 1 1	 2 3 1	 4 1 2 3	 1 2 1 1	 2 2 1 1	 1 1 1 1	 1 4 2 3	 4 1 3 2	 2 3 1 1	 1 3 2 4

Diagrammi per mandolino

 4 1 2	 1 1 2 4	 4 1 2 3	 2 1 4 3	 4 2 1	 4 1 1 2	 1 1 2 2	 3 1 4 2	 1 1 2 2	 3 1 1	 3 1 1 1	 1 3 2
 4 2 3 1	 2 3 1	 4 1	 2 1 1	 4 2 1 3	 4 1 1 2	 1 1 2 2	 3 1 4 2	 1 1 2 2	 1 1 3 4	 3 1 1 1	 2 1 3 4
 4 2 3 1	 2 3 1	 4 1	 2 1 1	 4 2 1 3	 4 1 1 2	 1 1 2 2	 3 1 4 2	 1 1 2 2	 1 1 3 4	 3 1 1 1	 2 1 3 4

D	Dm	D+	D ^{o7}	D ⁷	D ^Δ	Dm ⁷	D [∅]	D ⁶	D ^{sus2}	D ^{sus4}	D ⁹
1 2	2 1	3 12	1 32	1 32	1 42	2 31	1 32	1 23	1	1 2	4 2 1
D [#]	D [#] m	D [#] +	D ^{#o7}	D ^{#7}	D ^{#Δ}	D ^{#m7}	D ^{#∅}	D ^{#6}	D ^{#sus2}	D ^{#sus4}	D ^{#9}
3 1 1 4	3 1 1 2	1 2 3	2 1 4 3	2 1 4 3	2 1 4 3	3 1 4 2	2 1 4 3	2 1 3 4	3 1 1 1	3 1 1 4	2 1 3 4
E ^b	E ^b m	E ^b +	E ^{b o7}	E ^{b7}	E ^{bΔ}	E ^{b m7}	E ^{b ∅}	E ^{b6}	E ^{b sus2}	E ^{b sus4}	E ^{b9}
3 1 1 4	3 1 1 2	1 2 3	2 1 4 3	2 1 4 3	2 1 4 3	3 1 4 2	2 1 4 3	2 1 3 4	3 1 1 1	3 1 1 4	2 1 3 4
E	Em	E+	E ^{o7}	E ⁷	E ^Δ	Em ⁷	E [∅]	E ⁶	E ^{sus2}	E ^{sus4}	E ⁹
1 2 3	2 3	1 2 3 4	2 1 4 3	1 2	1 1 2	2	1	1 3 2	3 1 1 1	3 1	2 1 3 4
F	Fm	F+	F ^{o7}	F ⁷	F ^Δ	Fm ⁷	F [∅]	F ⁶	F ^{sus2}	F ^{sus4}	F ⁹
2 3 1	1 3 4 1	1 2 3 4	1 3 2	2 1 3 1	2 3 4 1	1 1 3 1	1 1 2 1	2 3 1	3 4 1	4 2 1 1	2 1 3 4
F [#]	F [#] m	F [#] +	F ^{# o7}	F ^{#7}	F ^{#Δ}	F ^{# m7}	F ^{# ∅}	F ^{#6}	F ^{# sus2}	F ^{# sus4}	F ^{#9}
2 3 4 1	1 3 4 1	1 2 3 4	2 1 4 3	2 1 3 1	2 3 4 1	1 1 3 1	1 1 2 1	3 1 4 2	3 1 1 1	4 2 1 1	2 1 3
G ^b	G ^b m	G ^b +	G ^{b o7}	G ^{b7}	G ^{bΔ}	G ^{b m7}	G ^{b ∅}	G ^{b6}	G ^{b sus2}	G ^{b sus4}	G ^{b9}
2 3 4 1	1 3 4 1	1 2 3 4	2 1 4 3	2 1 3 1	2 3 4 1	1 1 3 1	1 1 2 1	3 1 4 2	3 1 1 1	4 2 1 1	2 1 3
G	Gm	G+	G ^{o7}	G ⁷	G ^Δ	Gm ⁷	G [∅]	G ⁶	G ^{sus2}	G ^{sus4}	G ⁹
1 2	1 3	1 2 3	2 1 4 3	2 1	1 1	1 1	1 1 2 1	2	3	1 1	1 4
G [#]	G [#] m	G [#] +	G ^{# o7}	G ^{#7}	G ^{#Δ}	G ^{# m7}	G ^{# ∅}	G ^{#6}	G ^{# sus2}	G ^{# sus4}	G ^{#9}
1 1 3 4	1 1 2 4	1 2 3 4	1 3 2	1 1 3 2	1 1 3 3	1 1 2 2	1 2 2	1 1 3 1	1 1 1 4	1 1 3 4	1 3 2 4
A ^b	A ^b m	A ^b +	A ^{b o7}	A ^{b7}	A ^{bΔ}	A ^{b m7}	A ^{b ∅}	A ^{b6}	A ^{b sus2}	A ^{b sus4}	A ^{b9}
1 1 3 4	1 1 2 4	1 2 3 4	1 3 2	1 1 3 2	1 1 3 3	1 1 2 2	1 2 2	1 1 3 1	1 1 1 4	1 1 3 4	1 3 2 4



A.5 Formati carta predefiniti

I formati carta sono definiti nel file `scm/paper.scm`.

ISO 216, A series

"a10"	26 mm x 37 mm (1.02 in x 1.46 in)
"a9"	37 mm x 52 mm (1.46 in x 2.05 in)
"a8"	52 mm x 74 mm (2.05 in x 2.91 in)
"a7"	74 mm x 105 mm (2.91 in x 4.13 in)
"a6"	105 mm x 148 mm (4.13 in x 5.83 in)
"a5"	148 mm x 210 mm (5.83 in x 8.27 in)
"a4"	210 mm x 297 mm (8.27 in x 11.69 in)
"a3"	297 mm x 420 mm (11.69 in x 16.54 in)
"a2"	420 mm x 594 mm (16.54 in x 23.39 in)
"a1"	594 mm x 841 mm (23.39 in x 33.11 in)
"a0"	841 mm x 1189 mm (33.11 in x 46.81 in)

Two extended sizes as defined in DIN 476

"2a0"	1189 mm x 1682 mm (46.81 in x 66.22 in)
"4a0"	1682 mm x 2378 mm (66.22 in x 93.62 in)

ISO 216, B series

"b10"	31 mm x 44 mm (1.22 in x 1.73 in)
"b9"	44 mm x 62 mm (1.73 in x 2.44 in)
"b8"	62 mm x 88 mm (2.44 in x 3.46 in)
"b7"	88 mm x 125 mm (3.46 in x 4.92 in)
"b6"	125 mm x 176 mm (4.92 in x 6.93 in)
"b5"	176 mm x 250 mm (6.93 in x 9.84 in)
"b4"	250 mm x 353 mm (9.84 in x 13.90 in)
"b3"	353 mm x 500 mm (13.90 in x 19.69 in)
"b2"	500 mm x 707 mm (19.69 in x 27.83 in)
"b1"	707 mm x 1000 mm (27.83 in x 39.37 in)
"b0"	1000 mm x 1414 mm (39.37 in x 55.67 in)

ISO 269, C series

"c10"	28 mm x 40 mm (1.10 in x 1.57 in)
"c9"	40 mm x 57 mm (1.57 in x 2.24 in)
"c8"	57 mm x 81 mm (2.24 in x 3.19 in)
"c7"	81 mm x 114 mm (3.19 in x 4.49 in)
"c6"	114 mm x 162 mm (4.49 in x 6.38 in)
"c5"	162 mm x 229 mm (6.38 in x 9.02 in)
"c4"	229 mm x 324 mm (9.02 in x 12.76 in)
"c3"	324 mm x 458 mm (12.76 in x 18.03 in)
"c2"	458 mm x 648 mm (18.03 in x 25.51 in)
"c1"	648 mm x 917 mm (25.51 in x 36.10 in)
"c0"	917 mm x 1297 mm (36.10 in x 51.06 in)

North American paper sizes

"junior-legal"	5.0 in x 8.0 in (127 mm x 203 mm)
"legal"	8.5 in x 14.0 in (216 mm x 356 mm)
"ledger"	17.0 in x 11.0 in (432 mm x 279 mm)
"17x11"	17.0 in x 11.0 in (432 mm x 279 mm)
"letter"	8.5 in x 11.0 in (216 mm x 279 mm)
"tabloid"	11.0 in x 17.0 in (279 mm x 432 mm)
"11x17"	11.0 in x 17.0 in (279 mm x 432 mm)

Sizes by IEEE Printer Working Group, for children's writing

"government-letter"	8.0 in x 10.5 in (203 mm x 267 mm)
"government-legal"	8.5 in x 13.0 in (216 mm x 330 mm)
"philippine-legal"	8.5 in x 13.0 in (216 mm x 330 mm)

ANSI sizes

"ansi a"	8.5 in x 11.0 in (216 mm x 279 mm)
"ansi b"	11.0 in x 17.0 in (279 mm x 432 mm)
"ansi c"	17.0 in x 22.0 in (432 mm x 559 mm)
"ansi d"	22.0 in x 34.0 in (559 mm x 864 mm)
"ansi e"	34.0 in x 44.0 in (864 mm x 1118 mm)
"engineering f"	28.0 in x 40.0 in (711 mm x 1016 mm)

North American architectural sizes

"arch a"	9.0 in x 12.0 in (229 mm x 305 mm)
"arch b"	12.0 in x 18.0 in (305 mm x 457 mm)
"arch c"	18.0 in x 24.0 in (457 mm x 610 mm)
"arch d"	24.0 in x 36.0 in (610 mm x 914 mm)
"arch e"	36.0 in x 48.0 in (914 mm x 1219 mm)
"arch e1"	30.0 in x 42.0 in (762 mm x 1067 mm)

Other sizes, including antique sizes still used in the United Kingdom

"statement"	5.5 in x 8.5 in (140 mm x 216 mm)
"half letter"	5.5 in x 8.5 in (140 mm x 216 mm)
"quarto"	8.0 in x 10.0 in (203 mm x 254 mm)
"octavo"	6.75 in x 10.5 in (171 mm x 267 mm)
"executive"	7.25 in x 10.5 in (184 mm x 267 mm)
"monarch"	7.25 in x 10.5 in (184 mm x 267 mm)
"foolscap"	8.27 in x 13.0 in (210 mm x 330 mm)
"folio"	8.27 in x 13.0 in (210 mm x 330 mm)
"super-b"	13.0 in x 19.0 in (330 mm x 483 mm)
"post"	15.5 in x 19.5 in (394 mm x 495 mm)

"crown"	15.0 in x 20.0 in (381 mm x 508 mm)
"large post"	16.5 in x 21.0 in (419 mm x 533 mm)
"demy"	17.5 in x 22.5 in (445 mm x 572 mm)
"medium"	18.0 in x 23.0 in (457 mm x 584 mm)
"broadsheet"	18.0 in x 24.0 in (457 mm x 610 mm)
"royal"	20.0 in x 25.0 in (508 mm x 635 mm)
"elephant"	23.0 in x 28.0 in (584 mm x 711 mm)
"double demy"	22.5 in x 35.0 in (572 mm x 889 mm)
"quad demy"	35.0 in x 45.0 in (889 mm x 1143 mm)
"atlas"	26.0 in x 34.0 in (660 mm x 864 mm)
"imperial"	22.0 in x 30.0 in (559 mm x 762 mm)
"antiquarian"	31.0 in x 53.0 in (787 mm x 1346 mm)

PA4-based sizes

"pa10"	26 mm x 35 mm (1.02 in x 1.38 in)
"pa9"	35 mm x 52 mm (1.38 in x 2.05 in)
"pa8"	52 mm x 70 mm (2.05 in x 2.76 in)
"pa7"	70 mm x 105 mm (2.76 in x 4.13 in)
"pa6"	105 mm x 140 mm (4.13 in x 5.51 in)
"pa5"	140 mm x 210 mm (5.51 in x 8.27 in)
"pa4"	210 mm x 280 mm (8.27 in x 11.02 in)
"pa3"	280 mm x 420 mm (11.02 in x 16.54 in)
"pa2"	420 mm x 560 mm (16.54 in x 22.05 in)
"pa1"	560 mm x 840 mm (22.05 in x 33.07 in)
"pa0"	840 mm x 1120 mm (33.07 in x 44.09 in)

Additional format for use in Southeast Asia and Australia

"f4"	210 mm x 330 mm (8.27 in x 12.99 in)
------	--------------------------------------

A.6 Strumenti MIDI

Di seguito un elenco dei nomi che possono essere usati per la proprietà `midiInstrument`. Ciascuno viene presentato col suo numero preso dai 128 “numeri di programma” dello standard General MIDI.

1 acoustic grand	2 bright acoustic	3 electric grand
4 honky-tonk	5 electric piano 1	6 electric piano 2
7 harpsichord	8 clav	9 celesta
10 glockenspiel	11 music box	12 vibraphone
13 marimba	14 xylophone	15 tubular bells
16 dulcimer	17 drawbar organ	18 percussive organ
19 rock organ	20 church organ	21 reed organ
22 accordion	23 harmonica	24 concertina
25 acoustic guitar (nylon)	26 acoustic guitar (steel)	27 electric guitar (jazz)
28 electric guitar (clean)	29 electric guitar (muted)	30 overdriven guitar
31 distorted guitar	32 guitar harmonics	33 acoustic bass
34 electric bass (finger)	35 electric bass (pick)	36 fretless bass
37 slap bass 1	38 slap bass 2	39 synth bass 1
40 synth bass 2	41 violin	42 viola
43 cello	44 contrabass	45 tremolo strings
46 pizzicato strings	47 orchestral harp	48 timpani
49 string ensemble 1	50 string ensemble 2	51 synthstrings 1
52 synthstrings 2	53 choir aahs	54 voice oohs
55 synth voice	56 orchestra hit	57 trumpet
58 trombone	59 tuba	60 muted trumpet
61 french horn	62 brass section	63 synthbrass 1
64 synthbrass 2	65 soprano sax	66 alto sax
67 tenor sax	68 baritone sax	69 oboe
70 english horn	71 bassoon	72 clarinet

73	piccolo	74	flute	75	recorder
76	pan flute	77	blown bottle	78	shakuhachi
79	whistle	80	ocarina	81	lead 1 (square)
82	lead 2 (sawtooth)	83	lead 3 (calliope)	84	lead 4 (chiff)
85	lead 5 (charang)	86	lead 6 (voice)	87	lead 7 (fifths)
88	lead 8 (bass+lead)	89	pad 1 (new age)	90	pad 2 (warm)
91	pad 3 (polysynth)	92	pad 4 (choir)	93	pad 5 (bowed)
94	pad 6 (metallic)	95	pad 7 (halo)	96	pad 8 (sweep)
97	fx 1 (rain)	98	fx 2 (soundtrack)	99	fx 3 (crystal)
100	fx 4 (atmosphere)	101	fx 5 (brightness)	102	fx 6 (goblins)
103	fx 7 (echoes)	104	fx 8 (sci-fi)	105	sitar
106	banjo	107	shamisen	108	koto
109	kalimba	110	bagpipe	111	fiddle
112	shantai	113	tinkle bell	114	agogo
115	steel drums	116	woodblock	117	taiko drum
118	melodic tom	119	synth drum	120	reverse cymbal
121	guitar fret noise	122	breath noise	123	seashore
124	bird tweet	125	telephone ring	126	helicopter
127	applause	128	gunshot		

Sono disponibili anche le batterie elencate sotto.

1	standard kit	9	room kit	17	power kit
25	electronic kit	26	tr-808 kit	33	jazz kit
41	brush kit	49	orchestra kit	57	sfx kit
128	mt-32 kit/cm-64 kit				

A.7 Elenco dei colori

Colori normali

La sintassi è descritta in [Colorare gli oggetti], pagina 241.

black	white	red	green
blue	cyan	magenta	yellow
grey	darkred	darkgreen	darkblue
darkcyan	darkmagenta	darkyellow	

CSS color names

CSS color names may be used as-is in string arguments.

aliceblue	darkturquoise	lightsalmon	papayawhip
antiquewhite	darkviolet	lightseagreen	peachpuff
aqua	deeppink	lightskyblue	peru
aquamarine	deepskyblue	lightslategray	pink
azure	dimgray	lightslategrey	plum
beige	dimgrey	lightsteelblue	powderblue
bisque	dodgerblue	lightyellow	purple
black	firebrick	lime	rebeccapurple
blanchedalmond	floralwhite	limegreen	red
blue	forestgreen	linen	rosybrown
blueviolet	fuchsia	magenta	royalblue
brown	gainsboro	maroon	saddlebrown
burlywood	ghostwhite	mediumaquamarine	salmon
cadetblue	gold	mediumblue	sandybrown
chartreuse	goldenrod	mediumorchid	seagreen
chocolate	gray	mediumpurple	seashell
coral	green	mediumseagreen	sienna
cornflowerblue	greenyellow	mediumslateblue	silver
cornsilk	grey	mediumspringgreen	skyblue

crimson	honeydew	mediumturquoise	slateblue
cyan	hotpink	mediumvioletred	slategray
darkblue	indianred	midnightblue	slategrey
darkcyan	indigo	mintcream	snow
darkgoldenrod	ivory	mistyrose	springgreen
darkgray	khaki	moccasin	steelblue
darkgreen	lavender	navajowhite	tan
darkgrey	lavenderblush	navy	teal
darkkhaki	lawngreen	oldlace	thistle
darkmagenta	lemonchiffon	olive	tomato
darkolivegreen	lightblue	olivedrab	turquoise
darkorange	lightcoral	orange	violet
darkorchid	lightcyan	orangered	wheat
darkred	lightgoldenrodyellow	orchid	white
darksalmon	lightgray	palegoldenrod	whitesmoke
darkseagreen	lightgreen	palegreen	yellow
darkslateblue	lightgrey	paleturquoise	yellowgreen
darkslategray	lightpink	palevioletred	
darkslategrey			

Le definizioni di colore CSS differiscono dai nomi di colore X per i seguenti colori: green, grey, maroon, purple.

Nomi di colore X

I nomi di colore X11 (https://en.wikipedia.org/wiki/X11_color_names) offrono una scelta più ampia dei nomi CSS. Hanno diverse varianti:

- Qualsiasi nome scritto come un'unica parola con lettere iniziali maiuscole (es: 'LightSlateBlue') può essere scritto anche con parole separate da spazio e con o senza maiuscola (es: 'light slate blue').
- La parola 'grey' può essere sempre scritta come 'gray' (es: 'DarkSlateGray'), senza alcuna differenza nell'output.
- Alcuni nomi possono avere un suffisso numerico (es: 'LightSalmon4').

Usare la funzione Scheme `x11-color` per aver accesso ad essi.

Le tabelle seguenti presentano tutti i nomi di colore che possono essere usati senza un suffisso numerico, e poi il sottoinsieme di questi che possono essere usati con tale suffisso.

Nomi di colori senza un suffisso numerico

AliceBlue	LawnGreen	OrangeRed	firebrick
AntiqueWhite	LemonChiffon	PaleGoldenrod	gainsboro
BlanchedAlmond	LightBlue	PaleGreen	gold
BlueViolet	LightCoral	PaleTurquoise	goldenrod
CadetBlue	LightCyan	PaleVioletRed	green
CornflowerBlue	LightGoldenrod	PapayaWhip	grey
DarkBlue	LightGoldenrodYellow	PeachPuff	honeydew
DarkCyan	LightGreen	PowderBlue	ivory
DarkGoldenrod	LightGrey	RosyBrown	khaki
DarkGreen	LightPink	RoyalBlue	lavender
DarkGrey	LightSalmon	SaddleBrown	linen
DarkKhaki	LightSeaGreen	SandyBrown	magenta
DarkMagenta	LightSkyBlue	SeaGreen	maroon
DarkOliveGreen	LightSlateBlue	SkyBlue	moccasin

DarkOrange	LightSlateGrey	SlateBlue	navy
DarkOrchid	LightSteelBlue	SlateGrey	orange
DarkRed	LightYellow	SpringGreen	orchid
DarkSalmon	LimeGreen	SteelBlue	peru
DarkSeaGreen	MediumAquamarine	VioletRed	pink
DarkSlateBlue	MediumBlue	WhiteSmoke	plum
DarkSlateGrey	MediumOrchid	YellowGreen	purple
DarkTurquoise	MediumPurple	aquamarine	red
DarkViolet	MediumSeaGreen	azure	salmon
DeepPink	MediumSlateBlue	beige	seashell
DeepSkyBlue	MediumSpringGreen	bisque	sienna
DimGrey	MediumTurquoise	black	snow
DodgerBlue	MediumVioletRed	blue	tan
FloralWhite	MidnightBlue	brown	thistle
ForestGreen	MintCream	burlywood	tomato
GhostWhite	MistyRose	chartreuse	turquoise
GreenYellow	NavajoWhite	chocolate	violet
HotPink	NavyBlue	coral	wheat
IndianRed	OldLace	cornsilk	white
LavenderBlush	OliveDrab	cyan	yellow

Nomi di colori con un suffisso numerico

Nei seguenti nomi il suffisso *N* può essere un numero intero compreso tra 1 e 4, dalle tonalità più chiare a quelle più scure:

AntiqueWhite <i>N</i>	LightSkyBlue <i>N</i>	SteelBlue <i>N</i>	khaki <i>N</i>
CadetBlue <i>N</i>	LightSteelBlue <i>N</i>	VioletRed <i>N</i>	magenta <i>N</i>
DarkGoldenrod <i>N</i>	LightYellow <i>N</i>	aquamarine <i>N</i>	maroon <i>N</i>
DarkOliveGreen <i>N</i>	MediumOrchid <i>N</i>	azure <i>N</i>	orange <i>N</i>
DarkOrange <i>N</i>	MediumPurple <i>N</i>	bisque <i>N</i>	orchid <i>N</i>
DarkOrchid <i>N</i>	MistyRose <i>N</i>	blue <i>N</i>	pink <i>N</i>
DarkSeaGreen <i>N</i>	NavajoWhite <i>N</i>	brown <i>N</i>	plum <i>N</i>
DeepPink <i>N</i>	OliveDrab <i>N</i>	burlywood <i>N</i>	purple <i>N</i>
DeepSkyBlue <i>N</i>	OrangeRed <i>N</i>	chartreuse <i>N</i>	red <i>N</i>
DodgerBlue <i>N</i>	PaleGreen <i>N</i>	chocolate <i>N</i>	salmon <i>N</i>
HotPink <i>N</i>	PaleTurquoise <i>N</i>	coral <i>N</i>	seashell <i>N</i>
IndianRed <i>N</i>	PaleVioletRed <i>N</i>	cornsilk <i>N</i>	sienna <i>N</i>
LavenderBlush <i>N</i>	PeachPuff <i>N</i>	cyan <i>N</i>	snow <i>N</i>
LemonChiffon <i>N</i>	RosyBrown <i>N</i>	firebrick <i>N</i>	tan <i>N</i>
LightBlue <i>N</i>	RoyalBlue <i>N</i>	gold <i>N</i>	thistle <i>N</i>
LightCyan <i>N</i>	SeaGreen <i>N</i>	goldenrod <i>N</i>	tomato <i>N</i>
LightGoldenrod <i>N</i>	SkyBlue <i>N</i>	green <i>N</i>	turquoise <i>N</i>
LightPink <i>N</i>	SlateBlue <i>N</i>	honeydew <i>N</i>	wheat <i>N</i>
LightSalmon <i>N</i>	SpringGreen <i>N</i>	ivory <i>N</i>	yellow <i>N</i>

Scala di grigi

Una scala di grigi si ottiene con:

grey*N*

dove *N* è un numero compreso tra 0 e 100.

Color-blind-safe colors

La funzione `Scheme universal-color` fornisce un insieme di otto colori (<https://jfly.uni-koeln.de/color>) progettati per essere non ambigui per le persone affette da bicromatismo.

black	orange	skyblue	bluegreen
yellow	blue	vermillion	redpurple

A.8 Il font Emmentaler

Il font Emmentaler è composto da due *sottoinsiemi* di glifi. “Feta”, usato nella notazione classica e “Parmesan”, usato nella notazione antica.

A qualsiasi glifo del font Emmentaler si può accedere direttamente usando del testo insieme al nome del glifo (come è mostrato nelle tabelle seguenti). Per esempio:



















```
g~\markup {\musicglyph "scripts.segno" }
```

oppure

```
\markup {\musicglyph "five"}.
```

Maggiori informazioni in Sezione 1.8.2 [Formattazione del testo], pagina 260.

Glifi della chiave

<code>clefs.C</code>		<code>clefs.C_change</code>	
<code>clefs.varC</code>		<code>clefs.varC_change</code>	
<code>clefs.F</code>		<code>clefs.F_change</code>	
<code>clefs.G</code>		<code>clefs.G_change</code>	
<code>clefs.GG</code>		<code>clefs.GG_change</code>	
<code>clefs.tenorG</code>		<code>clefs.tenorG_change</code>	
<code>clefs.percussion</code>		<code>clefs.percussion_change</code>	
<code>clefs.varpercussion</code>		<code>clefs</code> <code>.varpercussion_change</code>	
<code>clefs.tab</code>		<code>clefs.tab_change</code>	

Glifi delle indicazioni di tempo

<code>timesig.C44</code>		<code>timesig.C22</code>	
--------------------------	---	--------------------------	---

Glifi dei numeri

<code>plus</code>	+	<code>comma</code>	,
<code>hyphen</code>	-	<code>period</code>	.

figuredash	—	endash	—
parenleft	(parenright)
slash	/	zero	0
one	1	two	2
three	3	four	4
four.alt	4	five	5
six	6	seven	7
seven.alt	7	eight	8
nine	9	fixedwidth.zero	0
fixedwidth.one	1	fixedwidth.two	2
fixedwidth.three	3	fixedwidth.four	4
fixedwidth.four.alt	4	fixedwidth.five	5
fixedwidth.six	6	fixedwidth.seven	7
fixedwidth.seven.alt	7	fixedwidth.eight	8
fixedwidth.nine	9	fattened.zero	0
fattened.one	1	fattened.two	2
fattened.three	3	fattened.four	4
fattened.four.alt	4	fattened.five	5
fattened.six	6	fattened.seven	7
fattened.seven.alt	7	fattened.eight	8
fattened.nine	9	fattened.fixedwidth.zero	0
fattened.fixedwidth.one	1	fattened.fixedwidth.two	2
fattened	3	fattened.fixedwidth.four	4
.fixedwidth.three	4	fattened.fixedwidth.five	5
fattened.fixedwidth	4	fattened	7
.four.alt	6	.fixedwidth.seven	8
fattened.fixedwidth.six	7	fattened	8
fattened.fixedwidth	9	.fixedwidth.eight	
.seven.alt		u2007	
fattened.fixedwidth.nine			

u2009

u200A

Glifi del simbolo di basso numerato


figbass.twoplus	2+	figbass.fourplus	4+
figbass.fiveplus	5+	figbass.sixstroked	6
figbass.sevenstroked	7	figbass.ninestroked	9

Glifi delle alterazioni

accidentals.sharp	#	accidentals	#
accidentals	#	.sharp.figbass	#
.sharp.arrowup	#	accidentals	#
accidentals	#	.sharp.arrowdown	#
.sharp.arrowboth	#	accidentals.sharp	#
accidentals.sharp	#	.slashslash.stem	#
.slashslashslash.stemstem	#	accidentals.sharp	#
accidentals	#	.slashslashslash.stem	#
.sharp.slash.stem	#	accidentals.sharp	#
accidentals.doublesharp	#	.slashslash.stemstemstem	#
accidentals.natural	b	accidentals	#
accidentals	b	.doublesharp.figbass	#
.natural.arrowup	b	accidentals	b
accidentals	b	.natural.figbass	b
.natural.arrowboth	b	accidentals	b
accidentals.flat.figbass	b	.natural.arrowdown	b
accidentals	b	accidentals.flat	b
.flat.arrowdown	b	accidentals.flat.arrowup	b
accidentals.flat.slash	b	accidentals	b
accidentals	b	.flat.arrowboth	b
.mirroredflat.flat	b	accidentals.flat	b
accidentals	b	.slashslash	b
.mirroredflat.backslash	b	accidentals.mirroredflat	b
accidentals	b	accidentals.flatflat	b
.flatflat.figbass	b	accidentals	b
accidentals.sharp.sori	#	.flatflat.slash	b
accidentals.rightparen)	accidentals.flat.koron	p
		accidentals.leftparen	(

Glifi delle teste di nota predefinite

noteheads.uM2	o	noteheads.dM2	o
noteheads.sM1	o	noteheads.s0	o

<code>noteheads.s1</code>		<code>noteheads.s2</code>	
---------------------------	---	---------------------------	---

Glifi delle teste di nota speciali

<code>noteheads.sM1double</code>		<code>noteheads.s0diamond</code>	
<code>noteheads.s1diamond</code>		<code>noteheads.s2diamond</code>	
<code>noteheads.s0triangle</code>		<code>noteheads.s1triangle</code>	
<code>noteheads.s2triangle</code>		<code>noteheads.s0slash</code>	
<code>noteheads.s1slash</code>		<code>noteheads.s2slash</code>	
<code>noteheads.s0cross</code>		<code>noteheads.s1cross</code>	
<code>noteheads.s2cross</code>		<code>noteheads.s2xcircle</code>	
<code>noteheads.d0arrow</code>		<code>noteheads.u0arrow</code>	
<code>noteheads.d2arrow</code>		<code>noteheads.u2arrow</code>	
<code>noteheads.s0harmonic</code>		<code>noteheads.s2harmonic</code>	

Glifi delle teste di nota a forma variabile

<code>noteheads.s0do</code>		<code>noteheads.s1do</code>	
<code>noteheads.s2do</code>		<code>noteheads.s0doThin</code>	
<code>noteheads.s1doThin</code>		<code>noteheads.s2doThin</code>	
<code>noteheads.s0re</code>		<code>noteheads.s1re</code>	
<code>noteheads.s2re</code>		<code>noteheads.s0reThin</code>	
<code>noteheads.s1reThin</code>		<code>noteheads.s2reThin</code>	
<code>noteheads.s0mi</code>		<code>noteheads.s1mi</code>	
<code>noteheads.s2mi</code>		<code>noteheads.s0miMirror</code>	
<code>noteheads.s1miMirror</code>		<code>noteheads.s2miMirror</code>	
<code>noteheads.s0miThin</code>		<code>noteheads.s1miThin</code>	
<code>noteheads.s2miThin</code>		<code>noteheads.u0fa</code>	
<code>noteheads.d0fa</code>		<code>noteheads.u1fa</code>	
<code>noteheads.d1fa</code>		<code>noteheads.u2fa</code>	











noteheads.d2fa	▶	noteheads.u0faThin	◀
noteheads.d0faThin	◀	noteheads.u1faThin	◀
noteheads.d1faThin	◀	noteheads.u2faThin	▶
noteheads.d2faThin	▶	noteheads.s0sol	○
noteheads.s1sol	○	noteheads.s2sol	●
noteheads.s0la	□	noteheads.s1la	□
noteheads.s2la	■	noteheads.s0laThin	□
noteheads.s1laThin	□	noteheads.s2laThin	■
noteheads.s0ti	◇	noteheads.s1ti	◇
noteheads.s2ti	◆	noteheads.s0tiThin	◇
noteheads.s1tiThin	◇	noteheads.s2tiThin	◆
noteheads.u0doFunk	▷	noteheads.d0doFunk	◁
noteheads.u1doFunk	▷	noteheads.d1doFunk	◁
noteheads.u2doFunk	▷	noteheads.d2doFunk	◁
noteheads.u0reFunk	▶	noteheads.d0reFunk	◀
noteheads.u1reFunk	▶	noteheads.d1reFunk	◀
noteheads.u2reFunk	▶	noteheads.d2reFunk	◀
noteheads.u0miFunk	◇	noteheads.d0miFunk	◇
noteheads.u1miFunk	◇	noteheads.d1miFunk	◇
noteheads.s2miFunk	◆	noteheads.u0faFunk	◀
noteheads.d0faFunk	▶	noteheads.u1faFunk	◀
noteheads.d1faFunk	▶	noteheads.u2faFunk	▶
noteheads.d2faFunk	▶	noteheads.s0solFunk	○
noteheads.s1solFunk	○	noteheads.s2solFunk	●
noteheads.s0laFunk	□	noteheads.s1laFunk	□
noteheads.s2laFunk	■	noteheads.u0tiFunk	◇
noteheads.d0tiFunk	◇	noteheads.ultiFunk	◇

noteheads.d1tiFunk	◁	noteheads.u2tiFunk	▷
noteheads.d2tiFunk	◀	noteheads.s0doWalker	Δ
noteheads.u1doWalker	▽	noteheads.d1doWalker	Δ
noteheads.u2doWalker	▼	noteheads.d2doWalker	▲
noteheads.s0reWalker	◁	noteheads.u1reWalker	▷
noteheads.d1reWalker	◁	noteheads.u2reWalker	▷
noteheads.d2reWalker	◀	noteheads.s0miWalker	◇
noteheads.s1miWalker	◇	noteheads.s2miWalker	◆
noteheads.s0faWalker	▷	noteheads.u1faWalker	▷
noteheads.d1faWalker	▷	noteheads.u2faWalker	▷
noteheads.d2faWalker	▶	noteheads.s0laWalker	□
noteheads.s1laWalker	□	noteheads.s2laWalker	■
noteheads.s0tiWalker	▷	noteheads.ultiWalker	▷
noteheads.d1tiWalker	▷	noteheads.u2tiWalker	▷
noteheads.d2tiWalker	▶		

Glifi delle pause

rests.0	—	rests.1	—
rests.0o	—	rests.1o	—
rests.M3		rests.M2	
rests.M1	■	rests.M1o	■
rests.2	~	rests.2classical	~
rests.2z	~	rests.3	~
rests.4	~	rests.5	~
rests.6	~	rests.7	~
rests.8	~	rests.9	~
rests.10	~		

Glifi delle code

flags.u3		flags.u4	
flags.u5		flags.u6	
flags.u7		flags.u8	
flags.u9		flags.u10	
flags.d3		flags.d4	
flags.d5		flags.d6	
flags.d7		flags.d8	
flags.d9		flags.d10	
flags.stackedu3		flags.stackedu4	
flags.stackedu5		flags.stackedu6	
flags.stackedu7		flags.stackedu8	
flags.stackedu9		flags.stackedu10	
flags.stackedd3		flags.stackedd4	
flags.stackedd5		flags.stackedd6	
flags.stackedd7		flags.stackedd8	
flags.stackedd9		flags.stackedd10	
flags.ugrace		flags.dgrace	

Glifi dei punti


`dots.dot` 

Glifi delle dinamiche

<code>space</code>		<code>f</code>	<i>f</i>
<code>m</code>	<i>m</i>	<code>n</code>	<i>n</i>
<code>p</code>	<i>p</i>	<code>r</code>	<i>r</i>
<code>s</code>	<i>s</i>	<code>z</code>	<i>z</i>





Glifi dei segni

<code>scripts.ufermata</code>		<code>scripts.dfermata</code>	
<code>scripts</code> <code>.uhenzeshortfermata</code>		<code>scripts</code> <code>.dhenzeshortfermata</code>	
<code>scripts</code> <code>.uhenzelongfermata</code>		<code>scripts</code> <code>.dhenzelongfermata</code>	
<code>scripts.ushortfermata</code>		<code>scripts.dshortfermata</code>	
<code>scripts</code> <code>.uveryshortfermata</code>		<code>scripts</code> <code>.dveryshortfermata</code>	
<code>scripts.ulongfermata</code>		<code>scripts.dlongfermata</code>	
<code>scripts.uverylongfermata</code>		<code>scripts.dverylongfermata</code>	
<code>scripts.thumb</code>		<code>scripts.sforzato</code>	
<code>scripts.espr</code>		<code>scripts.staccato</code>	
<code>scripts.ustaccatissimo</code>		<code>scripts.dstaccatissimo</code>	
<code>scripts.tenuto</code>		<code>scripts.uportato</code>	
<code>scripts.dportato</code>		<code>scripts.umarcato</code>	
<code>scripts.dmarcato</code>		<code>scripts.open</code>	
<code>scripts.halfopen</code>		<code>scripts.halfopenvertical</code>	
<code>scripts.stopped</code>		<code>scripts.uupbow</code>	
<code>scripts.dupbow</code>		<code>scripts.udownbow</code>	
<code>scripts.ddownbow</code>		<code>scripts.reverseturn</code>	
<code>scripts.turn</code>		<code>scripts.slashturn</code>	

<code>scripts.haydnturn</code>		<code>scripts.trill</code>	
<code>scripts.upedalheel</code>	U	<code>scripts.dpedalheel</code>	∩
<code>scripts.upedaltoe</code>	V	<code>scripts.dpedaltoe</code>	^
<code>scripts.pedalheelcircle</code>	O	<code>scripts.flageolet</code>	o
<code>scripts.segno</code>		<code>scripts.varsegno</code>	
<code>scripts.coda</code>		<code>scripts.varcoda</code>	
<code>scripts.rcomma</code>	,	<code>scripts.lcomma</code>	‘
<code>scripts.rvarcomma</code>	/	<code>scripts.lvarcomma</code>	/
<code>scripts.raltcomma</code>	,	<code>scripts.laltcomma</code>	‘
<code>scripts.arpeggio</code>		<code>scripts.trill_element</code>	~
<code>scripts.arpeggio</code> <code>.arrow.M1</code>		<code>scripts.arpeggio.arrow.1</code>	
<code>scripts.prall</code>		<code>scripts.mordent</code>	
<code>scripts.prallprall</code>		<code>scripts.prallmordent</code>	
<code>scripts.upprall</code>		<code>scripts.upmordent</code>	
<code>scripts.prallup</code>		<code>scripts.downprall</code>	
<code>scripts.downmordent</code>		<code>scripts.pralldown</code>	
<code>scripts.lineprall</code>		<code>scripts.bachschleifer</code>	
<code>scripts.caesura.curved</code>	//	<code>scripts.caesura.straight</code>	//
<code>scripts.tickmark</code>	✓	<code>scripts.snappizzicato</code>	♯
<code>scripts.ictus</code>	,	<code>scripts.uaccentus</code>	,
<code>scripts.daccentus</code>	,	<code>scripts.usemicirculus</code>	.
<code>scripts.dsemicirculus</code>	.	<code>scripts.circulus</code>	o
<code>scripts</code> <code>.usignumcongruentiae</code>	§	<code>scripts</code> <code>.dsignumcongruentiae</code>	§

Glifi delle teste a forma di freccia








<code>arrowheads.open.01</code>	>	<code>arrowheads.open.0M1</code>	<
<code>arrowheads.open.11</code>	^	<code>arrowheads.open.1M1</code>	Y

<code>arrowheads.close.01</code>		<code>arrowheads.close.0M1</code>	
<code>arrowheads.close.11</code>		<code>arrowheads.close.1M1</code>	









Glifi delle estremità delle parentesi

<code>brackettips.up</code>		<code>brackettips.down</code>	
-----------------------------	---	-------------------------------	---

Glifi dei pedali

<code>pedal.*</code>		<code>pedal.M</code>	
<code>pedal..</code>		<code>pedal.P</code>	
<code>pedal.d</code>		<code>pedal.e</code>	
<code>pedal.Ped</code>			


















Glifi della fisarmonica

<code>accordion.discant</code>		<code>accordion.dot</code>	
<code>accordion.freebass</code>		<code>accordion.stdbass</code>	
<code>accordion.bayanbass</code>		<code>accordion.oldEE</code>	
<code>accordion.push</code>		<code>accordion.pull</code>	

Glifi delle legature di valore

<code>ties.lyric.short</code>		<code>ties.lyric.default</code>	
-------------------------------	---	---------------------------------	---

Glifi della notazione vaticana

<code>clefs.vaticana.do</code>		<code>clefs.vaticana.do_change</code>	
<code>clefs.vaticana.fa</code>		<code>clefs.vaticana.fa_change</code>	
<code>custodes.vaticana.u0</code>		<code>custodes.vaticana.u1</code>	
<code>custodes.vaticana.u2</code>		<code>custodes.vaticana.d0</code>	
<code>custodes.vaticana.d1</code>		<code>custodes.vaticana.d2</code>	
<code>accidentals.vaticanaM1</code>		<code>accidentals.vaticana0</code>	
<code>dots.dotvaticana</code>		<code>noteheads</code>	
<code>noteheads.svaticana</code>		<code>noteheads.svaticana</code>	
<code>.punctum.cavum</code>		<code>.punctum</code>	

noteheads.svaticana	◻	noteheads.svaticana	◊
.linea.punctum.cavum		.inclinatum	
noteheads.svaticana.lpes	■	noteheads	■
		.svaticana.vlpes	
noteheads.svaticana.upes	■	noteheads	■
		.svaticana.vupes	
noteheads	.	noteheads	.
.svaticana.plica		.svaticana.vplica	
noteheads	┐	noteheads.svaticana	┐
.svaticana.epiphonus		.vepiphonus	
noteheads.svaticana	.	noteheads.svaticana	.
.reverse.plica		.reverse.vplica	
noteheads.svaticana	┐	noteheads.svaticana	┐
.inner.cephalicus		.cephalicus	
noteheads	.		
.svaticana.quilisma			















































Glifi della notazione medicaea





















































clefs.medicaea.do	⌋	clefs.medicaea.do_change	⌋
clefs.medicaea.fa	⌋	clefs.medicaea.fa_change	⌋
custodes.medicaea.u0		custodes.medicaea.u1	
custodes.medicaea.u2		custodes.medicaea.d0	
custodes.medicaea.d1		custodes.medicaea.d2	
accidentals.medicaeaM1	♭	noteheads.smedicaea	◊
		.inclinatum	
noteheads	■	noteheads	■
.smedicaea.punctum		.smedicaea.rvirga	
noteheads	┐		
.smedicaea.virga			

Glifi Hufnagel

clefs.hufnagel.do	⌋	clefs.hufnagel.do_change	⌋
clefs.hufnagel.fa	⌋	clefs.hufnagel.fa_change	⌋
clefs.hufnagel.do.fa	⌋	clefs.hufnagel	⌋
	⌋	.do.fa_change	⌋
custodes.hufnagel.u0	✓	custodes.hufnagel.u1	✓
custodes.hufnagel.u2	✓	custodes.hufnagel.d0	↗
custodes.hufnagel.d1	↗	custodes.hufnagel.d2	↗
accidentals.hufnagelM1	♭	noteheads	◊
		.shufnagel.punctum	
noteheads	↑	noteheads.shufnagel.lpes	▬
.shufnagel.virga			

Glifi della notazione mensurale

rests.M3mensural		rests.M2mensural	
rests.M1mensural		rests.0mensural	
rests.1mensural		rests.2mensural	
rests.3mensural		rests.4mensural	
clefs.mensural.c		clefs.mensural.c_change	
clefs.blackmensural.c		clefs.blackmensural.c_change	
clefs.mensural.f		clefs.mensural.f_change	
clefs.mensural.g		clefs.mensural.g_change	
custodes.mensural.u0		custodes.mensural.u1	
custodes.mensural.u2		custodes.mensural.d0	
custodes.mensural.d1		custodes.mensural.d2	
accidentals.mensural1		accidentals.mensuralM1	
flags.mensuralu03		flags.mensuralu13	
flags.mensuralu23		flags.mensurald03	
flags.mensurald13		flags.mensurald23	
flags.mensuralu04		flags.mensuralu14	
flags.mensuralu24		flags.mensurald04	
flags.mensurald14		flags.mensurald24	
flags.mensuralu05		flags.mensuralu15	
flags.mensuralu25		flags.mensurald05	
flags.mensurald15		flags.mensurald25	
flags.mensuralu06		flags.mensuralu16	
flags.mensuralu26		flags.mensurald06	

flags.mensurald16		flags.mensurald26	
timesig.mensural44		timesig.mensural22	
timesig.mensural32		timesig.mensural64	
timesig.mensural94		timesig.mensural34	
timesig.mensural68		timesig.mensural98	
timesig.mensural48		timesig.mensural68alt	
timesig.mensural24		noteheads.uM3mensural	
noteheads.dM3mensural		noteheads.sM3ligmensural	
noteheads.uM2mensural		noteheads.dM2mensural	
noteheads.sM2ligmensural		noteheads.sM1mensural	
noteheads.urM3mensural		noteheads.drM3mensural	
noteheads .srM3ligmensural		noteheads.urM2mensural	
noteheads.drM2mensural		noteheads .srM2ligmensural	
noteheads.srM1mensural		noteheads .uM3semimensural	
noteheads .dM3semimensural		noteheads .sM3semiligmensural	
noteheads .uM2semimensural		noteheads .dM2semimensural	
noteheads .sM2semiligmensural		noteheads .sM1semimensural	
noteheads .urM3semimensural		noteheads .drM3semimensural	
noteheads .srM3semiligmensural		noteheads .urM2semimensural	
noteheads .drM2semimensural		noteheads .srM2semiligmensural	
noteheads .srM1semimensural		noteheads .uM3blackmensural	
noteheads .dM3blackmensural		noteheads .sM3blackligmensural	
noteheads .uM2blackmensural		noteheads .dM2blackmensural	
noteheads .sM2blackligmensural		noteheads .sM1blackmensural	
noteheads.s0mensural		noteheads.s1mensural	
noteheads.s2mensural		noteheads .s0blackmensural	

Glifi della notazione neomensurale








rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural	┆	rests.0neomensural	.
rests.1neomensural	┆	rests.2neomensural	ʹ
rests.3neomensural	ʹ	rests.4neomensural	ʹ
clefs.neomensural.c		clefs.neomensural .c_change	
timesig.neomensural44	C	timesig.neomensural22	Ⓒ
timesig.neomensural32	O	timesig.neomensural64	Ⓒ
timesig.neomensural94	⊙	timesig.neomensural34	⊙
timesig.neomensural68	Ⓔ	timesig.neomensural98	⊙
timesig.neomensural48	⊙	timesig.neomensural68alt	⊙
timesig.neomensural24	⊙	noteheads.uM3neomensural	
noteheads.dM3neomensural		noteheads.uM2neomensural	
noteheads.dM2neomensural		noteheads.sM1neomensural	
noteheads .urM3neomensural		noteheads .drM3neomensural	
noteheads .urM2neomensural		noteheads .drM2neomensural	
noteheads .srM1neomensural		noteheads.s0neomensural	◊
noteheads.s1neomensural	◊	noteheads.s2neomensural	◊

Glifi Petrucci

clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	
clefs.petrucci.c5		clefs.petrucci.c5_change	
clefs.petrucci.f		clefs.petrucci.f_change	

<code>clefs.petrucci.g</code>		<code>clefs.petrucci.g_change</code>	
<code>noteheads.s0petrucci</code>		<code>noteheads.s1petrucci</code>	
<code>noteheads.s2petrucci</code>		<code>noteheads.s0blackpetrucci</code>	
<code>noteheads.s1blackpetrucci</code>		<code>noteheads.s2blackpetrucci</code>	

Glifi Solesmes

<code>noteheads.ssolesmes.incl.parvum</code>		<code>noteheads.ssolesmes.auct.asc</code>	
<code>noteheads.ssolesmes.auct.desc</code>		<code>noteheads.ssolesmes.incl.auctum</code>	
<code>noteheads.ssolesmes.stropha</code>		<code>noteheads.ssolesmes.stropha.aucta</code>	
<code>noteheads.ssolesmes.oriscus</code>			

Glifi della notazione di Kiev

<code>clefs.kievan.do</code>		<code>clefs.kievan.do_change</code>	
<code>accidentals.kievan1</code>		<code>accidentals.kievanM1</code>	
<code>scripts.barline.kievan</code>		<code>dots.dotkievan</code>	
<code>noteheads.sM2kievan</code>		<code>noteheads.sM1kievan</code>	
<code>noteheads.s0kievan</code>		<code>noteheads.d2kievan</code>	
<code>noteheads.u2kievan</code>		<code>noteheads.s1kievan</code>	
<code>noteheads.sr1kievan</code>		<code>noteheads.d3kievan</code>	
<code>noteheads.u3kievan</code>			

A.9 Stili delle teste di nota

Si possono usare i seguenti stili per le teste di nota.

default	mensural
altdefault	neomensural
baroque	petrucci
slash	diamond
arrow	harmonic
triangle	harmonic-black
cross	harmonic-mixed
xcircle	

A.10 Gruppi di glifi di alterazione

Sono disponibili le seguenti serie di glifi di alterazione.

standard-alteration-glyph-name-alist



alteration-hufnagel-glyph-name-alist



alteration-medicaea-glyph-name-alist



alteration-vaticana-glyph-name-alist



alteration-mensural-glyph-name-alist



alteration-kievan-glyph-name-alist





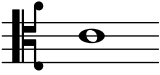
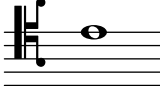

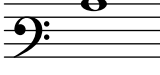

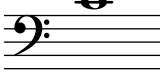



A.11 Stili della chiave

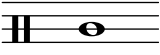
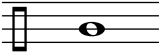
La seguente tabella mostra tutti gli stili di chiave possibili (inclusi quelli in cui la posizione del *Do centrale*) cambia a seconda della chiave).

Chiavi standard

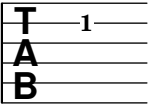
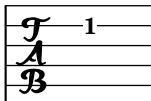
Esempio	Output	Esempio	Output
<code>\clef G</code>		<code>\clef "G2"</code>	
<code>\clef treble</code>		<code>\clef violin</code>	
<code>\clef french</code>		<code>\clef GG</code>	
<code>\clef tenorG</code>			
<code>\clef soprano</code>		<code>\clef mezzosoprano</code>	
<code>\clef C</code>		<code>\clef alto</code>	

<code>\clef tenor</code>		<code>\clef baritone</code>	
<code>\clef varC</code>		<code>\clef altovarC</code>	
<code>\clef tenorvarC</code>		<code>\clef baritonevarC</code>	
<code>\clef varbaritone</code>		<code>\clef baritonevarF</code>	
<code>\clef F</code>		<code>\clef bass</code>	
<code>\clef subbass</code>			

Chiave del rigo della percussione

Esempio	Output	Esempio	Output
<code>\clef percussion</code>		<code>\clef varpercussion</code>	

Chiavi del rigo dell'intavolatura

Esempio	Output	Esempio	Output
<code>\new TabStaff { \clef tab }</code>		<code>\new TabStaff { \clef moderntab }</code>	

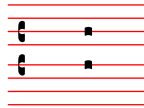
Chiavi della musica antica

Gregoriane

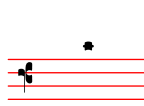
Esempio

Output

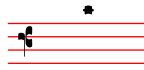
`\clef "vaticana-do1"`



`\clef "vaticana-do3"`



`\clef "vaticana-fa2"`



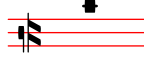
`\clef "medicaea-do1"`



`\clef "medicaea-do3"`



`\clef "medicaea-fa2"`



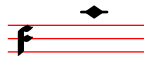
`\clef "hufnagel-do1"`



`\clef "hufnagel-do3"`

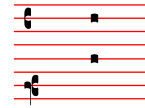


`\clef "hufnagel-fa2"`



Esempio

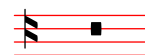
`\clef "vaticana-do2"`



`\clef "vaticana-fa1"`



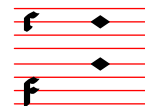
`\clef "medicaea-do2"`



`\clef "medicaea-fa1"`



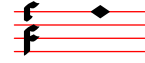
`\clef "hufnagel-do2"`



`\clef "hufnagel-fa1"`



`\clef "hufnagel-do-fa"`



Mensurali

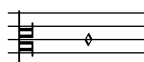
Esempio

Output

`\clef "mensural-c1"`



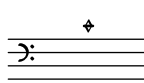
`\clef "mensural-c3"`



`\clef "mensural-c5"`



`\clef "mensural-f"`



`\clef "mensural-f3"`

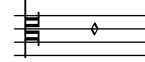


Esempio

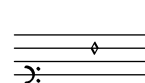
`\clef "mensural-c2"`



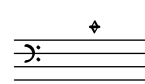
`\clef "mensural-c4"`

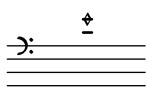



`\clef "mensural-f2"`




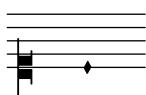
`\clef "mensural-f4"`



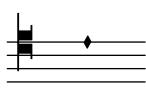
`\clef "mensural-f5"` 


`\clef "mensural-g1"` 


`\clef "mensural-g"` 


`\clef "blackmensural-c1"` 


`\clef "blackmensural-c3"` 

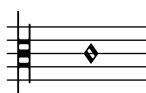
`\clef "blackmensural-c5"` 


`\clef "neomensural-c1"` 


`\clef "neomensural-c3"` 

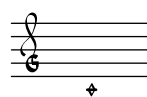
`\clef "neomensural-c5"` 


`\clef "petrucci-c1"` 


`\clef "petrucci-c3"` 


`\clef "petrucci-c5"` 

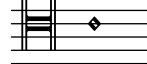
`\clef "petrucci-f"` 

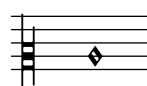
`\clef "mensural-g2"` 


`\clef "blackmensural-c2"` 


`\clef "blackmensural-c4"` 

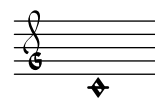
`\clef "neomensural-c2"` 

`\clef "neomensural-c4"` 

`\clef "petrucci-c2"` 

`\clef "petrucci-c4"` 

`\clef "petrucci-f2"` 

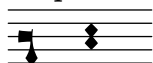
`\clef "petrucci-f3"``\clef "petrucci-f4"``\clef "petrucci-f5"``\clef "petrucci-g1"``\clef "petrucci-g2"``\clef "petrucci-g"`

Kievan

Esempio

`\clef "kievan-do"`

Output



A.12 Comandi per *markup*

Tutti i comandi seguenti possono essere usati all'interno di `\markup`.

The following commands can all be used inside `\markup { }`.

A.12.1 Font markup

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size (in points) to display *arg*.

This function adjusts the baseline-skip and word-space properties accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```

default text font size **text font size 16** text font size 12

Used properties:

- baseline-skip (3)
- word-space (0.6)

`\bold` *arg* (markup)

Print *arg* with a bold face.

```
\markup {
  default
  \hspace #2
```

```
\bold bold
}
```

default bold

The code `\markup \bold ...` is a shorthand for `\markup \override #'(font-series . bold) ...` – using the more verbose form, it is possible to obtain nuances such as semi-bold, if the text font has such variants. Refer to the documentation for the `font-series` properties (Sezione “User backend properties” in *Guida al Funzionamento Interno*).

`\box arg` (markup)

Draw a box around *arg*.

This function looks at the thickness, box-padding, and font-size properties to determine the line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box \line { V. S. }
}
```

V. S.

Note that the box does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

```
\markup {
  \override #'(box-padding . 1.5)
  \column {
    "text"
    \box "text"
    \left-align \box "text"
  }
}
```

text
text
text

Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps arg` (markup)

Print *arg* in (fake) small caps.

This function is a copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

Use `\fontCaps` for real small caps (if the font provides it).

`\dynamic arg` (markup)

Print *arg* using the (music) font for dynamics.

This font only contains letters **f**, **m**, **n**, **p**, **r**, **s**, and **z**. When producing phrases like ‘più **f**’, the normal words (like ‘più’) should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```

sfzp

`\figured-bass arg` (markup)

Set *arg* as small numbers for figured bass.

Specially slashed digits can be achieved with a trailing backslash (for numbers 6, 7, and 9) or a trailing plus (for numbers 2, 4, and 5).¹

The use of a backslash is in analogy to `\figuremode` (vedi [Inserimento del basso continuo], pagina 446). Note that to get a backslash character in markup it must be escaped by doubling it. Additionally, it must be put into double quotes.

```
\markup {
  \figured-bass {
    2 3 4+ 7 "9\"
  }
}
```

2 3 4 7 9

`\finger arg` (markup)

Set *arg* as small numbers for fingering instructions.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```

1 2 3 4 5

`\fontCaps arg` (markup)

Print *arg* in small caps.

This command sets the font-variant property to small-caps.

Unlike `\smallCaps`, which merely uses capital letters at a smaller font size, this uses the actual variant of the font for small caps. (As a consequence, if you configure a custom text font, this command has no effect if that font does not have a small caps variant.)

```
\markup \fontCaps "Small caps"
```

¹ Internally, this works by activating the ‘dlig’ OpenType feature of the Emmentaler font.

SMALL CAPS

`\fontsize` *increment* (number) *arg* (markup)

Increase current font size by *increment* to print *arg*.

This function adjusts the baseline-skip and word-space properties accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5 smaller
}
```

default *smaller*

Used properties:

- baseline-skip (2)
- word-space (1)
- font-size (0)

`\huge` *arg* (markup)

Set font size to value 2 to print *arg*.

```
\markup {
  default
  \hspace #2
  \huge huge
}
```

default *huge*

`\italic` *arg* (markup)

Print *arg* in italics.

This command sets the font-shape property to italic.

```
\markup {
  default
  \hspace #2
  \italic italic
}
```

default *italic*

`\large` *arg* (markup)

Set font size to value 1 to print *arg*.

```
\markup {
  default
  \hspace #2
  \large large
}
```

default *large*

`\larger` *arg* (markup)

Increase current font size by 1 to print *arg*.

This function adjusts the baseline-skip and word-space properties accordingly.

```
\markup {
```

```

    default
    \hspace #2
    \larger larger
}

```

default larger

`\magnify` *sz* (number) *arg* (markup)

Magnify current font by factor *sz* to print *arg*.

```

\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}

```

default 50% larger

Note that magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

`\normal-size-sub` *arg* (markup)

Set *arg* in subscript with a normal font size.

```

\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}

```

default subscript in standard size

Used properties:

- font-size (0)

`\normal-size-super` *arg* (markup)

Set *arg* in superscript with a normal font size.

```

\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}

```

default superscript in standard size

Used properties:

- font-size (0)

`\normal-text` *arg* (markup)

Print *arg* with default text font.

This resets all font-related properties (except the size), no matter what font was used earlier.

```

\markup {

```

```

\huge \bold \sans \fontCaps {
  huge bold sans caps
  \hspace #2
  \normal-text {
    huge normal
  }
  \hspace #2
  as before
}

```

HUGE BOLD SANS CAPS huge normal **AS BEFORE**

`\normal-weight arg` (markup)

Switch to normal weight (in contrast to bold) to print *arg*.

This command sets the font-series property to normal.

```

\markup {
  \bold {
    some bold text
    \hspace #2
    \normal-weight {
      normal font series
    }
    \hspace #2
    bold again
  }
}

```

some bold text normal font series **bold again**

`\normalsize arg` (markup)

Set font size to default (i.e., to value 0) to print *arg*.

```

\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}

```

this is very small **normal size** teeny again

`\number arg` (markup)

Print *arg* using the (music) font for numbers.

This font also contains symbols for figured bass, some punctuation, spaces of various widths, some letters and text variants of accidentals. Use `\dynamic` to access the (very small number of) letters. For accidentals you might use `\number` in combination with Unicode characters to access the text representation forms of accidental glyphs, as the following table shows.

Unicode value Unicode character

U+266D	♫
U+266E	♬
U+266F	♯
U+1D12A	✖
U+1D12B	ℬ

Examples:

```
\number ♫ → ♫
\number { \char ##x266F } → ♯
```

To get accidentals protected against overrides of font-name it is preferable to use `\text-doubleflat`, `\text-flat`, `\text-natural`, `\text-sharp`, `\text-doublesharp` or the general `\text-accidental` for the text variants of accidentals.

The appearance of digits in the Emmentaler font can be controlled with four OpenType features: ‘tnum’, ‘cv47’, ‘ss01’, and ‘kern’, which can be arbitrarily combined.

tnum	If off (which is the default), glyphs ‘zero’ to ‘nine’ have no left and right side bearings. If on, the glyphs all have the same advance width by making the bearings non-zero.
cv47	If on, glyphs ‘four’ and ‘seven’ have shorter vertical strokes. Default is off.
ss01	If on, glyphs ‘zero’ to ‘nine’ have a fatter design, making them more readable at small sizes. Default is off.
kern	If on (which is the default), provide pairwise kerning between (most) glyphs.

`\markuplist``\number``\fontsize #4.5`

```
\override #'((padding . 2)
               (baseline-skip . 4)
               (box-padding . 0)
               (thickness . 0.1))
```

```
\table #'(-1 -1 -1 -1) {
  0123456789 \box 147 \concat { \box 1 \box 4 \box 7 }
  \normal-text \normal-size "(time signatures)"
  \override #'(font-features .("cv47")) {
    0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
  \normal-text \normal-size "(alternatives)"
  \override #'(font-features .("tnum" "cv47" "-kern")) {
    0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
  \normal-text \normal-size "(fixed-width)"
```

```

\override #'(font-features . ("tnum" "cv47" "ss01")) {
  0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
\normal-text \normal-size "(figured bass)"
\override #'(font-features . ("cv47" "ss01")) {
  0123456789 \box 147 \concat { \box 1 \box 4 \box 7 } }
\normal-text \normal-size "(fingering)"
}

```

0123456789 **147** **147** (time signatures)

0123456789 **147** **147** (alternatives)

0123456789 **147** **147** (fixed-width)

0123456789 **147** **147** (figured bass)

0123456789 **147** **147** (fingering)

See also the markup commands `\figured-bass` and `\finger`, which set the font features accordingly.

`\overtie arg` (markup)

Print a tie over *arg*.

```

\markup \line {
  \overtie "overtied"
  \override #'(offset . 5) (thickness . 1))
  \overtie "overtied"
  \override #'(offset . 1) (thickness . 5))
  \overtie "overtied"
}

```

overtied **overtied** **overtied**

Used properties:

- shorten-pair ((0 . 0))
- height-limit (0.7)
- direction (1)
- offset (2)
- thickness (1)

`\replace replacements` (list) *arg* (markup)

Use *replacements* to replace strings in *arg*.

Each (*key* . *value*) pair of the *replacements* alist specifies what should be replaced; *key* gets replaced by *value*. Note the quasi-quoting syntax with a backquote in the second example.

```

\markup \replace #'(("2nd" . "Second"))
"2nd time"
\markup \replace
#`(("2nd" . ,#{ \markup \concat { 2 \super nd } #}))

```

```
\center-column {
  \line { Play only }
  \line { the 2nd time }
}
```

Second time

Play only
the 2nd time

Used properties:

- replacement-alist

`\sans arg` (markup)

Print *arg* with a sans-serif font.

This command sets the font-family property to sans.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

default sans serif

`\serif arg` (markup)

Print *arg* with a serif font.

This command sets the font-family property to serif.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \serif {
      text in serif font
    }
  }
  \hspace #2
  return to sans
}
```

sans serif, bold text in serif font return to sans

`\simple str` (string)

Print string *str*.

`\markup \simple "x"` is equivalent to `\markup "x"`. This command was previously used internally, but no longer is, and is being kept for backward compatibility only.

`\small arg` (markup)

Set font size to value -1 to print *arg*.

```
\markup {
  default
  \hspace #2
```

```
\small small
}
```

```
default small
```

`\smallCaps` *arg* (markup)

Print *arg* in (fake) small caps.

Unlike `\fontCaps`, which uses the actual small caps variant of the current font, this fakes small caps by using capital letters at a smaller font size. It can thus be used for fonts that don't have a small caps variant.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

```
default TEXT IN SMALL CAPS
```

`\smaller` *arg* (markup)

Decrease current font size by 1 to print *arg*.

This function adjusts the baseline-skip and word-space properties accordingly.

```
\markup {
  \fontsize #3.5 {
    large text
    \hspace #2
    \smaller { smaller text }
    \hspace #2
    large text
  }
}
```

```
large text smaller text large text
```

`\sub` *arg* (markup)

Set *arg* in subscript.

```
\markup { \concat { H \sub 2 O } }
```

```
H2O
```

See also `\super`.

Used properties:

- font-size (0)

`\super` *arg* (markup)

Set *arg* in superscript.

```
\markup { E = \concat { mc \super 2 } }
```

```
E = mc2
```

See also `\sub`.

Used properties:

- font-size (0)

`\teeny arg` (markup)

Set font size to value -3 to print *arg*.

```
\markup {
  default
  \hspace #2
  \teeny teeny
}

default    teeny
```

`\tie arg` (markup)

Add a horizontal bow at the bottom or top of *arg*.

This function uses `make-tie-stencil` to create the bow; it looks at the `thickness` and `offset` properties to determine the line thickness and vertical offset, respectively. The added bow fits the extent of *arg*; use the `shorten-pair` property to modify this. The `direction` property may be set explicitly using `override` or `direction` modifiers, or implicitly by using `voiceOne`, etc.

```
\markup {
  \override #'(direction . 1)
  \tie "above"
  \override #'(direction . -1)
  \tie "below"
}
```

above below

See also `\undertie` and `\overtie`, which are shorthands for this function.

Used properties:

- `shorten-pair` ((0 . 0))
- `height-limit` (0.7)
- `direction` (1)
- `offset` (2)
- `thickness` (1)

`\tiny arg` (markup)

Set font size to value -2 to print *arg*.

```
\markup {
  default
  \hspace #2
  \tiny tiny
}

default    tiny
```

`\typewriter arg` (markup)

Print *arg* in typewriter.

This command sets the `font-family` property to `typewriter`, also switching off the ‘`liga`’ OpenType feature to disable ligatures like ‘ff’ or ‘fi’.

```
\markup {
  "default fi ff"
  \hspace #2
```



```
\typewriter "typewriter fi ff"
}
```

```
default fi ff typewriter fi ff
```

`\underline` *arg* (markup)

Underline *arg*.

This function looks at the property `thickness` to determine the line thickness, at `offset` to determine the line's vertical offset from *arg*, and at `underline-skip` to determine the distance of additional lines from the others.

The `underline-shift` property is used to make subsequent calls work correctly. Overriding it makes little sense since it would end up adding the provided value to the one of `offset`.

```
\markup \justify-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(underline-skip . 4)
  \underline \underline \underline "underlined thrice"
}
```

```
underlined    underlined    underlined    underlined thrice
```

Used properties:

- `underline-skip` (2)
- `underline-shift` (0)
- `offset` (2)
- `thickness` (1)

`\undertie` *arg* (markup)

Print a tie under *arg*.

```
\markup \line {
  \undertie "undertied"
  \override #'((offset . 5) (thickness . 1))
  \undertie "undertied"
  \override #'((offset . 1) (thickness . 5))
  \undertie "undertied"
}
```

```
undertied undertied undertied
```

Used properties:

- `shorten-pair` ((0 . 0))
- `height-limit` (0.7)
- `direction` (1)
- `offset` (2)

- `thickness` (1)

`\upright arg` (markup)

Print *arg* upright.

This command is the opposite of `\italic`; it sets the font-shape property to upright.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

`\volta-number arg` (markup)

Set *arg* in a font appropriate for volta numbers.

```
\markup \volta-number "4."
```

4.

`\with-string-transformer transformer` (procedure) *arg* (markup)

Apply string transformer function *transformer* to *arg*.

Whenever a string is interpreted inside *arg*, function *transformer* is called first, and its result is then interpreted. The arguments passed to *transformer* are the output definition, the property alist chain, and the markup *arg*. See Sezione “New markup command definition” in *Estendere* about the two first arguments.

```
\markup \with-string-transformer
  #(\lambda (layout props str)
    (string-upcase str))
  \concat { "abc" \larger "def" }
```

ABCDEF

A.12.2 Markup for text alignment

`\abs-hspace amount` (number)

Create an invisible object taking up absolute horizontal space of *amount* points.

```
\markup {
  one
  \abs-hspace #20
  two
  \abs-hspace #40
  three
}
```

one two three

See also `\hspace`.

`\abs-vspace` *amount* (number)

Create an invisible object taking up absolute vertical space of *amount* points.

```
\markup {
  \center-column {
    one
    \abs-vspace #20
    two
    \abs-vspace #40
    three
  }
}
```

one

two

three

See also `\vspace`.

`\align-on-other` *axis* (non-negative, exact integer) *other-dir* (boolean-or-number) *other* (markup) *self-dir* (boolean-or-number) *self* (markup)

Align markup *self* on markup *other* along *axis*.

This function uses *self-dir* and *other-dir* for mutual alignment of *self* and *other*, respectively, translating *self* as requested relative to its surroundings. *other* is not printed.

If *self-dir* or *other-dir* is #f, use the reference point of *self* or *other*, respectively.

```
\markup \column {
  12
  \align-on-other #X #RIGHT 12
                                #LEFT 12345
  \align-on-other #X #RIGHT 123
                                #LEFT \fermata
  123
  \align-on-other #X #RIGHT 123
                                ##f \fermata
}
```

12

12345

◡

123

◡

`\center-align` *arg* (markup)

Align *arg* to its X center.

```
\markup {
  \column {
    one
    \center-align two
```

```

        three
      }
    }

```

```

      one
    two
  three

```

`\center-column` *args* (markup list)

Put *args* into a centered column.

See also `\column`.

```

\markup {
  \center-column {
    one
    two
    three
  }
}

```

```

      one
    two
  three

```

Used properties:

- `baseline-skip`

`\column` *args* (markup list)

Stack the markups in *args* vertically.

The property `baseline-skip` determines the space between markups in *args* (to be more precise, the space between the baselines of the markups).

```

\markup {
  \column {
    one
    two
    three
  }
}

```

```

      one
    two
  three

```

The baseline of the output of `\column` is the baseline of its first line.

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)

Print *arg1*, then print *arg2* on top of it.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; for this purpose use `\overlay` instead.

```

\markup {
  \fontsize #5
  \override #'(thickness . 2)

```

```
\combine
\draw-line #'(0 . 4)
\arrow-head #Y #DOWN ##f
}
```



`\concat args` (markup list)

Concatenate *args* in a horizontal line, without spaces in between.

Strings are concatenated on the input level, allowing ligatures. For example, `\concat { "f" "i" }` is equivalent to "fi".

```
\markup {
\concat {
one two three
}
}
```

onetwothree

`\dir-column args` (markup list)

Make a column of *args*.

Depending on the setting of the direction layout property, the arguments are stacked upwards or downwards.

```
\markup {
\override #`(direction . ,UP)
\dir-column {
going up
}
\hspace #1
\dir-column {
going down
}
\hspace #1
\override #'(direction . 1)
\dir-column {
going up
}
}
```

```
up          up
going going going
down
```

The baseline of the output of `\dir-column` is the baseline of its first line.

Used properties:

- `baseline-skip`
- `direction`

`\fill-line args` (markup list)

Put markups *args* into a horizontal line at fixed positions.

If there is a single argument, it gets centered. If there are two arguments, they get aligned to the left and right, respectively. Otherwise, if there are *n* arguments, the

markups are aligned to n equally spaced positions, with the first markup left-aligned, the last markup right-aligned, and the remaining markups centered at the respective positions. If there are no arguments, return an empty stencil.

The width of the horizontal line can be modified by overriding the `line-width` property. The space between arguments is at least as large as the value of the `word-space` property, at the cost of lengthening the line if necessary.

```
\markup {
  \column {
    \fill-line { Words positioned evenly across a line }
    \fill-line { | | | | | }
    \null
    \fill-line {
      \line { Text markups }
      \line { \italic { positioned evenly } }
      \line { across a line }
    }
    \null
    \override #'(line-width . 50)
    \fill-line { Width explicitly specified }
  }
}
```

Words	positioned	evenly	across	a	line
Text markups		<i>positioned evenly</i>		across a line	
Width		explicitly	specified		

See also `\justify-line`.

Used properties:

- `line-width` (#f)
- `word-space` (0.6)
- `text-direction` (1)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* at the start and end of a line, respectively, and fill the space in between with repeated *pattern* markups.

Patterns are spaced apart by *space* and aligned to direction *dir*. The width of the line is given by the `line-width` property.

```
\markup \column {
  "right-aligned:"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned:"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned:"
  \override #'(line-width . 50) {
```

```

\fill-with-pattern #2 #LEFT : left first
\fill-with-pattern #2 #LEFT : left second
}
}

```

right-aligned:

```

first ..... right
second ..... right

```

center-aligned:

```

left - - - - - right

```

left-aligned:

```

left : : : : : : : : : : : : : : first
left : : : : : : : : : : : : : : second

```

Used properties:

- line-width
- word-space

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)
Align *arg* in *axis* direction to the *dir* side.

```

\markup {
  \column {
    one
    \general-align #X #LEFT two
    three
    \null
    one
    \general-align #X #CENTER two
    three
    \null
    \line {
      one
      \general-align #Y #UP two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2 two
      three
    }
  }
}

```

```
one
two
three
```

```
one
two
three
```

```
one two three
one three
two
```

`\halign` *dir* (number) *arg* (markup)
Print *arg* with horizontal alignment set to *dir*.

If *dir* is -1, *arg* is left-aligned, while +1 makes it right-aligned. Values in between interpolate the alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT two
    three
    \null
    one
    \halign #CENTER two
    three
    \null
    one
    \halign #RIGHT two
    three
    \null
    one
    \halign #-5 two
    three
  }
}
```


one
two
three

one
two
three

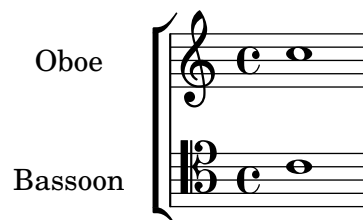
one
two
three

one
two
three

`\hcenter-in` *length* (number) *arg* (markup)

Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12 Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12 Bassoon
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)

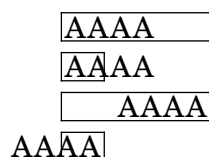
Create an invisible object taking up *amount* horizontal space.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```

one two three

amount can be also a negative value, which can be best visualized as if the current drawing point gets moved to the left.

```
\markup \concat {
  \hspace #4
  \column {
    \box \concat { AAAA \hspace #4 }
    \box \concat { AAAA \hspace #-4 }
    \box \concat { \hspace #4 AAAA }
    \box \concat { \hspace #-4 AAAA }
  }
}
```



See also `\abs-hspace`.

`\justify args` (markup list)

Print *args* as lines aligned both at the left and the right.

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

The baseline of the output of `\justify` is the baseline of its first line.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-field symbol` (symbol)

Justify the data that has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur"
```

```

        adipisicing elit, sed do eiusmod tempor incididunt
        ut labore et dolore magna aliqua. Ut enim ad minim
        veniam, quis nostrud exercitation ullamco laboris
        nisi ut aliquip ex ea commodo consequat."
    }

    \paper {
      bookTitleMarkup = \markup {
        \column {
          \fill-line { \fromproperty #'header:title }
          \null
          \justify-field #'header:myText
        }
      }
    }

    \markup {
      \null
    }

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify-line` *args* (markup list)

Put markups *args* into a horizontal line, equally spaced.

If there is a single argument, it gets centered. If there are two arguments, they get aligned to the left and right, respectively. Otherwise, the markups are spread to fill the entire line, separated by equally large spaces. If there are no arguments, return an empty stencil.

The width of the horizontal line can be modified by overriding the `line-width` property. The space between arguments is at least as large as the value of the `word-space` property, at the cost of lengthening the line if necessary.

```

    \markup {
      \justify-line {
        Constant space between neighboring words
      }
    }

```

Constant space between neighboring words

See also `\fill-line`.

Used properties:

- `line-width` (#f)
- `word-space` (0.6)
- `text-direction` (1)

`\justify-string arg (string)`

Print string *arg* as lines aligned both at the left and the right.

Paragraphs are indicated by double newlines. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \override #'(line-width . 40)
  \justify-string "Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut
    labore et dolore magna aliqua.
```

Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea commodo
consequat.

Excepteur sint occaecat cupidatat non proident, sunt
in culpa qui officia deserunt mollit anim id est
laborum"

```
}
```

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor
incidunt ut labore et dolore magna
aliqua.

Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.

Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

The baseline of the output of `\justify-string` is the baseline of its first line.

Used properties:

- `text-direction (1)`
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align arg (markup)`

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align two
    three
  }
}
```

one
two
three

`\left-column` *args* (markup list)

Put *args* into a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

```
one
two
three
```

Used properties:

- `baseline-skip`

`\line` *args* (markup list)

Put *args* into a horizontal line.

The property `word-space` determines the space between markups in *args*. For right-to-left scripts like Hebrew, `text-direction` should be set to -1.

```
\markup
  \override #'(word-space . 3)
  \column {
    \line { "A B" "C D" "E F" }
    \override #'(text-direction . -1)
    \line { "A B" "C D" "E F" }
  }
```

```
A B  C D  E F
E F  C D  A B
```

Used properties:

- `text-direction` (1)
- `word-space`

`\lower` *amount* (number) *arg* (markup)

Lower *arg* by the distance *amount*.

A negative *amount* indicates raising; see also `\raise`.

The argument to `\lower` is the vertical displacement amount, measured in (global) staff spaces, which is independent of the markup's current font size. If you need vertical movement that takes the font size into account, use `\translate-scaled` instead.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as `padding`, `Y-offset`, or `extra-offset`, or spacing variables such as `markup-system-spacing`.

```
\markup {
  one
  \lower #3 two
  three
```

}

one three
two

`\overlay` *args* (markup list)

Take a list of markups *args* and combine them.

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \overlay {
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
    \translate #'(0 . 4) \arrow-head #Y #UP ##f
  }
}
```



`\pad-around` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

Identical to function `\pad-markup`.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```

default

padded

`\pad-markup` *amount* (number) *arg* (markup)

Add padding *amount* all around *arg*.

Identical to function `\pad-around`.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}
```

default	padded
---------	--------

`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)

Make *arg* take at least *x-ext*, *y-ext* space.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}
```

default	padded
---------	--------

`\pad-x` *amount* (number) *arg* (markup)

Add padding *amount* around *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}
```

default	padded
---------	--------

`\pad-x-left` *amount* (number) *arg* (markup)

Add padding *amount* to the left of *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x-left #2 {
      padded
    }
  }
}
```

default	padded
---------	--------

`\pad-x-right` *amount* (number) *arg* (markup)

Add padding *amount* to the right of *arg* in the X direction.

```

\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x-right #2 {
      padded
    }
  }
}

```

default

padded

`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)

Put *arg2* next to *arg1* along *axis* to the *dir* side, without moving *arg1*.

```

\markup \column {
  text
  \put-adjacent #X #LEFT text *
  text
}

```

text

***text**

text

`\raise` *amount* (number) *arg* (markup)

Raise *arg* by the distance *amount*.

A negative *amount* indicates lowering, see also `\lower`.

The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces, which is independent of the markup's current font size. If you need vertical movement that takes the font size into account, use `\translate-scaled` instead.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as padding, Y-offset, or extra-offset, or spacing variables such as markup-system-spacing.

```

\markup { C \small \bold \raise #1.0 9/7+ }

```

C 9/7+

`\right-align` *arg* (markup)

Align *arg* on its right edge.

```

\markup {
  \column {
    one
    \right-align two
    three
  }
}

```


one
two
three

`\right-column` *args* (markup list)

Put *args* into a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)

Rotate *arg* by *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45°
  }
}
```

default

rotated 45°

`\translate` *offset* (pair of numbers) *arg* (markup)

Translate *arg* relative to its surroundings.

offset is a pair of numbers representing the displacement in the X and Y axes. See also `\translate-scaled`.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as padding, X-offset, Y-offset or extra-offset, or spacing variables such as markup-system-spacing.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
}
```

translated two spaces right, three up

*

`\translate-scaled` *offset* (pair of numbers) *arg* (markup)

Translate *arg* by *offset*, scaling the offset by the font size.

This function is normally used to move one element inside of a markup relative to the other elements. When using it on the whole markup, bear in mind that spacing mechanisms that place the markup itself on the page could cancel this shift. Consider using grob properties such as padding, X-offset, Y-offset or extra-offset, or spacing variables such as markup-system-spacing.

See also `\translate`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```

$\begin{matrix} & & \text{translate-scaled} \\ * & \text{translate} & * \end{matrix}$

Used properties:

- font-size (0)

`\vcenter` *arg* (markup)

Align *arg* to its Y center.

```
\markup {
  one
  \vcenter two
  three
}
```

one _{two} three

`\vspace` *amount* (number)

Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
  \center-column {
    one
    \vspace #1
    two
    \vspace #3
    three
  }
}
```

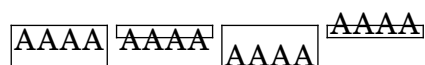
one

two

three

amount can be also a negative value, which can be best visualized as if the current drawing point gets moved up.

```
\markup {
  \vspace #1
  \box \column { AAAA \vspace #0.4 }
  \box \column { AAAA \vspace #-0.4 }
  \box \column { \vspace #0.4 AAAA }
  \box \column { \vspace #-0.4 AAAA }
}
```



See also `\abs-vspace`.

`\wordwrap` *args* (markup list)

Print *args* as left-aligned lines.

This function provides simple word-wrap. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore
    magna aliqua. Ut enim ad minim veniam, quis nostrud
    exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
 eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
 enim ad minim veniam, quis nostrud exercitation ullamco
 laboris nisi ut aliquip ex ea commodo consequat.

The baseline of the output of `\wordwrap` is the baseline of its first line.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-field` *symbol* (symbol)

Word-wrap the data that has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut
    labore et dolore magna aliqua. Ut enim ad minim
    veniam, quis nostrud exercitation ullamco laboris nisi
    ut aliquip ex ea commodo consequat."
}
```

```

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:myText
    }
  }
}

\markup {
  \null
}

```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\wordwrap-string arg (string)`

Print string *arg* as left-aligned lines.

Paragraphs are indicated by double newlines. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```

\markup {
  \override #'(line-width . 40)
  \wordwrap-string "Lorem ipsum dolor sit amet,
    consectetur adipisicing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua.

    Ut enim ad minim veniam, quis nostrud exercitation
    ullamco laboris nisi ut aliquip ex ea commodo
    consequat.

    Excepteur sint occaecat cupidatat non proident,
    sunt in culpa qui officia deserunt mollit anim id
    est laborum"
}

```

Lorem ipsum dolor sit amet,
 consectetur adipisicing elit, sed do
 eiusmod tempor incididunt ut labore et
 dolore magna aliqua.
 Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris
 nisi ut aliquip ex ea commodo
 consequat.
 Excepteur sint occaecat cupidatat non
 proident, sunt in culpa qui officia
 deserunt mollit anim id est laborum

The baseline of the output of `\wordwrap-string` is the baseline of its first line.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

A.12.3 Graphical markup

`\arrow-head axis (integer) dir (direction) filled (boolean)`

Print an arrow head along *axis* in direction *dir*.

Fill the head if *filled* is set to `#t`.

```

\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}

```

▲Υ ><

`\beam width (number) slope (number) thickness (number)`

Draw a beam with given *width*, *slope*, and *thickness*.

```

\markup {
  \beam #5 #1 #2
}

```



`\bracket arg (markup)`

Draw vertical brackets around *arg*.

```

\markup {

```

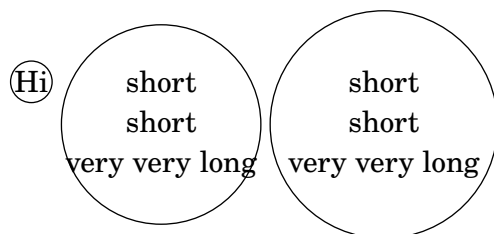
```
\bracket {
  \note {2.} #UP
}
```

`\circle arg` (markup)

Draw a circle around *arg*.

Use properties *thickness*, *circle-padding*, and *font-size* to set the line thickness and padding around the markup. If *bbox* is set to *#t*, make the circle enclose the bounding box of *arg*, otherwise use either the width or the height of *arg* (whatever is larger) as the diameter.

```
\markup {
  \circle {
    Hi
  }
  \circle {
    \center-column { "short" "short" "very very long" }
  }
  \override #'(bbox . #t) \circle {
    \center-column { "short" "short" "very very long" }
  }
}
```



Note that the circle does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

- *bbox* (*#f*)
- *circle-padding* (0.2)
- *font-size* (0)
- *thickness* (1)

`\draw-circle radius` (number) *thickness* (number) *filled* (boolean)

Draw a circle with given *radius* and *thickness*.

Fill the circle if *filled* is set to *#t*.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-dashed-line` *dest* (pair of numbers)

Draw a dashed line along vector *dest*.

Properties `on` and `off` give the length of a dash and the space between two dashes, respectively; `phase` shortens the first dash by the given amount.

If the `full-length` property is set to `#t` (which is the default), the value of property `off` (and `on` under some boundary conditions) gets adjusted so that there is neither whitespace at the end of the line nor the last dash truncated.

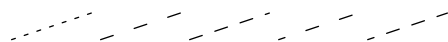
```
\markup {
  \override #'((on . 0.3) (off . 0.5))
  \draw-dashed-line #'(6 . 2)

  \draw-dashed-line #'(6 . 2)

  \override #'(full-length . #f)
  \draw-dashed-line #'(6 . 2)

  \override #'(phase . 0.5)
  \draw-dashed-line #'(6 . 2)

  \override #'((full-length . #f) (phase . 0.5))
  \draw-dashed-line #'(6 . 2)
}
```



Used properties:

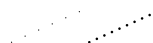
- `full-length` (`#t`)
- `phase` (0)
- `off` (1)
- `on` (1)
- `thickness` (1)

`\draw-dotted-line` *dest* (pair of numbers)

Draw a dotted line along vector *dest*.

Property `off` gives the space between two dots; its value gets adjusted so that the first and last dot exactly start and end the line, respectively. `phase` shifts all dots along the vector by the given amount.

```
\markup {
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'((thickness . 2) (off . 0.2))
  \draw-dotted-line #'(5.1 . 2.3)
}
```



Used properties:

- `phase` (0)
- `off` (1)
- `thickness` (1)

`\draw-hline`

Draw a horizontal line.

The property `span-factor` sets the length of the line as a multiple of the `line-width` property.

```
\markup {
  \column {
    \draw-hline
    \override #'(span-factor . 1/3)
    \draw-hline
  }
}
```



Used properties:

- `span-factor` (1)
- `line-width`
- `thickness` (1)

`\draw-line` *dest* (pair of numbers)

Draw a line along vector *dest*.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



Used properties:

- `thickness` (1)

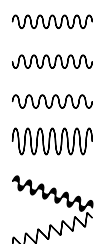
`\draw-squiggle-line` *sq-length* (number) *dest* (pair of numbers) *eq-end?* (boolean)

Draw a squiggled line along vector *dest*.

sq-length is the length of the first bow; this value gets always adjusted so that an integer number of squiggles is printed. If *eq-end?* is set to `#t`, the squiggled line ends with a bow in the same direction as the starting one.

The appearance of the squiggled line may be customized by overriding the `thickness`, `angularity`, `height`, and `orientation` properties.

```
\markup
\column {
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \override #'(orientation . -1)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \draw-squiggle-line #0.5 #'(6 . 0) ##f
  \override #'(height . 1)
  \draw-squiggle-line #0.5 #'(6 . 0) ##t
  \override #'(thickness . 5)
  \draw-squiggle-line #0.5 #'(6 . -2) ##t
  \override #'(angularity . 2)
  \draw-squiggle-line #0.5 #'(6 . 2) ##t
}
```

Used properties:

- orientation (1)
- height (0.5)
- angularity (0)
- thickness (0.5)

`\ellipse` *arg* (markup)

Draw an ellipse around *arg*.

Use properties `thickness`, `x-padding`, `y-padding`, and `font-size` to set the line thickness and padding around the markup.

This is the same as function `\oval` but with different padding defaults.

```
\markup {
  \ellipse {
    Hi
  }
}
```

(Hi)

Note that the ellipse does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

- y-padding (0.2)
- x-padding (0.2)
- font-size (0)
- thickness (1)

`\epsfile` *axis* (number) *size* (number) *file-name* (string)

Inline an image *file-name*, scaled along *axis* to *size*.

See `\image` for details on this command; calling

```
\markup \epsfile axis size file-name
```

is the same as

```
\markup
  \override #'(background-color . #f)
  \image axis size file-name
```

`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box of dimensions *xext* and *yext*, with rounded corners given by *blot*.

For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```
\markup {
```

```

\filled-box #'(0 . 4) #'(0 . 4) #0
\filled-box #'(0 . 2) #'(-4 . 2) #0.4
\combine
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
    \filled-box #'(3.6 . 5.6) #'(3.5 . 5.5) #0.7
}

```



`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

```

\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}

```

one two three

`\image` *axis* (number) *size* (number) *file-name* (string)

Inline an image *file-name*, scaled along *axis* to *size*.

The image format is determined based on the extension of *file-name*, which should be .png for a PNG image, or .eps for an EPS image (.PNG and .EPS are allowed as well).

EPS files are only supported in the PostScript backend – for all output formats –, and in the Cairo backend – when creating PostScript or EPS output.

If the image has transparency, it will appear over a colored background with the color specified by the background-color property, which defaults to "white".

To disable the colored background, set background-color to #f. For EPS images, this always works (where EPS images work in the first place). On the other hand, for PNG images, it works in the Cairo backend (which can output all supported formats), as well as in the SVG backend, but *not* in the PostScript backend, which is the default. See Sezione “Advanced command-line options for LilyPond” in *Uso del Programma* for how to activate the Cairo backend.

Use `\withRelativeDir` as a prefix to *file-name* if the file should be found relative to the input file.

Used properties:

- background-color ("white")

`\oval` *arg* (markup)

Draw an oval around *arg*.

Use properties thickness, x-padding, y-padding, and font-size to set the line thickness and padding around the markup.

This is the same as function `\ellipse` but with different padding defaults.

```

\markup {

```

```
\oval {
  Hi
}
```



Note that the oval does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

Used properties:

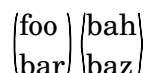
- `y-padding` (0.75)
- `x-padding` (0.75)
- `font-size` (0)
- `thickness` (1)

`\parenthesize arg` (markup)

Draw parentheses around *arg*.

This is useful for parenthesizing a column containing several lines of text.

```
\markup {
  \parenthesize
  \column {
    foo
    bar
  }
  \override #'(angularity . 2)
  \parenthesize
  \column {
    bah
    baz
  }
}
```



Used properties:

- `width` (0.25)
- `line-thickness` (0.1)
- `thickness` (1)
- `size` (1)
- `padding`
- `angularity` (0)

`\path thickness` (number) *commands* (list)

Draw a path with line *thickness* according to the directions given in *commands*.

commands is a list of lists where the car of each sublist is a drawing command and the cdr comprises the associated arguments for each command.

There are seven commands available to use in *commands*: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with ‘r’ are the relative variants of the other three commands. You may also use

the standard SVG single-letter equivalents: `moveto` = M, `lineto` = L, `curveto` = C, `closepath` = Z. The relative commands are written lowercase: `rmoveto` = r, `rlineto` = l, `rcurveto` = c.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments, namely the X and Y coordinates of the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `butt`, `round`, and `square`. Available line-join styles are `miter`, `round`, and `bevel`.

The property `filled` specifies whether or not the path is filled with color.

```
samplePath =
  #'((lineto -1 1)
    (lineto 1 1)
    (lineto 1 -1)
    (curveto -5 -5 -5 5 -1 0)
    (closepath))

\markup \scale #'(2 . 2) {
  \path #0.25 #samplePath

  \override #'(line-join-style . miter)
  \path #0.25 #samplePath

  \override #'(filled . #t)
  \path #0.25 #samplePath
}
```



Used properties:

- `filled` (#f)
- `line-join-style` (round)
- `line-cap-style` (round)

`\polygon` *points* (list of number pairs)

A polygon delimited by the list of *points*.

Property `extroversion` defines how the shape of the polygon is adapted to its thickness: if it is 0, the polygon is traced as-is. If it is -1, the outer side of the line is just on the given points. If set to 1, the line has its inner side on the points. The thickness property controls the thickness of the line; for filled polygons, this means the diameter of the blot.

```
regularPentagon =
  #'((1 . 0) (0.31 . 0.95) (-0.81 . 0.59)
```

```
(-0.81 . -0.59) (0.31 . -0.95))
```

```
\markup \scale #'(2 . 2) {
  \polygon #'((-1 . -1) (0 . -3) (2 . 2) (1 . 2))
  \override #'(filled . #f)
  \override #'(thickness . 2)
  \combine
    \with-color #(universal-color 'blue)
    \polygon #regularPentagon
    \with-color #(universal-color 'vermillion)
    \override #'(extroversion . 1)
    \polygon #regularPentagon
}
```



Used properties:

- thickness (1)
- filled (#t)
- extroversion (0)

`\postscript` *str* (string)

Insert *str* directly into the output as a PostScript command string.

This command is meant as a *last resort*. Almost all needs are better fulfilled by other markup commands (see, for example, `\path` and `\draw-line`). If you do use this command, keep the following points in mind:

- `\postscript` does not work in SVG output.
- Only a subset of the PostScript language is supported during the conversion from PostScript to PDF.
- There are no stability guarantees on the details of how LilyPond produces its own output (i.e., the context into which the PostScript code is inserted). They may change substantially across versions.
- LilyPond cannot understand the shape of the drawing, leading to suboptimal spacing. Usually, it is necessary to explicitly set up dimensions with a command like `\with-dimensions`.
- Depending on how you install LilyPond, the version of the PostScript interpreter (Ghostscript) can vary, and some of its features may be disabled.

str is processed with the following constraints.

- The string is embedded into the (intermediate) output file with the PostScript commands

```
gsave currentpoint translate 0.1 setlinewidth
before and
grestore
after it.
```

- After these preceding commands (i.e., `currentpoint translate`) the origin of the current transformation is the reference point of `\postscript`. Scale and

rotation of the current transformation reflect the global staff line distance and (if applied) other transformation markup commands (e.g., `\scale` and `\rotate`) encapsulating the `\postscript` command.

- The current point is set to the coordinate (0, 0).
- If an unwanted line appears at the beginning of your PostScript code, you are probably missing a call to `newpath`.

```
ringsps = "
  0.15 setlinewidth
  0.9 0.6 moveto
  0.4 0.6 0.5 0 361 arc
  stroke
  1.0 0.6 0.5 0 361 arc
  stroke
"
```

```
rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}
```

```
\relative c'' {
  c2^\rings
  a2_\rings
}
```



`\rounded-box arg` (markup)

Draw a box with rounded corners around *arg*.

This function looks at properties `thickness`, `box-padding`, and `font-size` to determine the line thickness and padding around the *arg*. The `corner-radius` property defines the radius of the round corners (default value is 1).

```
c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



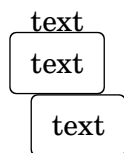
Note that the box does not horizontally displace its argument. Use markup commands like `\left-align` or `\table` to make LilyPond realign it.

```
\markup {
  \override #'(box-padding . 1.5)
  \column {
    "text"
  }
}
```

```

\rounded-box "text"
\left-align \rounded-box "text"
}
}

```



Used properties:

- box-padding (0.5)
- font-size (0)
- corner-radius (1)
- thickness (1)

`\scale` *factor-pair* (pair of numbers) *arg* (markup)

Scale *arg*.

factor-pair is a pair of numbers representing the scaling factor of the X and Y axes. Negative values may be used to produce mirror images.

```

\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}

```

stretched 

`\triangle` *filled* (boolean)

Draw a triangle.

Fill the triangle if *filled* is set to #t.

The appearance can be controlled with properties *extroversion*, *font-size*, and *thickness*.

```

\markup {
  \triangle ##t
  \triangle ##f
  \override #'(font-size . 5)
  \override #'(thickness . 5) {
    \override #'(extroversion . 1)
    \triangle ##f
    \override #'(extroversion . -1)
    \triangle ##f
  }
}

```



Used properties:

- thickness (1)

- `font-size` (0)
- `extroversion` (0)

`\with-url` *url* (string) *arg* (markup)

Add a link to URL *url* around *arg*.

This only works in the PDF backend.²

```
\markup {
  \with-url "https://lilypond.org/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... music notation for everyone

A.12.4 Markup for music and musical symbols

`\accidental` *alteration* (an exact rational number)

Select an accidental glyph for *alteration*, given as a rational number.

Use `\text-accidental` instead if you need glyph representation forms that fit and align well with text.

```
\markup {
  text
  \tiny { \accidental #1/2 \accidental #-1/2 }
  text
  \tiny { \text-accidental #1/2 \text-accidental #-1/2 }
  text
}
```

text #b text #b text

Used properties:

- `alteration-glyph-name-alist`

`\bar-line` *strg* (string)

Print a bar line in markup.

The allowed characters for input string *strg* are ‘;|.:!S[]{}’, having the same meaning as with the `\bar` command. The additional characters ‘{’ and ‘}’ denote left and right braces, respectively.

The output is vertically centered.

Changes of font-size are respected.

The default of height is 4 staff-space units. Apart from the bracket tips of a bracket bar line and the segno bar line all other bar lines are scaled with height. We don’t scale bracket tips and segno to meet the behavior of `SystemStartBracket` and the segno bar line.

`\bar-line` is further customizable by overriding `dot-count` and `dash-count` for dotted and dashed bar lines. The values for `hair-thickness`, `kern` and `thick-thickness` are customizable as well; defaults are the same as the values of the corresponding `BarLine` grob properties.


```
\markup {
```


² Due to technical limitations the link doesn’t work here in the Notation Reference.



```

\override #'(word-space . 2)
\column {
  \line {
    Examples
    \fontsize #-5 \translate-scaled #'(0 . 2) {
      \bar-line ":|.|:"
      \bar-line ";!S!;"
      \bar-line "]{|}["
    }
  }
  \line {
    Examples
    \fontsize #0 \translate-scaled #'(0 . 2) {
      \bar-line ":|.|:"
      \bar-line ";!S!;"
      \bar-line "]{|}["
    }
  }
  \line {
    Examples
    \fontsize #5 \translate-scaled #'(0 . 2) {
      \bar-line ":|.|:"
      \bar-line ";!S!;"
      \bar-line "]{|}["
    }
  }
}

```

Examples 

Examples 

Examples 

Used properties:

- thick-thickness (6.0)
- kern (3.0)
- hair-thickness (1.9)
- dash-count (5)
- dot-count (4)
- height (4)
- font-size (0)

\coda

Draw a coda sign.

```

\markup {

```

```
\coda
}
```

$$\emptyset$$

`\compound-meter` *time-sig* (number or pair)

Draw a numeric time signature based on *time-sig*.

time-sig can be a single number, a pair of numbers, a simple list, or a list of lists, as the following example demonstrates.

```
\markuplist {
  \override #'(baseline-skip . 4.5)
  \override #'(padding . 4.5)
  \table #'(-1 -1) {
    "Single number" \compound-meter #3
    "Conventional" \line {
      \compound-meter #'(4 . 4) or
      \compound-meter #'(4 4)
    }
    "Subdivided" \compound-meter #'(2 3 5 8)
    "Alternating" \line {
      \compound-meter #'((2) (3)) or
      \compound-meter #'((2 3 8) (3 4))
    }
  }
}
```

Single number **3**

Conventional **$\frac{4}{4}$ or $\frac{4}{4}$**

Subdivided **$2 + \frac{3}{8} + 5$**

Alternating **$2 + 3$ or $2 + \frac{3}{8} + \frac{3}{4}$**

Setting the `denominator-style` property to `note` prints denominators as a note and dots when exact representation is possible. Example:

```
\markup {
  \override #'(denominator-style . note)
  \line {
    \compound-meter #'(2 2) or
    \compound-meter #'(4 1/2) or
    \compound-meter #'((2 8/3) (3 4)) but not
    \compound-meter #'(8 20)
  }
}
```

$\frac{2}{2}$ or $\frac{4}{4}$ or $\frac{2}{2} + \frac{3}{8}$ but not $\frac{8}{20}$

The `nested-fraction-mixed` property controls whether fractional parts are printed as mixed numbers or as common fractions. Example:

```
\markup {
  \override #'(nested-fraction-mixed . #f)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-mixed . #t)
  \compound-meter #'(5/2 4)
}
```

$5/2$ or $2\frac{1}{4}$

The `nested-fraction-orientation` property controls how nested fractions are arranged. Supported values are `horizontal` and `vertical`. Example:

```
\markup {
  \override #'(nested-fraction-orientation . horizontal)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-orientation . vertical)
  \compound-meter #'(5/2 4)
}
```

$2^{1/2}$ or $2\frac{1}{4}$

The `nested-fraction-relative-font-size` property controls the size of the numerals in nested fractions. Recommended values are `-5.5` and `0`. Using large numerals may take precedence over related properties. Example:

```
\markup {
  \override #'(nested-fraction-relative-font-size . -5.5)
  \compound-meter #'(5/2 4) or
  \override #'(nested-fraction-relative-font-size . 0)
  \compound-meter #'(5/2 4)
}
```

$2\frac{1}{4}$ or $2\frac{1}{2}$

Used properties:

- `note-staff-position` (-2)
- `note-head-style` (())
- `note-flag-style` (())
- `note-dots-direction` (0)
- `nested-fraction-relative-font-size` (())
- `nested-fraction-orientation` (default)
- `nested-fraction-mixed` (#t)
- `font-size` (0)
- `denominator-style` (default)

`\customTabClef` *num-strings* (integer) *staff-space* (number)

Draw a clef in sans-serif style for a tablature with *num-strings* lines spaced by *staff-space*.

This markup command is used to implement `\clef moderntab` within a `TabStaff` context.

```
\markup {
  \customTabClef #4 #1
}
```



`\doubleflat`

Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`

Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```



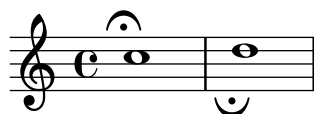
`\fermata` Create a fermata glyph.

If property `direction` is `DOWN`, use an inverted glyph.

Note that within music, one would normally use the `\fermata` articulation instead of a markup.

```
{ c''1^\markup \fermata d''1_\markup \fermata }
```

```
\markup { \fermata \override #^(direction . ,DOWN) \fermata }
```



Used properties:

- `direction (1)`

`\flat` Draw a flat symbol.

```
\markup {
  \flat
}
```



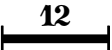
`\multi-measure-rest-by-number length` (non-negative, exact integer)

Return a multi-measure rest symbol for *length* measures.

If the number of measures is greater than the number given by `expand-limit` a thick horizontal line is printed. For every multi-measure rest lasting more than one

measure a number is printed on top. However, if property `multi-measure-rest-number` is set to `#t`, this number gets suppressed.

```
\markup {
  Multi-measure rests may look like
  \multi-measure-rest-by-number #12
  or
  \override #'(multi-measure-rest-number . #f)
  \multi-measure-rest-by-number #7
  (church rests)
}
```

Multi-measure rests may look like  or  (church rests)

Used properties:

- `multi-measure-rest-number` (`#t`)
- `width` (8)
- `expand-limit` (10)
- `hair-thickness` (2.0)
- `thick-thickness` (6.6)
- `word-space`
- `style` (`()`)
- `font-size` (0)

`\musicglyph` *glyph-name* (string)

Print music symbol *glyph-name*.

See Sezione “The Emmentaler font” in *Guida alla Notazione* for a complete listing of the possible glyph names.

```
\markup {
  \musicglyph "f"
  \musicglyph "rests.2"
  \musicglyph "clefs.G_change"
}
```



`\natural` Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note duration` (duration) *dir* (number)

Draw a note with a given duration and a stem.

The note duration is specified with *duration* (setting both the note head type and the number of augmentation dots). The stem direction and length is given by *dir*.

This function is wrapper around `\note-by-number`; its documentation gives more details on the available properties.

```
\markup {
```

```

\note {4..} #UP
\hspace #2
\override #'(style . cross)
  \note {4..} #0.75
\hspace #2
\note {\breve} #0
}

```



Used properties:

- `style ()`
- `dots-direction (0)`
- `flag-style ()`
- `font-size (0)`

`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)

Draw a note with a given length, a number of dots, and a stem.

The note length is specified with *log*, the number of dots with *dot-count*, and the stem direction and length with *dir*. Fractional values for *dir* can be used to control the length of the stem. Value 0 suppresses the stem completely; this is useful for note head styles that don't take stems or come with built-in stems (like Kievan).

Ancient note head styles (via the `style` property, vedi Sezione A.9 [Stili delle teste di nota], pagina 707) get mensural-style flags by default; use the `flag-style` property to override this. Supported flag styles are `default`, `old-straight-flag`, `modern-straight-flag`, `flat-flag`, `stacked`, `mensural`, and `neomensural`. The last flag style is the same as `mensural` and provided for convenience.

```

\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
  \hspace #2
  \override #'(style . petrucci)
    \note-by-number #3 #0 #UP
  \hspace #2
  \override #'(flag-style . modern-straight-flag)
    \note-by-number #4 #0 #DOWN
  \override #'(style . kievan)
    \note-by-number #2 #0 #0
}

```



See also function `\note`.

Used properties:

- `style ()`
- `dots-direction (0)`
- `flag-style ()`
- `font-size (0)`



`\rest` *duration* (*duration*)

Return a rest symbol with length *duration*.

If the `multi-measure-rest` property is set to `#t`, a multi-measure rest symbol may be returned. In this case the duration needs to be entered as `{ 1*N }` to get a multi-measure rest for *N* bars. Actually, only the scaling factor (i.e., the number after ‘*’) determines the length; the basic duration is disregarded.

See also functions `\rest-by-number` and `\multi-measure-rest-by-number` for more information on the used properties.

```
\markup {
  Rests:
  \hspace #2
  \rest { 4.. }
  \hspace #2
  \rest { \breve }
  \hspace #2
  Multi-measure rests:
  \override #'(multi-measure-rest . #t)
  {
    \hspace #2
    \rest { 1*7 }
    \hspace #2
    \rest { 1*12 }
  }
}
```

Rests:  Multi-measure rests: 

Used properties:

- `multi-measure-rest-number` (`#t`)
- `width` (8)
- `expand-limit` (10)
- `hair-thickness` (2.0)
- `thick-thickness` (6.6)
- `word-space`
- `style` (`()`)
- `font-size` (0)
- `style` (`()`)
- `ledgers` (`((-1 0 1))`)
- `font-size` (0)

`\rest-by-number` *log* (*integer*) *dot-count* (*integer*)

Draw a rest of length *log*, with *dot-count* dots.

For duration logs that appear in the `ledgers` property, rest symbols with ledger lines are selected.

```
\markup {
  \rest-by-number #3 #2
  \hspace #2
  \rest-by-number #0 #1
  \hspace #2
}
```

```

\rest-by-number #-1 #0
\hspace #2
\override #'(ledgers . ())
\rest-by-number #-1 #0
}

γ·· ▬. I ■

```

Used properties:

- `style (())`
- `ledgers ((-1 0 1))`
- `font-size (0)`

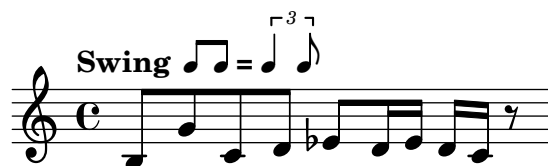
`\rhythm music (music)`

Draw embedded rhythmic pattern as specified by *music*.

```

\relative {
  \tempo \markup {
    Swing
    \hspace #0.4
    \rhythm { 8[ 8] } = \rhythm { \tuplet 3/2 { 4 8 } }
  }
  b8 g' c, d ees d16 ees d c r8
}

```



Within `\rhythm`, there is no time signature and no division in measures (as with `\cadenzaOn`, vedi [Musica in tempo libero], pagina 79). Beaming must be added explicitly with the syntax explained in [Travature manuali], pagina 101.

```

\markup {
  The rhythmic pattern \rhythm { 16[ 8 16] } is
  a type of syncopation.
}

```

The rhythmic pattern  is a type of syncopation.

`\stemDown` can be used to flip the stems.

```

\markup \rhythm { \stemDown 8 16 8 }

```



`\rhythm` works by creating a `StandaloneRhythmVoice` context, which is enclosed in a `StandaloneRhythmStaff` context, which is enclosed in a `StandaloneRhythmScore` context. It is possible to apply global tweaks to the output by using a `\layout` block.

```

\layout {
  \context {
    \StandaloneRhythmVoice
    \xNotesOn
  }
}

```


}

`\markup \rhythm { 8 16 8 }`

Used properties:

- `font-size` (-2)

`\score score (score)`Inline an image of music as specified by *score*.

The reference point (usually the middle staff line) of the lowest staff in the top system is placed on the baseline.

No page breaks and no MIDI output, i.e., both `\pageBreak` commands and `\midi{}` blocks get ignored.

```

\markup {
  Text before the score.
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
      }
    >>

    \layout {
      \indent = 0.0\cm
    }
  }
  Text after the score.
}

```

Text before the score.

Text after the score.

Used properties:

- `baseline-skip`

`\segno` Draw a segno symbol.

```

\markup {
  \segno

```

	}
	‰
<code>\semiflat</code>	Draw a semiflat symbol.
	<code>\markup {</code> <code>\semiflat</code> <code>}</code>
	♭
<code>\semisharp</code>	Draw a semisharp symbol.
	<code>\markup {</code> <code>\semisharp</code> <code>}</code>
	♯
<code>\sesquiflat</code>	Draw a 3/2 flat symbol.
	<code>\markup {</code> <code>\sesquiflat</code> <code>}</code>
	♭
<code>\sesquisharp</code>	Draw a 3/2 sharp symbol.
	<code>\markup {</code> <code>\sesquisharp</code> <code>}</code>
	♯
<code>\sharp</code>	Draw a sharp symbol.
	<code>\markup {</code> <code>\sharp</code> <code>}</code>
	♯
<code>\text-accidental</code>	<i>alteration</i> (an exact rational number)
	Select an accidental glyph for <i>alteration</i> (given as a rational number) that aligns well with text.
	<code>\markup {</code> <i>text</i> <code>\tiny { \text-accidental #1/2 \text-accidental #-1/2 }</code> <i>text</i> <code>}</code>
	<code>text #♭ text</code>
	Used properties:
	<ul style="list-style-type: none"> • <code>alteration-glyph-name-alist</code>

`\text-doubleflat`

Draw a double flat symbol for text.

```
\markup {
  \text-doubleflat
}
```

bb

`\text-doublesharp`

Draw a double sharp symbol for text.

```
\markup {
  \text-doublesharp
}
```

xx

`\text-flat`

Draw a flat symbol for text.

```
\markup {
  \text-flat
}
```

b

`\text-natural`

Draw a natural symbol for text.

```
\markup {
  \text-natural
}
```

n

`\text-sharp`

Draw a sharp symbol for text.

```
\markup {
  \text-sharp
}
```

#

`\tied-lyric str (string)`

Replace ‘~’ tilde symbols with tie characters in *str*.

```
\markup \column {
  \tied-lyric
    "Siam navi~all'onde~argenti Lasciate~in abbandono"
  \tied-lyric
    "Impetuosi venti I nostri~affetti sono"
  \tied-lyric
    "Ogni diletto~e scoglio Tutta la vita~e~un mar."
}
```

Siam navi~all'onde~argenti Lasciate~in abbandono
 Impetuosi venti I nostri~affetti sono
 Ogni diletto~e scoglio Tutta la vita~e~un mar.

Used properties:

- word-space

`\varcoda` Draw a varcoda sign.

```
\markup {
  \varcoda
}
```

⌘

A.12.5 Conditional markup

`\if condition?` (procedure) *argument* (markup)

Test *condition?*, and only insert *argument* if it is true.

The condition is provided as a procedure taking an output definition and a property alist chain. The procedure is applied, and its result determines whether to print the markup. This command is most useful inside `oddHeaderMarkup` or similar. Here is an example printing page numbers in bold:

```
\paper {
  oddHeaderMarkup =
    \markup \fill-line {
      ""
      \if #print-page-number
        \bold \fromproperty #'page:page-number-string
      }
  evenHeaderMarkup =
    \markup \fill-line {
      \if #print-page-number
        \bold \fromproperty #'page:page-number-string
      ""
    }
}
```

`\unless condition?` (procedure) *argument* (markup)

Test *condition?*, and only insert *argument* if it is false.

This function is similar to `\if`; the following example shows how to print the copyright notice on all pages but the last instead of just the first page.

```
\paper {
  oddFooterMarkup = \markup {
    \unless #on-last-page-of-part \fill-line {
      \fromproperty #'header:copyright
    }
  }
}

\header {
  copyright = "© LilyPond Authors. License: GFDL."
  tagline = "© LilyPond Authors. Documentation placed
under the GNU Free Documentation License
version 1.3."
}
```

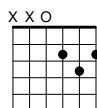
A.12.6 Instrument-specific markup

`\fret-diagram` *definition-string* (string)

Make a (guitar) fret diagram based on *definition-string*.

For example, say

```
\markup \fret-diagram "s:1.25;6-x;5-x;4-o;3-2;2-3;1-2;"
```



for fret spacing 5/4 of staff space, D chord diagram.

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
 - *s: number* – Set the fret spacing of the diagram (in staff spaces). Default: 1.
 - *t: number* – Set the line thickness (relative to normal line thickness). Default: 0.5.
 - *h: number* – Set the height of the diagram in frets. Default: 4.
 - *w: number* – Set the width of the diagram in strings. Default: 6.
 - *f: number* – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
 - *d: number* – Set radius of dot, in terms of fret spacing. Default: 0.25.
 - *p: number* – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
 - *c: string1-string2-fret* – Include a barre mark from *string1* to *string2* on *fret*.
 - *string-fret* – Place a dot on *string* at *fret*. If *fret* is 'o', *string* is identified as open. If *fret* is 'x', *string* is identified as muted.
 - *string-fret-fingering* – Place a dot on *string* at *fret*, and label with *fingering* as defined by the *f:* code.
- Note: There is no limit to the number of fret indications per string.

Used properties:

- *thickness* (0.5)
- *fret-diagram-details*
- *size* (1.0)
- *align-dir* (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

For example,

```
\markup \fret-diagram-terse "x;x;o;2;3;2;"
```



displays a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.

- Mute strings are indicated by ‘x’.
- Open strings are indicated by ‘o’.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a ‘-’ followed by the finger indicator, e.g., ‘3-2’ for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with -(to start a barre and -) to end the barre.

Used properties:

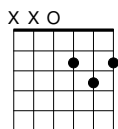
- thickness (0.5)
- fret-diagram-details
- size (1.0)
- align-dir (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

The following example produces a standard D chord diagram without fingering indications.

```
\markup \scale #'(1.5 . 1.5)
  \fret-diagram-verbose
    #'((mute 6) (mute 5) (open 4)
      (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```



Possible elements in *marking-list*:

(mute *string-number*)

Place a small ‘x’ at the top of string *string-number*.

(open *string-number*)

Place a small ‘o’ at the top of string *string-number*.

(barre *start-string end-string fret-number*)

Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

(capo *fret-number*)

Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

(place-fret *string-number fret-number [finger-value] [color-modifier]*
[color] ['parenthesized ['default-paren-color]])

Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, an optional color modifier *color-modifier*, an optional color *color*, an optional parenthesis 'parenthesized and an optional parenthesis color 'default-paren-color.

Use the `size` property to control the overall size, and the `thickness` property for the line thickness of the horizontal line and the divider.

The remaining configuration is done via the `harp-pedal-details` property; it contains the following elements:

```

box-offset
    vertical shift of the box center for up/down pedals

box-width
    box width

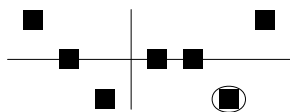
box-height
    box height

space-before-divider
    spacing between two boxes before the divider

space-after-divider
    spacing between two boxes after the divider

\markup {
  \override #'((size . 1.5)
                (harp-pedal-details . ((box-width . 1)
                                         (box-offset . 2))))

  \harp-pedal "^-v|--ov^"
}
```



For global modification of `harp-pedal-details`, i.e., outside of the current `\markup` block, you can also use code similar to

```
\override Voice.TextScript.harp-pedal-details.box-width = 1
```

Used properties:

- `thickness` (0.5)
- `harp-pedal-details` (())
- `size` (1.2)

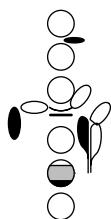
`\woodwind-diagram` *instrument* (symbol) *user-draw-commands* (list)

Make a woodwind-instrument diagram for *instrument* using *user-draw-commands*.

user-draw-commands is a list of alists, specifying the left-hand keys, the elements on the central column, and the right-hand keys. For example, this diagram

```

\markup \woodwind-diagram
  #'oboe #'((lh . (d ees))
            (cc . (five3qT1q))
            (rh . (gis)))
```



shows an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed, while the five-hole of the central column effectuating a trill between 1/4 and 3/4 is closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function call `(print-keys 'instrument)` in your `.ly` file, where *instrument* is the instrument whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

1q 1/4 covered

1h 1/2 covered

3q 3/4 covered

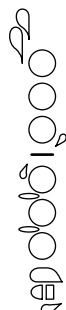
R ring depressed

F fully covered; the default if no state put

Additionally, these configurations can be used in trills. So, for example, `three3qTR` effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, `threeRT` effectuates a trill between R and open, whereas `threeTR` effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke `(print-keys-verbose 'instrument)`.

Lastly, substituting an empty list for the pressed-key alist results in a diagram with all of the keys drawn but none filled, for example

```
\markup \woodwind-diagram #'flute #'()
```



Used properties:

- `woodwind-diagram-details (())`
- `font-size (0)`
- `graphical (#t)`
- `thickness (0.1)`
- `size (1)`

A.12.7 Accordion registers

`\discant` *name* (string)

Generate a discant accordion register symbol for *name*.

To make it available,





```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The register names in the default `\discant` register set have been modeled after the numeric Swiss notation (as depicted in http://de.wikipedia.org/wiki/Register_%28Akkordeon%29), omitting the slashes and dropping leading zeros.

The string *name* is basically a three-digit number with the lowest digit specifying the number of 16' reeds, the tens the number of 8' reeds, and the hundreds specifying the number of 4' reeds. Without modification, the specified number of reeds in 8' is centered in the symbol. Newer instruments may have registrations where 8' can be used either within or without a tone chamber, 'cassotto'. Notationally, the central dot then indicates use of cassotto. One can suffix the tens' digits '1' and '2' with '+' or '-' to indicate clustering the dots at the right or left, respectively, rather than centered.

Some examples are

	
<code>\discant "1"</code>	<code>\discant "1+0"</code>
	
<code>\discant "120"</code>	<code>\discant "131"</code>

Used properties:

- `font-size` (0)

`\freeBass` *name* (string)




Generate a free bass/converter accordion register symbol for the usual two-reed layout as given by *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

Available registrations are

	
<code>\freeBass "1"</code>	<code>\freeBass "11"</code>
	
<code>\freeBass "10"</code>	

Used properties:

- `font-size` (0)

`\stdBass` *name* (string)

Generate a standard bass accordion register symbol for *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The default bass register definitions have been modeled after the article <http://www.accordions.com/index/art/stradella.shtml> originally appearing in *Accord Magazine*.








The underlying register model is



This kind of overlapping arrangement is common for Italian instruments though the exact location of the octave breaks differ.

When not composing for a particular target instrument, using the five-reed definitions makes more sense than using a four-reed layout: in that manner, the ‘Master’ register is unambiguous. This is rather the rule in literature bothering about bass registrations at all.

Available registrations are

	
<code>\stdBass "Soprano"</code>	<code>\stdBass "Soft Bass"</code>
	
<code>\stdBass "Alto"</code>	<code>\stdBass "Soft Tenor"</code>
	
<code>\stdBass "Tenor"</code>	<code>\stdBass "Bass/Alto"</code>
	
<code>\stdBass "Master"</code>	

Used properties:

- `font-size (0)`

`\stdBassIV` *name* (string)

Generate a standard bass accordion register symbol for *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.









The main use is for four-reed standard bass instruments with reedbank layout



Notable instruments are Morino models with MIII (the others are five-reed instead) and the Atlantic IV. Most of those models have three register switches. Some newer Morinos with MIII might have five or even seven.

The prevalent three-register layout uses the middle three switches ‘Tenor’, ‘Master’, ‘Soft Bass’. Note that the sound is quite darker than the same registrations of ‘c,’-based instruments.

Available registrations are

	
<code>\stdBassIV "Soprano"</code>	<code>\stdBassIV "Soft Bass"</code>
	
<code>\stdBassIV "Alto"</code>	<code>\stdBassIV "Bass/Alto"</code>
	
<code>\stdBassIV "Tenor"</code>	<code>\stdBassIV "Soft Bass/Alto"</code>
	
<code>\stdBassIV "Master"</code>	<code>\stdBassIV "Soft Tenor"</code>

Used properties:

- `font-size (0)`

`\stdBassV` *name* (string)

Generate a standard bass accordion register symbol for *name*.

To make it available,

```
#(use-modules (lily accreg))
```

is required near the top of your input file.

The main use is for five-reed standard bass instruments with reedbank layout





This tends to be the bass layout for Hohner's Morino series without converter or MIII manual.

With the exception of the rather new 7-register layout, the highest two chord reeds are usually sounded together. Older instruments offer 5 or 3 bass registers. The Tango VM offers an additional 'Solo Bass' setting that mutes the chord reeds. The symbol on the register buttons of the Tango VM would actually match the physical five-octave layout reflected here, but it is not used in literature.

Composers should likely prefer the five-reed versions of these symbols. The mismatch of a four-reed instrument with five-reed symbols is easier to resolve for the player than the other way round.

Available registrations are

	
<code>\stdBassV "Bass/Alto"</code>	<code>\stdBassV "Soft Bass"</code>
	
<code>\stdBassV "Soft Bass/Alto"</code>	<code>\stdBassV "Soft Tenor"</code>
	
<code>\stdBassV "Alto"</code>	<code>\stdBassV "Soprano"</code>
	
<code>\stdBassV "Tenor"</code>	<code>\stdBassV "Sopranos"</code>
	
<code>\stdBassV "Master"</code>	<code>\stdBassV "Solo Bass"</code>

Used properties:

- `font-size (0)`

`\stdBassVI` *name* (string)

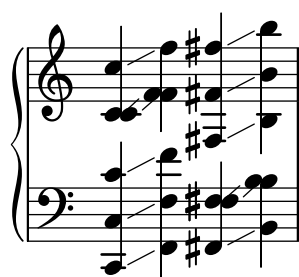
Generate a standard bass accordion register symbol for six-reed basses as given by *name*.

To make it available,

```
 #(use-modules (lily accreg))
```








is required near the top of your input file.

This is primarily the register layout for the Hohner "Gola" model. The layout is



The registers are effectively quite similar to that of `\stdBass`. An additional bass reed at alto pitch is omitted for aesthetical reasons from the ‘Master’ setting, so the symbols are almost the same except for the ‘Alto/Soprano’ register with bass notes at Alto pitch and chords at Soprano pitch.

Available registrations are

	
<code>\stdBassVI "Soprano"</code>	<code>\stdBassVI "Alto/Soprano"</code>
	
<code>\stdBassVI "Alto"</code>	<code>\stdBassVI "Bass/Alto"</code>
	
<code>\stdBassVI "Soft Tenor"</code>	<code>\stdBassVI "Soft Bass"</code>
	
<code>\stdBassVI "Master"</code>	

Used properties:

- `font-size (0)`

A.12.8 Other markup commands

`\annotate-moving arg (markup)`

Indicate `\vspace` and `\hspace` movement with an arrow.

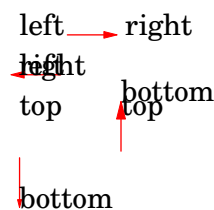
The arrow changes its size and thickness depending on the printed length; the maximum size of the arrow head can be controlled with the `size` property. If `size` exceeds a third of the length of the final arrow, it falls back to that third.

Note that the arrows do not reflect the actual extents of the objects created by `\vspace` and `\hspace`; you might use `\box` for that.

```

\markup
\column {
  \line { left \annotate-moving \hspace #4 right }
  \line { left \annotate-moving \hspace #-4 right }
  \line {
    \column {
      top
      \override #'(size . 0.6)
      \annotate-moving \vspace #4/3 bottom
    }
    \column {
      top
      \override #'(size . 2.0)
      \annotate-moving \vspace #-4/3 bottom
    }
  }
}

```



Used properties:

- size (1)
- color ("red")

`\append-to-tag` *tag* (symbol) *more* (markup) *arg* (markup)

Append *more* to all markup in var *arg* tagged with *tag*.

It works similar to `\appendToTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \append-to-tag #'foo postfoo \tagged }
```

A postfoo B

Used properties:

- tags-with-appends-alist (())

`\auto-footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \auto-footnote a b
  \override #'(padding . 0.2)
  \auto-footnote c d
}
```

a¹ c²

¹b _____
²d

The footnote will be annotated automatically.

Used properties:

- padding (0.0)

- `raise (0.5)`

`\backslashed-digit` *num* (integer)

Print number *num* with the Emmentaler font, crossed through with a backslash.

This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```

5 7

Used properties:

- `thickness (1.6)`
- `font-size (0)`

`\char` *num* (integer)

Produce a single Unicode character with code *num*.

Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```

A ©

`\eyeglasses`

Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```

👁👁

`\first-visible` *args* (markup list)

Use the first markup in *args* that yields a non-empty stencil and ignore the rest.

```
\markup {
  \first-visible {
    \fromproperty #'header:composer
    \italic Unknown
  }
}
```

Unknown

`\footnote` *mkup* (markup) *note* (markup)

Have footnote *note* act as an annotation to the markup *mkup*.

```
\markup {
  \footnote a b
  \override #'(padding . 0.2)
  \footnote c d
}
```


a c

b _____
d

The footnote will not be annotated automatically.

`\fraction` *arg1* (markup) *arg2* (markup)

Make a fraction of markups *arg1* and *arg2*.

```
\markup {
  π ≈ \fraction 355 113
}
```

$$\pi \approx \frac{355}{113}$$

Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)

Read *symbol* from the property settings and produce a stencil from the markup contained within.

If *symbol* is not defined or is not a markup, return an empty markup.

Currently, the following properties can be accessed.

- Within a `\paper` block defining titles, headers, or footers, or within a `\header` block: all fields from the `\header` block (that produce markup) are available, with `header:` as a name prefix.
- Within a `\paper` block defining headers or footers: the current page number (`symbol page:page-number-string`).
- Within the `tocItemMarkup` paper variable (or in custom-made Scheme code that uses function `add-toc-item!`) defining a table of contents entry: the entry's text and page number are available as `toc:text` and `toc:page`, respectively. An entry's indentation markup is available as `toc:indent`.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
```

```
\null
}
```

from *myTitle*

`\keep-with-tag` *tags* (symbol list or symbol) *arg* (markup)

Keep markup from *arg* that is untagged or tagged with *tags*.

All other parts of *arg* that are using a different tag are replaced with empty stencils.

It works similar to `\keepWithTag` for music, but only with markups.

```
tagged = \markup {
  untagged
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \keep-with-tag #'bar \tagged }
\markup { \keep-with-tag #'foo \tagged }
```

untagged B

untagged A

Used properties:

- `tags-to-keep` `(())`

`\left-brace` *size* (number)

Print a brace from the music font, of height *size* (in points).

```
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}
```

{ }

`\lookup` *glyph-name* (string)

Print a brace glyph with name *glyph-name*.

This is a historical command; `\left-brace` (which looks up the glyph by absolute size and is independent of the font size) is recommended instead.

```
\markup \lookup "brace200"
```

{

`\markalphabet` *num* (integer)

Make a markup letter for *num*.

The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

H Z

`\markletter` *num* (integer)

Make a markup letter for *num*.

The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

H AA

`\null`

An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (procedure) *arg* (markup)

Apply the *procedure* markup command to *arg*.

procedure takes the same arguments as `interpret-markup` and returns a stencil.

`\override` *new-prop* (pair) *arg* (markup)

Add the argument *new-prop* to the property list for printing *arg*.

In general, any property may be overridden that is part of font-interface (Sezione “font-interface” in *Guida al Funzionamento Interno*), text-interface (Sezione “text-interface” in *Guida al Funzionamento Interno*), or instrument-specific-markup-interface (Sezione “instrument-specific-markup-interface” in *Guida al Funzionamento Interno*). Additionally, various markup commands listen to other properties, too, as described in a markup function’s documentation.

new-prop is either a single alist pair or a non-empty list of alist pairs.

```
\markup {
  \undertie "undertied"
  \override #'(offset . 15)
  \undertie "offset undertied"
  \override #'((offset . 15) (thickness . 3))
  \undertie "offset thick undertied"
}
```

undertied offset undertied offset thick undertied

`\page-link` *page-number* (number) *arg* (markup)

Add a link to a score’s page *page-number* around *arg*.

This only works in the PDF backend.

```
\markup {
  \page-link #2 { \italic { This links to page 2... } }
}
```

This links to page 2...

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)

Print a page number reference.

label is the label set on the referenced page (using `\label` or `\tocItem`), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

If the current book or book part is set to use roman numerals for page numbers, the reference will be formatted accordingly – in which case the *gauge*'s width may require additional tweaking.

Used properties:

- `x-align` (1)

`\pattern` *count* (non-negative, exact integer) *axis* (non-negative, exact integer) *space* (number) *pattern* (markup)

Print a *pattern* markup *count* times.

Patterns are spaced apart by *space* (defined as for `\hspace` or `\vspace`, respectively) and distributed on *axis*.

```
\markup \column {
  "Horizontally repeated:"
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated:"
  \pattern #3 #Y #0.5 \flat
}
```

Horizontally repeated:

b b b b b b b

Vertically repeated:

b
b
b

`\property-recursive` *symbol* (symbol)

Print out a warning when header field markup in *symbol* contains some recursive markup definition.

`\push-to-tag` *tag* (symbol) *more* (markup) *arg* (markup)

Prepend *more* to all markup in *arg* tagged with *tag*.

It works similar to `\pushToTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \push-to-tag #'foo prefoo \tagged }

prefoo A B
```

Used properties:

- tags-with-pushes-alist (())

`\qr-code` *width* (non-negative number) *str* (string)

Insert a QR code for string *str*, usually a URL, with a given *width*.

```
\markup \vcenter {
  \center-column { Engraved with LilyPond }
  \hspace #1.5
  \qr-code #10.0 "https://lilypond.org"
}
```

Engraved
with
LilyPond



The `error-correction-level` property can be set to one of the symbols `low`, `medium`, `quarter`, or `high`. The higher the level of error correction is, the more the QR code contains redundancy, potentially helping detectors, e.g., in poor lighting conditions; however, a higher correction level also makes the code denser.

```
\markup \vcenter {
  \center-column { Engraved with LilyPond }
  \hspace #1.5
  \override #'(error-correction-level . high)
  \qr-code #10.0 "https://lilypond.org"
}
```

Engraved
with
LilyPond



The `quiet-zone-size` property specifies the width of the “quiet zone”, namely the white area around the QR code. It is expressed as a multiple of the width of one little square inside the QR code. Use at least 4 for best results.

Used properties:

- quiet-zone-size (4)
- error-correction-level (low)

`\remove-with-tag` *tags* (symbol list or symbol) *arg* (markup)

Remove markup from *arg* that is tagged with *tags*.

The removed markup is replaced with empty stencils. It works similar to `\removeWithTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \remove-with-tag #'foo \tagged }
\markup { \remove-with-tag #'bar \tagged }
```

B

A

Used properties:

- `tags-to-remove (())`

`\replace-with-tag` *tag* (symbol) *replacement* (markup) *arg* (markup)

Replace tagged markups in *arg*.

Everything tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

It works similar to `\replaceWithTag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}
```

```
\markup { \replace-with-tag #'foo C \tagged }
```

C B

Used properties:

- `tags-with-replacement-alist (())`

`\right-brace` *size* (number)

A music brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```

} }

`\slashed-digit` *num* (integer)

Print number *num* with the Emmentaler font, crossed through with a slash.

This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```

5 7

Used properties:

- `thickness (1.6)`

- `font-size (0)`

`\stencil stil (stencil)`

Use stencil *stil* as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`

Create a box of the same height as the space in the current font.

`\tag tags (symbol list or symbol) arg (markup)`

Tag markup *arg* with *tag*.

tag can be one or multiple tags. This allows later on to reference *arg*; for example, to remove it or to add markup before or after the tagged markup. It works similar to `\tag` for music, but only with markups.

```
tagged = \markup {
  \tag #'foo A
  \tag #'bar B
}

\markup { \keep-with-tag #'bar \tagged }
\markup { \keep-with-tag #'foo \tagged }
```

B

A

Used properties:

- `tags-with-replacement-alist (())`
- `tags-with-appends-alist (())`
- `tags-with-pushes-alist (())`
- `tags-to-remove (())`
- `tags-to-keep (())`

`\transparent arg (markup)`

Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

`\verbatim-file name (string)`

Read the contents of file *name* and include it verbatim.

```
\markup {
  \verbatim-file "en/included/simple.ly"
}
```

```
% A simple piece in LilyPond, a scale.
\version "2.19.21"
\relative {
  c' d e f g a b c
}
```

Use `\withRelativeDir` as a prefix to *name* if the file should be found relative to the input file.

`\whiteout` *arg* (markup)

Provide a white background for *arg*.

The shape of the white background is determined by the *style* property. The default is *box* which produces a rectangle. *rounded-box* produces a rounded rectangle, and *outline* approximates the outline of the markup.

The color of the background can be controlled with the *color* property, defaulting to "white".

```
\markup {
  \combine
    \filled-box #'(-1 . 62) #'(-3 . 4) #1
    \override #'(line-width . 60)
    \fill-line {
      \override #'(thickness . 1.5)
      \whiteout box
      \override #'((style . rounded-box) (thickness . 3))
      \whiteout rounded-box
      \override #'((style . outline) (thickness . 3))
      \whiteout outline
      \override #'((color . "red") (style . outline))
      \whiteout red-outline
    }
}
```

box

rounded-box

outline

red-outline

Used properties:

- *color* ("white")
- *thickness* (())
- *style* (box)

`\with-color` *col* (color) *arg* (markup)

Use color *col* to draw *arg*.

Vedi [Colorare gli oggetti], pagina 241, for valid color specifications.

```
\markup {
  \with-color #red red
  \hspace #2
  \with-color #green green
  \hspace #2
  \with-color "#0000ff" blue
}
```

red **green** **blue**

`\with-dimension axis (integer) val (pair of numbers) arg (markup)`

Set the dimension of *arg* along *axis* to *val*.

If *axis* is equal to X, set the horizontal dimension. If *axis* is equal to Y, set the vertical dimension.

`\with-dimension-from axis (integer) arg1 (markup) arg2 (markup)`

Print *arg2* but replace the dimension along *axis* with the one from *arg1*.

If *axis* is set to X, replace the horizontal dimension. If *axis* is set to Y, replace the vertical dimension.

`\with-dimensions x (pair of numbers) y (pair of numbers) arg (markup)`

Set the horizontal and vertical dimensions of *arg* to *x* and *y*.

`\with-dimensions-from arg1 (markup) arg2 (markup)`

Print *arg2* with the horizontal and vertical dimensions of *arg1*.

`\with-link label (symbol) arg (markup)`

Add a link to the page holding label *label* around *arg*.

This only works in the PDF backend.

```
\markup {
  \with-link #'label {
    \italic { This links to the page
              containing the label... }
  }
}
```

`\with-outline outline (markup) arg (markup)`

Print *arg* with the outline and dimensions of *outline*.

The outline is used by skylines to resolve collisions (not for whiteout).

`\with-true-dimension axis (integer) arg (markup)`

Give *arg* its actual dimension (extent) on *axis*.

Sometimes, the extents of a markup's printed ink differs from the default extents. The main case is if glyphs are involved. By default, the extents of a glyph are based on the glyph's *metrics* (i.e., a default vertical and horizontal size for the glyph), which, for various reasons, are often not identical to its *bounding box* (i.e., the smallest rectangle that completely encompasses the glyph's outline) – in most cases, the outline protrudes the box spanned up by the metrics.

```
\markup {
  text
  \fontsize #10
  \override #'((box-padding . 0) (thickness . 0.2))
  \box
  \musicglyph "scripts.trill"
  text
}
```



For purposes other than setting text, this behavior may not be wanted. You can use `\with-true-dimension` in order to give the markup its actual printed extent.

```
\markup {
```

```

text
\fontsize #10
\override #'((box-padding . 0) (thickness . 0.2))
\box
  \with-true-dimension #X
  \musicglyph "scripts.trill"
text
}
```



`\with-true-dimensions arg (markup)`

Give *arg* its actual dimensions (extents).

Calling

```
\markup \with-true-dimensions arg
```

is short for

```

\markup
  \with-true-dimension #X
  \with-true-dimension #Y
arg
```

i.e., `\with-true-dimensions` has the effect of `\with-true-dimension` on both axes.

A.13 Comandi per una lista di *markup*

Tutti i comandi seguenti possono essere usati all'interno di `\markuplist`:

`\column-lines args (markup list)`

Stack the markups in *args* vertically.

Like `\column`, but return a list of lines instead of a single markup. The property `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines args (markup list)`

Print *args* as lines aligned both at the left and the right.

Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction (1)`
- `word-space`
- `line-width (#f)`
- `baseline-skip`

`\override-lines new-prop (pair) args (markup list)`

Add the argument *new-prop* to the property list for printing *args*.

Like `\override` but for markup lists.

`\score-lines score (score)`

Inline an image of music as specified by *score*.

Like `\score` but return a list of lines instead of a single markup.

Used properties:

- `tags-with-replacement-alist` (`()`)
- `tags-with-appends-alist` (`()`)
- `tags-with-pushes-alist` (`()`)
- `tags-to-remove` (`()`)
- `tags-to-keep` (`()`)

`\string-lines` *str* (string)

Split string *str* into lines.

The character to split at is specified by the property `split-char`, defaulting to `#\newline`. Surrounding whitespace is removed from every resulting string. The returned list of markups is ready to be formatted by other markup or markup list commands like `\column`, `\line`, etc.

```
\markup {
  \column
    \string-lines
      "foo, foo,
      bar, bar,
      buzz, buzz!"
}
```

```
foo, foo,
bar, bar,
buzz, buzz!
```

Used properties:

- `split-char` (`#\newline`)

`\table` *column-align* (number list) *lst* (markup list)

Print a table.

column-align specifies how each column is aligned; possible values are -1, 0, and 1. The number of elements in *column-align* determines how many columns will be printed.

The entries to print are given by *lst*, a markup list. If needed, the last row is filled up with point-stencils.

Override the padding property to increase the horizontal distance between columns.

Override `baseline-skip` to increase the vertical distance between rows.

```
% A markup command to print a fixed-width number.
\markup fnum =
  \markup \override #'(font-features . ("ss01" "-kern"))
  \number \etc
```

```
\markuplist {
  \override #'(padding . 2)
  \table #'(0 1 0 -1) {
    \underline { center-aligned right-aligned
                  center-aligned left-aligned }
one      \fnum    1 thousandth \fnum 0.001
eleven   \fnum    11 hundredth  \fnum 0.01
twenty   \fnum    20 tenth      \fnum 0.1
```

```

    thousand \fwnum 1000 one \fwnum 1.0
  }
}

```

center-aligned right-aligned center-aligned left-aligned

one	1	thousandth	0.001
eleven	11	hundredth	0.01
twenty	20	tenth	0.1
thousand	1000	one	1.0

Used properties:

- baseline-skip
- padding (0)

`\table-of-contents`

Print a table of contents.

This function uses the paper variable `tocTitleMarkup` for the title; it then prints `\tocItem` entries line by line.

Vedi Sezione 3.2.6 [Indice], pagina 521, for a complete discussion.

Used properties:

- baseline-skip

`\tag-list` *tags* (symbol list or symbol) *arg* (markup list)

Tag markup list *arg* with *tag*.

tag can be one or multiple tags. This allows later on to reference *arg*; for example, to remove it or to add markup before or after the tagged markup list.

It works like the `\tag` command for markups but with markup lists. You will need it if you have to reference a whole list; for example, to use `\push-to-tag` and `\append-to-tag` without pushing or appending before or after every single item of the list, but before or after the whole list instead.

```

tagged = \markuplist {
  \tag-list #'foo { foo bar }
}

\markup { \push-to-tag #'foo test \tagged }

test foo bar

```

Used properties:

- tags-with-replacement-alist (())
- tags-with-appends-alist (())
- tags-with-pushes-alist (())
- tags-to-remove (())
- tags-to-keep (())

`\wordwrap-lines` *args* (markup list)

Print *args* as left-aligned lines.

Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

A.14 Elenco dei caratteri speciali

Si possono usare i seguenti riferimenti per i caratteri speciali; maggiori informazioni in vedi [Alias ASCII], pagina 533.

Si usa la sintassi HTML. Di questi riferimenti molti sono identici a quelli HTML, alcuni sono ispirati a L^AT_EX.

I caratteri sono compresi in un riquadro per rendere visibile la loro dimensione. È stato aggiunto un po' di spazio tra il carattere e il riquadro per migliorare la leggibilità.

<code>&iexcl;</code>	<code>¡</code>	<code>&iquest;</code>	<code>¿</code>	<code>&solidus;</code>	<code>/</code>	<code>&flq;</code>	<code>◁</code>
<code>&frq;</code>	<code>▷</code>	<code>&flqq;</code>	<code>◀</code>	<code>&frqq;</code>	<code>▶</code>	<code>&glq;</code>	<code>▷</code>
<code>&grq;</code>	<code>◁</code>	<code>&glqq;</code>	<code>▶</code>	<code>&grqq;</code>	<code>◁</code>	<code>&elq;</code>	<code>◁</code>
<code>&erq;</code>	<code>▷</code>	<code>&elqq;</code>	<code>◁</code>	<code>&erqq;</code>	<code>▷</code>	<code>&ensp;</code>	<code>—</code>
<code>&emsp;</code>	<code>—</code>	<code>&thinsp;</code>	<code>—</code>	<code>&nbsp;</code>	<code>—</code>	<code>&nnbsp;</code>	<code>—</code>
<code>&zwj;</code>	<code>◌</code>	<code>&zwnj;</code>	<code>◌</code>	<code>&middot;</code>	<code>◌</code>	<code>&bull;</code>	<code>•</code>
<code>&copyright;</code>	<code>©</code>	<code>&registered;</code>	<code>®</code>	<code>&trademark;</code>	<code>™</code>	<code>&dagger;</code>	<code>†</code>
<code>&Dagger;</code>	<code>‡</code>	<code>&numero;</code>	<code>№</code>	<code>&ordf;</code>	<code>ª</code>	<code>&ordm;</code>	<code>º</code>
<code>&para;</code>	<code>¶</code>	<code>&sect;</code>	<code>§</code>	<code>&deg;</code>	<code>°</code>	<code>&numero;</code>	<code>№</code>
<code>&permil;</code>	<code>‰</code>	<code>&brvbar;</code>	<code>̈́</code>	<code>&acute;</code>	<code>◌</code>	<code>&acutedbl;</code>	<code>◌</code>
<code>&grave;</code>	<code>◌</code>	<code>&breve;</code>	<code>◌</code>	<code>&caron;</code>	<code>◌</code>	<code>&cedilla;</code>	<code>◌</code>
<code>&circumflex;</code>	<code>◌</code>	<code>&diaeresis;</code>	<code>◌</code>	<code>&macron;</code>	<code>◌</code>	<code>&aa;</code>	<code>◌</code>
<code>&AA;</code>	<code>◌</code>	<code>&ae;</code>	<code>æ</code>	<code>&AE;</code>	<code>Æ</code>	<code>&auml;</code>	<code>ä</code>
<code>&Auml;</code>	<code>◌</code>	<code>&dh;</code>	<code>ð</code>	<code>&DH;</code>	<code>Ð</code>	<code>&dj;</code>	<code>đ</code>
<code>&DJ;</code>	<code>◌</code>	<code>&l;</code>	<code>ł</code>	<code>&L;</code>	<code>Ł</code>	<code>&ng;</code>	<code>ŋ</code>

&NG;	Ⓝ &o;	∅ &O;	∅ &oe;	œ
&OE;	Ⓔ ö	ö Ö	Ö &s;	ƒ
&ss;	ß &th;	þ &TH;	Ɔ ü	ü
Ü	Ü +	⊕ −	⊞ ×	×
÷	÷ ¹	¹ ²	² ³	³
&sqrt;	√ &increment;	△ &infty;	∞ ∑	Σ
±	± &bulletop;	▫ &partial;	∂ &neg;	◻
¤cy;	Ⓓ $	Ⓔ €	£ £s;	£
¥	¥ ¢	¢		

A.15 Elenco delle articolazioni

Nella logica interna di LilyPond, un’“articolazione” è un qualsiasi oggetto (diverso dalle dinamiche) che possa essere attaccato direttamente dopo un evento ritmico: note, accordi; perfino i silenzi e i salti, o il costrutto dell’accordo vuoto <> (vedi Sezione “Struttura dell’inserimento note” in *Manuale di Apprendimento*). Anche le legature di portamento, le diteggiature e le scritte testuali, tecnicamente, sono articolazioni, sebbene non siano mostrate qui.

Le liste seguenti comprendono quindi non solo i segni di articolazione, ma anche tutti gli altri segni del tipo di carattere Emmentaler che possono essere attaccati alle note (il modo in cui un accento è inserito come ‘c’\accent’ o ‘c’->’). Ogni esempio mostra il segno nelle sue due possibili posizioni verticali: rispettivamente *up* and *down*, così come la sua posizione predefinita (*neutral*). Vedi anche [Glifi dei segni], pagina 700, per un elenco di glifi più esaustivo, da usare col comando markup \musicglyph come è spiegato in [Notazione musicale nel blocco markup], pagina 270.

Articolazioni

\accent o ->



\espressivo



\marcato o -^



\portato or -_



\staccatissimo o -!



\staccato o -.



\tenuto o --



Ornamenti

`\prall`



`\prallup`



`\pralldown`



`\upprall`



`\downprall`



`\prallprall`



`\lineprall`



`\prallmordent`



`\mordent`



`\upmordent`



`\downmordent`



`\trill`



`\turn`



`\reverseturn`



`\slashturn`



`\haydnturn`



Punti coronati

`\veryshortfermata`



`\shortfermata`



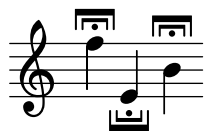
`\fermata`



`\longfermata`



`\verylongfermata`



`\henzeshortfermata`

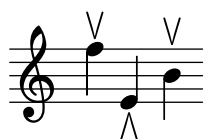


`\henzelongfermata`



Segni specifici per strumento

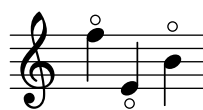
\upbow



\downbow



\flageolet



\open



\halfopen



\heel



\varheel



\heelcircle



\toe



\vartoe



\snappizzicato



\stopped or -+



\thumb



Segni di ripetizione

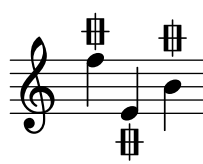
\segno



\coda

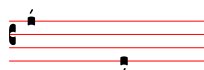


\varcoda

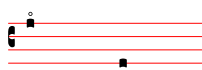


Segni antichi

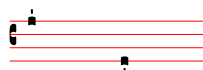
\accentus



\circculus



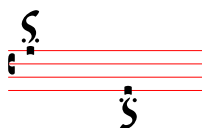
\ictus



\semicirculus



\signumcongruentiae



A.16 Elenco dei respiri

'chantquarterbar



'chanthalfbar



'chantfullbar



'chantdoubar



'comma



'varcomma



'caesura



'curvedcaesura



'tickmark



'spacer



A.17 Note percussive

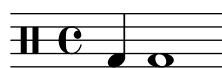
bassdrum

bd



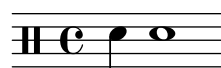
acousticbassdrum

bda



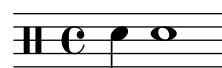
snare

sn



acousticsnare

sna



electricsnare

sne



lowfloortom

tomfl



highfloortom

tomfh



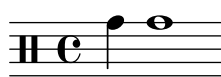
lowtom

toml



hightom

tomh



lowmidtom

tomml



himidtom

tommh



hihat

hh



closedhihat

hhc



openhihat

hho



halfopenhihat

hhho



pedalhihat

hhp



crashcymbal
cymc



crashcymbala
cymca



crashcymbalb
cymcb



ridecymbal
cymr



ridecymbala
cymra



ridecymbalb
cymrb



chinese cymbal
cymch



splashcymbal
cyms



ridebell
rb



cowbell
cb



hibongo
boh



openhibongo
boho



mutehibongo
bohbm



lobongo
bol



openlobongo
bolo



mutelobongo
bolm



hiconga
cgh



openhiconga
cgho



mutehiconga
cghm



loconga
cgl



openloconga
cglo



muteloconga
cglm



hitimbale
timh



lotimbale
timl



hiagogo
agh



loagogo
agl



sidestick
ss



hisidestick
ssh



losidestick
ssl



guiro
gui



shortguiro
guis



longguiro
guil



cabasa
cab



maracas
mar



shortwhistle
whs



longwhistle
whl



handclap
hc



tambourine
tamb



vibraslap
vibs



tamtam
tt



claves
cl



hiwoodblock
wbl



lowoodblock
wbl



opencuica
cuio



mutecuica
cuim



triangle
tri



opentriangle
trio



mutetriangle
trim



A.18 Valori predefiniti di outside-staff-priority

La tabella seguente mostra il valore predefinito di outside-staff-priority per tutti i grob fuori dal rigo. Gli oggetti con valori più bassi vengono posti più vicini al rigo.

Grob	Priority
AccidentalSuggestion	0
BassFigureAlignmentPositioning	25
MultiMeasureRestScript	40
TrillSpanner	50
BarNumber	100
LigatureBracket	200
DynamicLineSpanner	250
TextSpanner	350
OttavaBracket	400
MultiMeasureRestText	450
TextScript	450
CombineTextScript	475

InstrumentSwitch	500
VoltaBracketSpanner	600
MeasureCounter	750
MeasureSpanner	750
HorizontalBracket	800
SostenutoPedalLineSpanner	1000
SustainPedalLineSpanner	1000
UnaCordaPedalLineSpanner	1000
CenteredBarNumberLineSpanner	1200
TextMark	1250
MetronomeMark	1300
JumpScript	1350
CodaMark	1400
SegnoMark	1400
SectionLabel	1450
RehearsalMark	1500

A.19 Glossario tecnico

Un glossario dei termini tecnici e dei concetti usati internamente in LilyPond. Questi termini appaiono nei manuali, nelle mailing list e nel codice sorgente.

alist

Una lista di associazioni – **alist** per *association list* –, è una coppia Scheme che associa un valore a una parola chiave: (chiave . valore). Per esempio, in scm/lily.scm, la lista di associazioni “type-p-name-alist” associa alcuni tipi di predicato (come ly:music?) ai nomi (come “music”) in modo che gli errori relativi al controllo del tipo possano essere segnalati con un messaggio che includa il nome del tipo di predicato atteso.

callback

callback indica una routine, funzione o metodo il cui riferimento sia passato come argomento quando si richiama un'altra routine, permettendo così alla routine richiamata di invocarla. La tecnica fa sì che a un livello più basso del software si possa richiamare una funzione definita a un livello più alto. I “callback” sono molto utilizzati in LilyPond per far sì che il codice Scheme a livello utente possa definire quante azioni di basso livello sono eseguite.

closure

In Scheme, si parla di **closure** (chiusura) quando una funzione, di solito un'espressione lambda, viene passata come variabile. La chiusura contiene il codice della funzione più i riferimenti ai collegamenti lessicali delle variabili libere della funzione (ovvero quelle variabili usate nell'espressione ma definite al di fuori di essa). Quando questa funzione viene applicata a diversi argomenti successivamente, i collegamenti delle variabili libere che sono stati catturati nella chiusura vengono usati per ottenere i valori delle variabili libere da usare nel calcolo. Una caratteristica utile delle chiusure è la conservazione dei valori delle variabili interne tra un'invocazione e l'altra, facendo sì che uno stato possa essere mantenuto.

glyph

Un **glifo** è una particolare rappresentazione grafica di un carattere tipografico o una combinazione di due caratteri che formano una legatura. Un insieme di glifi con un solo stile e forma costituiscono un tipo di carattere (font), e un insieme di tipi di carattere con vari stili e dimensioni costituiscono una famiglia di caratteri tipografici.

Vedi anche

Guida alla notazione: Sezione 1.8.3 [Tipi di carattere], pagina 273, Sezione 3.3.3 [Caratteri speciali], pagina 531.

grob

Gli oggetti di LilyPond che rappresentano elementi della notazione nell’output – come teste di nota, gambi, legature di portamento e di valore, ditekgiature, chiavi, etc. – sono chiamati, in inglese, ‘Layout objects’ (‘Oggetti della formattazione’) o anche ‘GRaphical OBjects’ o **grobs** in forma breve. Sono rappresentati da istanze della classe Grob.

Vedi anche

Manuale di apprendimento: Sezione “Oggetti e interfacce” in *Manuale di Apprendimento*, Sezione “Proprietà degli oggetti di formattazione” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione A.22 [Convenzioni per i nomi], pagina 821.

Guida al funzionamento interno: Sezione “grob-interface” in *Guida al Funzionamento Interno*, Sezione “All layout objects” in *Guida al Funzionamento Interno*.

immutable

Un oggetto si dice **immutabile** – in inglese *immutable* – se il suo stato non può essere modificato dopo la sua creazione, in contrasto con un oggetto variabile, che può essere modificato dopo la sua creazione.

In LilyPond, le proprietà immutabili o condivise definiscono lo stile e il comportamento predefinito dei grob. Sono condivise tra molti oggetti. In apparente contraddizione col loro nome, possono essere modificate con `\override` e `\revert`.

Vedi anche

Guida alla notazione: [mutable], pagina 801.

interface

Le azioni e le proprietà comuni a un insieme di grob sono raggruppate in un oggetto chiamato grob-interface o semplicemente ‘interface’.

Vedi anche

Manuale di apprendimento: Sezione “Oggetti e interfacce” in *Manuale di Apprendimento*, Sezione “Proprietà presenti nelle interfacce” in *Manuale di Apprendimento*.

Guida alla notazione: Sezione A.22 [Convenzioni per i nomi], pagina 821, Sezione 5.2.2 [Interfacce di formattazione], pagina 625.

Guida al funzionamento interno: Sezione “Graphical Object Interfaces” in *Guida al Funzionamento Interno*.

lexer

Un **lexer** è un programma che converte una sequenza di caratteri in una sequenza di “token”, un processo chiamato *analisi lessicale*. Il lexer di LilyPond converte il flusso ottenuto da un file di input .ly in un flusso “tokenizzato” più adatto al passo successivo di elaborazione – il parsing (vedi [parser], pagina 801). Il lexer LilyPond viene costruito con Flex dal file `lexer/lily/lexer.ll`, che contiene le regole lessicali. Questo file fa parte del codice sorgente e non è incluso nell’installazione dei binari di LilyPond.

mutable

Si dice che un oggetto è **variabile** – *mutable* in inglese – se il suo stato può essere modificato dopo la sua creazione, in contrasto con un oggetto immutabile, il cui stato viene fissato al momento della sua creazione.

In LilyPond, le proprietà variabili contengono valori specifici di un grob. Di solito, le liste di altri oggetti o i risultati di calcoli sono salvati in proprietà variabili.

Vedi anche

Guida alla notazione: [immutable], pagina 800.

output-def

Un'istanza della classe `Output-def` contiene i metodi e le strutture dei dati associate con un blocco di output. Tali istanze vengono create per i blocchi `midi`, `layout` e `paper`.

parser

Un **analizzatore sintattico** – in inglese *parser* – analizza la sequenza di *token* prodotti da un *lexer* per determinare la sua struttura grammaticale, raggruppando i token progressivamente in gruppi più ampi in base a certe regole grammaticali. Se la sequenza di token è valida, il risultato finale è l'insieme dei token ordinati a albero, la cui radice è il simbolo iniziale della grammatica. Se ciò non può essere ottenuto, il file non è valido e viene generato un appropriato messaggio di errore. I gruppi sintattici e le regole che li definiscono nella sintassi di LilyPond sono definiti in `lily/parser.yy` e mostrati in Backus Normal Form (BNF) in `LilyPond grammar`. Questo file viene usato dal generatore di parser Bison per generare il parser durante la compilazione del programma. Fa parte del codice sorgente e non è incluso nell'installazione binaria di LilyPond.

parser variable

Si tratta di variabili definite direttamente in Scheme. Il loro uso da parte degli utente è fortemente scoraggiato, perché la semantica del loro raggio d'azione può creare confusione.

Se il valore di una simile variabile viene cambiato in un file `.ly`, la modifica è globale e, se non viene ripristinato esplicitamente, il nuovo valore viene mantenuto fino alla fine del file, agendo su blocchi `\score` successivi così come su file esterni aggiunti col comando `\include`. Ciò può portare a conseguenze non volute e in progetti complessi gli errori conseguenti possono essere difficili da individuare.

LilyPond usa le seguenti variabili dell'analizzatore sintattico:

- `afterGraceFraction`
- `musicQuotes`
- `mode`
- `output-count`
- `output-suffix`
- `partCombineListener`
- `pitchnames`
- `toplevel-bookparts`
- `toplevel-scores`
- `showLastLength`
- `showFirstLength`

prob

Le proprietà dell'oggetto – **prob** per PProperty Objects – sono istanze della classe Prob, una semplice classe per oggetti che hanno liste associative di proprietà variabili e invariabili e metodi per manipolarle. Le classi Music e Stream_event derivano da Prob. Vengono create istanze della classe Prob anche per conservare il contenuto formattato dei grob di un sistema e i blocchi dei titoli durante la formattazione della pagina.

smob

Gli oggetti Scheme – **Smobs** per ScheMe Objects – fanno parte del meccanismo con cui Guile esporta gli oggetti C e C++ in codice Scheme. In LilyPond, gli smob vengono creati dagli oggetti C++ attraverso delle macro. Esistono due tipi di oggetti smob: smob semplici, intesi per oggetti invariabili semplici come i numeri; e smob complessi, usati per oggetti aventi delle identità. Maggiori informazioni si trovano nei sorgenti di LilyPond e precisamente nel file `lily/includes/smob.hh`.

spanner

Gli **spanner** sono una classe di grob che non sono fissi orizzontalmente in un punto della partitura ma si estendono da un punto a un altro. Per esempio, le travature, le legature di valore e di portamento, così come le forcelle e le linee del rigo. Mentre i non-spanner possono solo dividersi al massimo in due pezzi visibili (per esempio, una chiave duplicata alla fine di un rigo e all'inizio di quello successivo), gli spanner si dividono in tanti pezzi quanti sono richiesti dalle loro estremità iniziale e finale (come i lunghi crescendo che si estendono su tre sistemi o le linee del rigo che si estendono sempre per l'intera partitura).

Tecnicamente, gli spanner sono definiti come dei grob aventi l'interfaccia `spanner-interface`; nella parte C++ di LilyPond, sono istanze della sottoclasse `Spanner` di `Grob`. I confini sinistro e destro di uno spanner possono essere recuperati e impostati usando `ly:spanner-bound` e `ly:spanner-set-bound!`, rispettivamente. I confini sono sempre degli elementi. Il genitore `X` di uno spanner ha poco senso musicale, ma viene solitamente impostato sul confine sinistro.

Vedi anche

Guida alla notazione: Sezione 5.4.6 [Estensori], pagina 645.

Guida al funzionamento interno: Sezione “All layout objects” in *Guida al Funzionamento Interno*, Sezione “spanner-interface” in *Guida al Funzionamento Interno*.

stencil

Un'istanza della classe **stencil** contiene l'informazione necessaria per stampare un oggetto tipografico. È un semplice smob che contiene un riquadro che definisce l'estensione verticale e orizzontale dell'oggetto, e un'espressione Scheme che stamperà l'oggetto quando esaminata. Gli stencil possono essere combinati per formare stencil più complessi, definiti da una gerarchia di espressioni Scheme degli stencil che li compongono.

La proprietà `stencil`, che connette un grob al suo stencil, è definita nell'interfaccia `grob-interface`.

Vedi anche

Guida al funzionamento interno: Sezione “grob-interface” in *Guida al Funzionamento Interno*.

A.20 Funzioni musicali disponibili

```
\absolute music (music) ⇒ music
Make music absolute.
```

This does not actually change the music itself but rather hides it from surrounding `\relative` and `\fixed` commands.

`\acciaccatura music (music) ⇒ music`

Create an acciaccatura from *music*.

`\accidentalStyle style (symbol list) ⇒ music`

Set accidental style to *style*.

style is a (predefined) symbol list like `piano-cautionary`; vedi [Alterazioni automatiche], pagina 29, for the available styles. If it is preceded by a context name, the settings are applied to that context (example: `Staff.piano-cautionary`). Otherwise, the context defaults to `Staff`, except for piano styles, which use `GrandStaff` as a context.

`\addChordShape key-symbol (symbol) tuning (pair) shape-definition (string or pair) ⇒ void`

Add *shape-definition* as a chord shape.

It gets added to the `chord-shape-table` hash with the key `(cons key-symbol tuning)`.

`\addInstrumentDefinition name (string) lst (list) ⇒ void`

Create instrument *name* with properties *lst*.

This function is deprecated.

`\addQuote name (string) music (music) ⇒ void`

Define *music* as a quotable music expression named *name*.

`\addToTagGroup tag-group (symbol list) more-tags (symbol list) ⇒ void`

Add *more-tags* to the tag group identified by *tag-group*.

`\after delta (duration) ev (music) mus (music) ⇒ music`

Add music *ev* with a delay of *delta* after the onset of *mus*.

ev is usually a post-event.

`\afterGrace [fraction (non-negative rational, fraction, or moment)] main (music) grace (music) ⇒ music`

Create *grace* as grace notes after a *main* music expression.

The musical position of the grace expression is after a given fraction of the main note's duration has passed. If optional argument *fraction* is omitted, the fraction value is taken from `afterGraceFraction`, defaulting to 3/4.

`\allowPageTurn ⇒ music`

Allow a page turn.

May be used at top level (i.e., between scores or markups), or inside a score.

`\allowVoltaHook bar (string) ⇒ void`

Allow the volta bracket hook being drawn over bar line *bar*.

`\alterBroken property (key list or symbol) arg (list) target (key list or music) ⇒ music`

Override *property* for pieces of broken spanner *target* with *arg*.

target may either be music in the form of a starting spanner event, or a symbol list of the form *Context.Grob* or just *Grob*. If *target* is in the form of a spanner event, *property* may also have the form *Grob.property* for specifying a directed tweak.

arg is a list of values, one for each broken piece.

`\ambitusAfter target (symbol) ⇒ music`

Move the ambitus after the break-align symbol *target*.

- `\appendToTag tag (symbol) more (music) music (music) ⇒ music`
 Append *more* to *music* tagged with *tag*.
 A post-event can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.
- `\appendToTagMarkup tag (symbol) more (markup) music (music) ⇒ music`
 Append *more* to every markup in *music* tagged with *tag*.
- `\applyContext proc (procedure) ⇒ music`
 Modify context properties with Scheme procedure *proc*.
- `\applyMusic func (procedure) music (music) ⇒ music`
 Apply procedure *func* to *music*.
- `\applyOutput target (symbol list or symbol) proc (procedure) ⇒ music`
 Apply function *proc* to every layout object matched by *target*.
target takes the form *Context* or *Context.Grob*.
- `\appoggiatura music (music) ⇒ music`
 Create an appoggiatura from *music*.
- `\approximatePitch note (music) ⇒ music`
 Indicate that the pitch of *note* approximates a pitch that cannot be known exactly, such as the highest note a singer can sing.
- `\arpeggioArrowDown ⇒ music`
 Let `\arpeggio` produce a wavy line with a down arrow.
- `\arpeggioArrowUp ⇒ music`
 Let `\arpeggio` produce a wavy line with an up arrow.
- `\arpeggioBracket ⇒ music`
 Let `\arpeggio` produce a square bracket.
 Note: For a bracket designating a non-arpeggiated chord, it is better to use `\nonArpeggiato` than to use `\arpeggio` and alter the appearance.
- `\arpeggioNormal ⇒ music`
 Let `\arpeggio` produce a wavy line without an arrow.
- `\arpeggioParenthesis ⇒ music`
 Let `\arpeggio` produce a vertical slur.
 Note: For a vertical slur designating a quasi-non-arpeggiated chord, it is better to use `\chordSlur` than to use `\arpeggio` and alter the appearance.
- `\arpeggioParenthesisDashed ⇒ music`
 Let `\arpeggio` produce a dashed vertical slur.
 Note: For a vertical slur designating a quasi-non-arpeggiated chord, it is better to use `\chordSlur` than to use `\arpeggio` and alter the appearance.
- `\assertBeamQuant l (pair) r (pair) ⇒ music`
 Testing function: check whether the beam quant *l* and *r* are correct.
- `\assertBeamSlope comp (procedure) ⇒ music`
 Testing function: check whether the slope of the beam is the same as *comp*.
- `\atLeft m (music) ⇒ post-event`
 Set side-axis to X and direction to LEFT for *mus*.
- `\atRight m (music) ⇒ post-event`
 Set side-axis to X and direction to RIGHT for *mus*.

`\augmentum expr (music) ⇒ music`

Add augmentum dots (*morae*) to Gregorian chant *expr*.

`\autoChange [pitch (pitch)] [clef-1 (context modification)] [clef-2 (context modification)]
music (music) ⇒ music`

Make voices for *music* that switch between staves automatically.

The optional argument *pitch* specifies where to switch staves; the default is *c'*.
Optional arguments *clef-1* and *clef-2* specify which clefs to use; this only works for
implicitly instantiated staves.

Example:

```
\autoChange d' \with { \clef alto } { g4 d' g' }
```

`\balloonGrobText grob-name (symbol) offset (pair of numbers) text (markup) ⇒ music`

Attach *text* to *grob-name* at offset *offset* (use like `\once`).

`\balloonText offset (pair of numbers) text (markup) ⇒ post-event`

Attach *text* at *offset* (use like `\tweak`).

`\bar type (string) ⇒ music`

Insert a bar line of type *type*, overriding any automatic bar lines.

`\barNumberCheck n (integer) ⇒ music`

Print a warning if the current bar number is not *n*.

`\beamExceptions music (music) ⇒ any type`

Set beam exceptions.

This function extracts a value suitable for setting `Timing.beamExceptions` from the
given pattern with explicit beams in *music*. A bar check ‘|’ has to be used between
bars of patterns in order to reset the timing.

`\bendAfter delta (real number) ⇒ post-event`

Create a fall or doit of pitch interval *delta*.

`\bendHold mus (music) ⇒ post-event`

Set `BendSpanner.style` to ‘hold’ for *mus*.

`\bendStartLevel idx (non-negative, exact integer) mus (music) ⇒ post-event`

Set `BendSpanner.details.successive-level` to *idx* for *mus*.

`\bookOutputName newfilename (string) ⇒ void`

Direct output for the current book block to *newfilename*.

This is equivalent to setting `output-filename` in the current book’s `\paper` block.

`\bookOutputSuffix newsuffix (string) ⇒ void`

Set the output file name suffix for the current book block to *newsuffix*.

This is equivalent to setting `output-suffix` in the current book’s `\paper` block.

`\breakAlignInsert [positions (vector)] symbol-to-move (symbol) sorting-symbol (symbol)
anchor-symbol (symbol) ⇒ music`

Re-order break-align symbols.

This function is for tasks like putting a clef change after the bar line, which would
be accomplished by `\breakAlignInsert clef` after `staff-bar`.

sorting-symbol can be before or after. Depending on its value, the break-align
symbol *symbol-to-move* gets moved directly before/after the symbol *anchor-symbol*
in the `break-align-orders` vector of `Score.BreakAlignment`.

Available break-align symbols are: *ambitus*, *breathing-sign*, *clef*, *cue-clef*,
cue-end-clef, *custos*, *key-cancellation*, *key-signature*, *left-edge*,
signum-repetitionis, *staff-bar*, *staff-ellipsis*, *time-signature*.

The optional argument *positions* determines whether the specified change should become visible at the beginning/the end/the middle of a line. Possible values are: `#all-visible` (default), `#begin-of-line-invisible`, `#center-invisible`, `#end-of-line-invisible`, `#begin-of-line-visible`, `#center-visible`, `#end-of-line-visible`, `#all-invisible`.

`\breathe` \Rightarrow music

Insert a breath mark.

`\caesura` \Rightarrow music

Insert a caesura.

`\chordRepeats` [*event-types* (list)] *music* (music) \Rightarrow music

Extend ‘q’ to also repeat articulation.

This function walks through *music*, putting the notes of the previous chord into repeat chords, as well as an optional list of *event-types* such as `#'(string-number-event)`.

`\clef` *type* (string) \Rightarrow music

Set the current clef to *type*.

`\codaMark` [*num* (non-negative, exact integer)] \Rightarrow music

Create a coda mark.

num may be 1 for the first mark, 2 for the second, etc., or it may be `\default` to use the next number in sequence automatically.

`\compressMMRests` *music* (music) \Rightarrow music

Convert empty bars to multi-measure rests in *music*.

`\contextPropertyCheck` *property-path* (symbol list or symbol) [*expected-value* (any type)] \Rightarrow music

Check that the context property identified by *property-path* is set to *expected-value* in that very context: being set in an enclosing context is insufficient. If *expected-value* is `\default`, check that the property is unset in that context.

If *property-path* does not name a context, the check occurs in the current context.

Print a warning if the requested context is not visible looking upward from the current context or if the state of the property in the requested context is unexpected.

Caveat: Like some other context-specific commands, a context given in *property-path* is ignored in certain cases, such as inside `\with`.

`\crossStaff` *notes* (music) \Rightarrow music

Create cross-staff stems for *notes*.

`\cueClef` *type* (string) \Rightarrow music

Set the current cue clef to *type*.

`\cueClefUnset` \Rightarrow music

Unset the current cue clef.

`\cueDuring` *what* (string) *dir* (direction) *main-music* (music) \Rightarrow music

Create a cue.

This function inserts the contents of quote *what* corresponding to *main-music*, in a CueVoice context called cue oriented by *dir*.

`\cueDuringWithClef` *what* (string) *dir* (direction) *clef* (string) *main-music* (music) \Rightarrow music

Create a cue with clef.

This function inserts the contents of quote *what* corresponding to *main-music*, in a CueVoice context called cue oriented by *dir* and using clef *clef*.

- `\deadNote` *note* (music) \Rightarrow music
 Print *note* with a cross-shaped note head.
- `\defineBarLine` *bar* (string) *glyph-list* (list) \Rightarrow void
 Define bar line settings for bar line *bar*.
 The list *glyph-list* must have three entries, defining substitute glyphs for the end of a line, the beginning of a line, and a span bar, respectively. The substitute glyphs may be either strings or Booleans: `#t` calls for the same value as *bar* and `#f` calls for no glyph.
- `\displayLilyMusic` [*port* (output port)] *music* (music) \Rightarrow music
 Write LilyPond's input representation of *music*.
 If *port* is omitted, the output defaults to the console (stdout).
- `\displayMusic` [*port* (output port)] *music* (music) \Rightarrow music
 Write the internal representation of *music*.
 If *port* is omitted, the output defaults to the console (stdout).
- `\displayScheme` [*port* (output port)] *expr* (any type) \Rightarrow any type
 Write the internal Scheme representation of *expr*.
 If *port* is omitted, the output defaults to the console (stdout).
- `\dropNote` *num* (integer) *music* (music) \Rightarrow music
 'Drop' the *num*-th note in each chord of *music*.
 This function moves the affected notes down (usually by an octave) to be lower than the other notes of the chord. The position in a chord is counted downwards from the top.
 The opposite function is `\raiseNote`.
- `\enablePerStaffTiming` \Rightarrow void
 Enable polymeter with unaligned measures.
 This function moves the timing management from Score to Staff-like contexts. This is done by removing the `Timing_translator` from Score, and adding it to all contexts having the Staff alias.
 Use this within an output definition.
- `\endSpanners` *music* (music) \Rightarrow music
 Terminate spanners.
 This function prematurely ends all spanners in *music* by inserting an end-spanner event at the end of the argument, without the need of specific end-spanner commands.
- `\eventChords` *music* (music) \Rightarrow music
 Compatibility function: Handle isolated rhythmic events in *music*.
 Use this to wrap `EventChord` around isolated rhythmic events occurring since version 2.15.28, after expanding repeat chords 'q'.
 Not needed for new code.
- `\featherDurations` *scale* (non-negative rational, fraction, or moment) *music* (music) \Rightarrow music
 Adjust feathered beam durations in *music* by *scale*.
- `\finger` *finger* (index or markup) \Rightarrow post-event
 Apply *finger* as a fingering indication.
- `\fixed` *pitch* (pitch) *music* (music) \Rightarrow music
 Use the octave of *pitch* as the default octave for *music*.

`\footnote [mark (markup)] offset (pair of numbers) footnote (markup) item (symbol list or music) ⇒ music`

Make the markup *footnote* a footnote on *item*.

The footnote is marked with a markup *mark* moved by *offset* with respect to the marked music.

If *mark* is not given or specified as `\default`, it is replaced by an automatically generated sequence number. If *item* is a symbol list of form *Grob* or *Context.Grob*, then grobs of that type are marked at the current time step in the given context (default Bottom).

If *item* is music, the music gets a footnote attached to a grob immediately attached to the event, like `\tweak` does. For attaching a footnote to an *indirectly* caused grob, write `\single\footnote`, use *item* to specify the grob, and follow it with the music to annotate.

Like with `\tweak`, if you use a footnote on a following post-event, the `\footnote` command itself needs to be attached to the preceding note or rest as a post-event with ‘-’.

`\grace music (music) ⇒ music`

Insert *music* as grace notes.

`\grobdescriptions descriptions (list) ⇒ any type`

Create a context modification from *descriptions*.

The argument is a list in the format of all-grob-descriptions.

`\harmonicByFret fret (number) music (music) ⇒ music`

Convert *music* into mixed harmonics.

The resulting notes resemble harmonics played on a fretted instrument by touching the strings at *fret*.

`\harmonicByRatio ratio (number) music (music) ⇒ music`

Convert *music* into mixed harmonics.

The resulting notes resemble harmonics played on a fretted instrument by touching the strings at the point given through *ratio*.

`\harmonicNote note (music) ⇒ music`

Print *note* with a diamond-shaped note head.

`\harmonicsOn ⇒ music`

Set the default note head style to a diamond-shaped style.

`\hide item (symbol list or music) ⇒ music`

Make *item* invisible while still retaining its dimensions.

If *item* is a symbol list of form *GrobName* or *Context.GrobName*, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

This function sets *item*’s transparent property to #t.

`\incipit [offset (number)] incipit-music (music) ⇒ music`

Output *incipit-music* as an incipit.

incipit-music is typeset within a *MensuralStaff* context; the result is positioned before the main staff (as part of an *InstrumentName* grob) to indicate the music’s original notation. The optional argument *offset* specifies how much the incipit should be moved to the left (default value is 0).

In the special case that *incipit-music* has the form `\new xxx ...`, where *xxx* is a context type not accepted by *MensuralStaff*, it is taken directly.

`\inherit-acceptability to (symbol) from (symbol) ⇒ void`

Make two contexts ‘compatible’.

When used in an output definition, modify all context definitions such that context *to* is accepted as a child by all contexts that also accept *from*.

`\initialContextFrom music (music) ⇒ music`

Enter the initial context of *music* and ignore the rest of it.

This is useful for prepending music while preserving the influence of the original music on the context.

Example:

```
{
  \initialContextFrom \originalMusic
  \prependedMusic
  \originalMusic
  \appendedMusic
}
```

`\inStaffSegno ⇒ music`

Put the segno variant varsegno at this position into the staff.

This is compatible with the repeat command.

`\instrumentSwitch name (string) ⇒ music`

Switch instrument to *name*.

name must have been predefined with function `\addInstrumentDefinition`.

This function is deprecated.

`\inversion around (pitch) to (pitch) music (music) ⇒ music`

Invert *music* about *around* and transpose from *around* to *to*.

`\invertChords num (integer) music (music) ⇒ music`

Invert any chords in *music* into their *num*-th position.

Chord inversions may be directed downwards using negative integers.

`\jump text (markup) ⇒ music`

Use *text* to mark a point of departure, e.g., ‘Gavotte I D.C.’.

`\keepWithTag tags (symbol list or symbol) music (music) ⇒ music`

Keep tagged music.

This function only includes elements of *music* that are tagged with one of the tags in *tags*. *tags* may be either a single symbol or a list of symbols.

Each tag may be declared as a member of at most one tag group (defined with `\tagGroup`). If none of a *music* element’s tags share a tag group with one of the specified *tags*, the element is retained.

`\key [tonic (pitch)] [pitch-alist (list of number pairs)] ⇒ music`

Set key to *tonic* and scale *pitch-alist*.

If both arguments are omitted (i.e., replaced by `\default`), just generate a `KeyChangeEvent`, which prints the current key signature again.

`\killCues music (music) ⇒ music`

Remove cue notes from *music*.

`\label label (symbol) ⇒ music`

Create *label* as a referable label.

The value stored in *label* is the page number, which can be extracted with the `\page-ref` markup command later on.

`\language language (string) ⇒ void`

Set note names for language *language*.

This function updates the Scheme variables *pitchnames* and *input-language*. The former is an alist that holds the pitches of all note names for the selected language; the latter holds the language name itself (as a symbol).

`\languageRestore ⇒ void`

Restore the previously-saved *pitchnames* alist.

`\languageSaveAndChange language (string) ⇒ void`

Save current *pitchnames* alist and change note names to *language*.

`\ligature music (music) ⇒ music`

Make a ligature from Gregorian Chant *music*.

This is equivalent to enclosing *music* with `\[` and `\]`.

`\magnifyMusic mag (positive number) music (music) ⇒ music`

Magnify the size of *music* by factor *mag*.

This doesn't change the staff size. Stems, beams, slurs, ties, and horizontal spacing are adjusted automatically.

`\magnifyStaff mag (positive number) ⇒ music`

Change the staff size by factor *mag*.

This adjusts notation size and horizontal spacing automatically.

`\makeClusters arg (music) ⇒ music`

Display chords in *arg* as clusters.

`\makeDefaultStringTuning symbol (symbol) pitches (list) ⇒ void`

Define string tuning *symbol* by a list of *pitches*.

symbol also gets registered in *defaultStringTunings* for documentation purposes.

`\mark [label (index or markup)] ⇒ music`

Create a rehearsal mark.

If *label* is an integer, create the rehearsal mark for the given sequence number. If *label* is `\default`, create the next sequential rehearsal mark. If *label* is markup, use it for the mark.

`\markupMap path (symbol list or symbol) markupfun (markup-function) music (music) ⇒ music`

Apply *markupfun* to property *path* in *music*.

Argument *path* is either of the form *property* or *MusicExpression.property*. If *MusicExpression* is not given, *markupfun* gets applied to all properties called *property*, otherwise it is restricted to *MusicExpression* events. If *property* is not a markup, it is ignored.

In the following example, both the tempo indication and the bowing instruction are printed in red. If you replace text with `TempoChangeEvent.text`, only the tempo indication changes the color.

```
\markupMap
  text
  \markup \with-color #red \etc
  { \tempo "Largo" g'2_"arco" c'' }
```

`\modalInversion around (pitch) to (pitch) scale (music) music (music) ⇒ music`

Invert *music* about *around* using *scale* and transpose from *around* to *to*.

`\modalTranspose` *from* (pitch) *to* (pitch) *scale* (music) *music* (music) \Rightarrow music
Transpose *music* from *pitch from* to *pitch to* using *scale*.

`\musicLength` *music* (music) \Rightarrow any type
Return the duration of *music* as a `ly:moment`.

Examples:

`\musicLength 8` \Rightarrow `#{ly:make-moment 1/8}`
`\musicLength {8. 8 8}` \Rightarrow `#{ly:make-moment 7/16}`

`\musicMap` *proc* (procedure) *mus* (music) \Rightarrow music
Apply *proc* to *mus* and all of the music it contains.

`\noPageBreak` \Rightarrow music
Forbid a page break.
May be used at top level (i.e., between scores or markups), or inside a score.

`\noPageTurn` \Rightarrow music
Forbid a page turn.
May be used at top level (i.e., between scores or markups), or inside a score.

`\octaveCheck` *pitch* (pitch) \Rightarrow music
Do an octave check.
This prints a warning if the interval between the previous note and *pitch* is not within a fourth.

`\offset` *property* (symbol list or symbol) *offsets* (any type) *item* (key list or music) \Rightarrow music
Offset the default value of *property* of *item* by *offsets*.
If *item* is a string, the result is an override for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.

`\omit` *item* (symbol list or music) \Rightarrow music
Omit *item* without taking up space.
If *item* is a symbol list of form *GrobName* or *Context.GrobName*, the result is an override for the grob name specified by it. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied to it.
This function sets *item*'s stencil property to `#f`.

`\once` *music* (music) \Rightarrow music
Set property once to `#t` on all layout instruction events in *music*.

`\ottava` *octave* (integer) \Rightarrow music
Set the octavation to *octave*.
A positive value *n* indicates *n* octaves higher; a negative value *n* octaves lower, and value 0 means no octavation.

`\overrideProperty` *grob-property-path* (list of indices or symbols) *value* (any type) \Rightarrow music
Set the grob property specified by *grob-property-path* to *value*.
grob-property-path is a symbol list of the form *Context.GrobName.property* or *GrobName.property*, possibly with subproperties given as well.
As opposed to `\override`, which overrides the context-dependent defaults with which a grob is created, this command uses `Output_property_engraver` at the grob acknowledge stage. This may be necessary for overriding values set after the initial grob creation.

`\overrideTimeSignatureSettings` *time-signature* (boolean-or-fraction) *beat-base* (positive exact rational, fraction (as pair), moment, or +inf.0) *structure* (a number list or a list of them) *beam-exceptions* (list) \Rightarrow music

Override time signature settings.

This function sets the `timeSignatureSettings` entry for *time-signature* to use *beat-base*, *structure*, and *beam-exceptions*.

beat-base and *beam-exceptions* are used as-is for `beatBase` and `beamExceptions`.

structure is used to derive values for `beatStructure` and `submeasureStructure` as described for the `\time` command.

`\pageBreak` \Rightarrow music

Force a page break.

May be used at top-level (i.e., between scores or markups), or inside a score.

`\pageTurn` \Rightarrow music

Force a page turn.

May be used at top-level (i.e., between scores or markups), or inside a score.

`\palmMute` *note* (music) \Rightarrow music

Print *note* with a triangle-shaped note head.

`\palmMuteOn` \Rightarrow music

Set the default note head style to a triangle-shaped style.

`\parallelMusic` *voice-ids* (list) *music* (music) \Rightarrow void

Define parallel music sequences.

Within *music*, parallel music sequences are separated by ‘|’ characters. The sequences are assigned to the LilyPond music identifiers provided in *voice-ids*.

For example, this code

```
\parallelMusic A,B,C {
  c c | d d | e e |
  d d | e e | f f |
}
```

is equivalent to

```
A = { c c | d d }
B = { d d | e e }
C = { e e | f f }
```

The last bar checks in a sequence are not copied to the result in order to facilitate ending the last entry at non-bar boundaries.

`\parenthesize` *arg* (symbol list or music) \Rightarrow music

Tag *arg* to be parenthesized.

arg may be either a music event or a grob path.

`\partCombine` [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) \Rightarrow music

Combine two parts into a single staff.

This takes the music in *part1* and *part2* and returns a music expression containing simultaneous Voice contexts (called one for the upper and two for the lower voice). Where appropriate, *part1* and *part2* are combined into a single voice (called shared or solo, depending on context).

Optional argument *chord-range* is a pair (*start* . *stop*) that defines the range in which the two voices are printed as chords (or unison); the default value is (0 . 8), which means that intervals up to and including a ninth are unified.

`\partCombineDown` [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) \Rightarrow music

Combine two parts into a single staff with all stems downwards.

See function `\partCombine` for details.

`\partCombineForce` [*type* (symbol)] \Rightarrow music

Override the part-combiner mode with *type*.

The following table gives the possible values for *type*, together with the corresponding shorthand functions.

<code>apart</code>	<code>\partCombineApart</code>
<code>chords</code>	<code>\partCombineChords</code>
<code>unisono</code>	<code>\partCombineUnisono</code>
<code>solo1</code>	<code>\partCombineSoloI</code>
<code>solo2</code>	<code>\partCombineSoloII</code>
<code>\default</code>	<code>\partCombineAutomatic</code>

`\partCombineUp` [*chord-range* (pair of numbers)] *part1* (music) *part2* (music) \Rightarrow music

Combine two parts into a single staff with all stems upwards.

See function `\partCombine` for details.

`\partial` *dur* (duration) \Rightarrow music

Adjust the measure position to end the current measure at *dur* past the point of use. As a special case, when used at the start, create an anacrusis before the first measure.

`\phrasingSlurDashPattern` *dash-fraction* (number) *dash-period* (number) \Rightarrow music

Set up a custom dash pattern style for phrasing slurs.

dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

More complex patterns can be achieved by directly manipulating the `PhrasingSlur`.dash-definition property.

`\pitchedTrill` *main-note* (music) *secondary-note* (music) \Rightarrow music

Print a pitched trill.

main-note is the main note of the trill; *secondary-note* gets printed as a stemless note head in parentheses.

`\pointAndClickOff` \Rightarrow void

Suppress links to LilyPond source code in music output.

`\pointAndClickOn` \Rightarrow void

Generate links to LilyPond source code in music output.

This enables the creation of code in a PDF or SVG output file to reference the originating LilyPond source code (i.e., file name, line number, and column). This is helpful when developing a score; however, the output file becomes much larger.

`\pointAndClickTypes` *types* (symbol list or symbol) \Rightarrow void

Generate point-and-click info for music of type *types* only.

types is a single music expression (such as `#'note-event`) or a list of music expressions.

`\polymetric` [*time-signature-command* (music)] \Rightarrow music

Apply *time-signature-command* to the local context rather than to Timing.

Example:

```
\context Staff \polymetric \time 6/8
```

The local measure must align with the reference measure defined in `Timing`. So that explicit changes to `Timing.measureLength` (e.g., for ad-hoc irregular measures) remain visible, this command does not set the `measureLength` property in the local context.

A nominally incompatible local measure can be fitted to the controlling `Timing` measure by applying `\scaleDurations` to this command. That adjusts the value of the internal property `meterScalingFactor` in the local context.

To unset the local properties and resume using the values from the `Timing` context, use `\polymetric \default`.

`\popContextProperty path` (list of indices or symbols) \Rightarrow music

Pop value of context property *path* from stack and set it.

This is the opposite to function `\pushContextProperty`.

`\preBend mus` (music) \Rightarrow post-event

Set `BendSpanner.style` to 'pre-bend for *mus*.

`\preBendHold mus` (music) \Rightarrow post-event

Set `BendSpanner.style` to 'pre-bend-hold for *mus*.

`\propertyOverride grob-property-path` (list of indices or symbols) *value* (any type) \Rightarrow music

Set the grob property specified by *grob-property-path* to *value*.

grob-property-path is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\override` command.

`\propertyRevert grob-property-path` (list of indices or symbols) \Rightarrow music

Revert the grob property specified by *grob-property-path* to its previous value.

grob-property-path is a symbol list of the form `Context.GrobName.property` or `GrobName.property`, possibly with subproperties given as well. This music function is mostly intended for use from Scheme as a substitute for the built-in `\revert` command.

`\propertySet property-path` (symbol list or symbol) *value* (any type) \Rightarrow music

Set the context property specified by *property-path* to *value*.

This music function is mostly intended for use from Scheme as a substitute for the built-in `\set` command.

`\propertyTweak prop` (key list or symbol) *value* (any type) *item* (key list or music) \Rightarrow music

Add a tweak to *item*, usually music.

This function sets the value of property *prop* to *value*; it generally behaves like `\tweak` but will turn into an `\override` when *item* is a symbol list. In that case, *item* specifies the grob path to override. This is mainly useful when using `\propertyTweak` as a component for building other functions like `\omit`. It is not the default behavior for `\tweak` since many input strings in `\lyricmode` can serve equally as music or as symbols, which causes surprising behavior when tweaking lyrics using the less specific semantics of `\propertyTweak`.

prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

`\propertyUnset property-path` (symbol list or symbol) \Rightarrow music

Unset the context property specified by *property-path*.

This music function is mostly intended for use from Scheme as a substitute for the built-in `\unset` command.

`\pushContextProperty` *path* (list of indices or symbols) \Rightarrow music

Push the current value of context property *path* to a stack.

The property can later be restored to the saved value with function `\popContextProperty`.

`\pushToTag` *tag* (symbol) *more* (music) *music* (music) \Rightarrow music

Add *more* to the front of *music* tagged with *tag*.

A post-event can be added to the articulations of rhythmic events or chords; other expressions may be added to chords, sequential or simultaneous music.

`\pushToTagMarkup` *tag* (symbol) *more* (markup) *music* (music) \Rightarrow music

Prepend *more* to every markup in *music* tagged with *tag*.

`\quoteDuring` *what* (string) *main-music* (music) \Rightarrow music

Indicate a section of music to be quoted.

what indicates the name of the quoted voice, as specified in an `\addQuote` command. *main-music* is used to indicate the length of music to be quoted; it usually contains spacers or multi-measure rests.

`\raiseNote` *num* (integer) *music* (music) \Rightarrow music

‘Raise’ the *num*-th note in each chord of *music*.

This function moves the affected notes up (usually by an octave) to be higher than the other notes of the chord. The position in a chord is counted upwards from the bottom.

The opposite function is `\dropNote`.

`\reduceChords` *music* (music) \Rightarrow music

Reduce chords contained in *music* to single notes.

This is intended mainly for reusing music in a `RhythmicStaff` context. It does not reduce simultaneous music.

`\relative` [*pitch* (pitch)] *music* (music) \Rightarrow music

Make *music* relative to *pitch*.

If *pitch* is omitted, the first note in *music* is given in absolute pitch.

`\removeFromTagGroup` *tag-group* (symbol list) *tags* (symbol list) \Rightarrow void

Remove *tags* from the tag group identified by *tag-group*.

`\removeWithTag` *tags* (symbol list or symbol) *music* (music) \Rightarrow music

Remove elements of *music* that are tagged with one of the tags in *tags*.

tags may be either a single symbol or a list of symbols.

`\replaceWithTag` *tag* (symbol) *replacement* (music) *music* (music) \Rightarrow music

Replace tagged elements in *music*.

Everything tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

`\replaceWithTagMarkup` *tag* (symbol) *replacement* (markup) *music* (music) \Rightarrow music

Replace tagged markup in *music*.

All markup tagged with *tag* (including the tagging itself) gets replaced with *replacement*.

`\resetRelativeOctave` *pitch* (pitch) \Rightarrow music

Set the octave inside a `\relative` section to *pitch*.

`\resetTagGroup` *tag-group* (symbol list) \Rightarrow void

Reset the tag group equal to *tag-group*.

`\resetTagGroups` \Rightarrow void

Reset all tag groups previously created with `\tagGroup`.

`\responsum` *music* (music) \Rightarrow music

Prepend character U+211F (RESPONSE) to the lyrics represented by *music*.

`\retrograde` *music* (music) \Rightarrow music

Return *music* in reverse order.

`\revertTimeSignatureSettings` *time-signature* (pair) \Rightarrow music

Revert `timeSignatureSettings` for time signatures equal to *time-signature*.

`\rightHandFinger` *finger* (index or markup) \Rightarrow post-event

Apply *finger* as a right-hand fingering indication.

`\scaleDurations` *fraction* (non-negative rational, fraction, or moment) *music* (music) \Rightarrow music

Multiply the duration of events in *music* by *fraction*.

`\sectionLabel` *text* (markup) \Rightarrow music

Mark the beginning of a named passage with *text*, e.g., “Coda”.

This is well suited for use at a section division created with `\section`, but it does not imply `\section` and may be used alone.

`\segnoMark` [*num* (non-negative, exact integer)] \Rightarrow music

Create a segno mark (or bar line).

num may be 1 for the first segno, 2 for the second, etc., or it may be `\default` to use the next number in sequence automatically.

If the `segnoStyle` context property is 'bar-line', a segno bar line is created instead of a segno mark.

`\shape` *offsets* (list) *item* (key list or music) \Rightarrow music

Offset control points of *item* by *offsets*.

offsets is a list of number pairs (*x* . *y*) or a list of such lists. Each pair represents an offset to a control point. The ‘y’ value of each pair is scaled by staff space.

If *item* is a string, the result is `\once\override` for the specified grob type. If *item* is a music expression, the result is the same music expression with an appropriate tweak applied.

`\shiftDurations` *dur* (integer) *dots* (integer) *arg* (music) \Rightarrow music

Change duration of *arg*.

This function walks over all durations and dot counts in *arg*, adding *dur* to the durations and *dots* to the dot counts.

`\single overrides` (music) *music* (music) \Rightarrow music

Convert *overrides* to tweaks and apply them to *music*.

This does not convert `\revert`, `\set` or `\unset`.

`\skip` *arg* (duration-or-music) \Rightarrow music

Skip over *arg*, which may be music or a duration.

`\slashedGrace` *music* (music) \Rightarrow music

Create slashed graces from *music*.

This produces slashes through stems, but no slur.

`\slurDashPattern` *dash-fraction* (number) *dash-period* (number) \Rightarrow music

Set up a custom dash pattern style for slurs.

dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

More complex patterns can be achieved by directly manipulating the `Slur` `.dash-definition` property.

`\staffHighlight color (color) ⇒ music`

Start a highlight with color *color*.

`\storePredefinedDiagram fretboard-table (hash table) chord (music) tuning (pair)`

diagram-definition (string or pair) ⇒ void

Add a predefined fret diagram to *fretboard-table*.

It is defined by *diagram-definition* for the chord pitches *chord* and the string tuning *tuning*.

`\stringTuning chord (music) ⇒ any type`

Convert *chord* to a string tuning.

chord must be in absolute pitches and should have the highest string number (generally the lowest pitch) first.

`\styledNoteHeads style (symbol) heads (symbol list or symbol) music (music) ⇒ music`

Set *heads* in *music* to *style*.

`\tabChordRepeats [event-types (list)] music (music) ⇒ music`

Extend ‘q’ to also repeat string and fingering information.

This function walks through *music* putting the notes, fingerings and string numbers of the previous chord into repeat chords, as well as an optional list of *event-types* such as #‘(articulation-event)’.

`\tabChordRepetition ⇒ void`

Include the string and fingering information in a chord repetition.

This function is deprecated; use `\tabChordRepeats` instead.

`\tag tags (symbol list or symbol) music (music) ⇒ music`

Tag *music* with *tags*.

This function adds the single symbol or symbol list *tags* to the tags property of *music* and returns the result.

`\tagGroup tags (symbol list) ⇒ void`

Define a tag group comprising the symbols in the symbol list *tags*.

Tag groups must not overlap.

`\tagGroupRef tags (symbol list) ⇒ any type`

Define a tag group comprising the symbols in the symbol list *tags*.

It returns the created tag group for further reference.

`\temporary music (music) ⇒ music`

Make `\override` reversible with `\revert`.

This function makes any `\override` in *music* replace an existing grob property value only temporarily, restoring the old value when a corresponding `\revert` is executed. This is achieved by clearing the pop-first property normally set on `\overrides`.

An `\override/\revert` sequence created by using `\temporary` and `\undo` on the same music containing overrides will cancel out perfectly or cause a warning.

Non-property-related music is ignored, warnings are generated for any property-changing music that isn’t an `\override`.

`\textEndMark text` (markup) \Rightarrow music

Create a right-aligned text mark using *text*.

`\textMark text` (markup) \Rightarrow music

Create a (left-aligned) text mark using *text*.

`\tieDashPattern dash-fraction` (number) *dash-period* (number) \Rightarrow music

Set up a custom dash pattern style for ties.

dash-fraction gives the size of one dash relative to *dash-period*; *dash-period* is the length of one dash plus one space. LilyPond adjusts *dash-period* to produce symmetrical output.

More complex patterns can be achieved by directly manipulating the `Tie` `.dash-definition` property.

`\time [structure` (a number list or a list of them)] *time-sig* (time signature) \Rightarrow music

Set the time signature to *time-sig*.

When *structure* is a plain list of numbers, it is used as-is for `beatStructure`, and `submeasureStructure` is left at the default.

When *structure* is a list of lists, `beatStructure` is derived by discarding the grouping, and `submeasureStructure` is derived by summing each group. For example, a *structure* of `'((1 2) (4 8))` yields a `beatStructure` of `'(1 2 4 8)` and a `submeasureStructure` of `'(3 12)`.

time-sig may be a fraction, e.g., $3/4$.

time-sig may also describe a complex time signature as a Scheme expression. Fractions are represented as pairs, (*numerator* . *denominator*), where the denominator is always a number. The numerator is one number or a list of two or more numbers. A list represents concatenation.

For example, a time signature of $(3+1)/8 + 2/4$ can be created with `\time #'(((3 1) . 8) (2 . 4))`

`\timeAbbrev time-sig` (pair) \Rightarrow music

Set the time signature to *time-sig*.

This is like `\time time-sig`, except that it allows abbreviating fractions as lists. For example, a time signature of $(3+1)/8 + 2/4$ can be created with `\timeAbbrev #'((3 1 8) (2 4))`, and a time signature of $(3+2)/8$ with either `\timeAbbrev #'((3 2 8))` or the shorter version `\timeAbbrev 3,2,8`.

This is for backward compatibility. Using `\time` instead is recommended.

`\times fraction` (fraction, as pair) *music* (music) \Rightarrow music

Scale *music* in time by *fraction*.

`\tocItem [label` (symbol list or symbol)] *text* (markup) \Rightarrow music

Add *text* as an entry to the table of contents.

This uses the `tocItemMarkup` paper variable markup for formatting and assigns it to *label* if one is provided. If a hierarchy of labels is given, make the current item a child of the corresponding objects.

`\transpose from` (pitch) *to* (pitch) *music* (music) \Rightarrow music

Transpose *music* from pitch *from* to pitch *to*.

`\transposedCueDuring what` (string) *dir* (direction) *pitch* (pitch) *main-music* (music) \Rightarrow music

Create a transposed cue.

This function inserts notes from the part *what* into a CueVoice context called *cue*, using the transposition defined by *pitch*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`\transposition pitch (pitch) ⇒ music`

Set instrument transposition to *pitch*.

`\tuplet ratio (fraction, as pair) [tuplet-span (duration)] music (music) ⇒ music`

Scale the given *music* to tuplets.

ratio is a fraction that specifies how many notes are played in place of the nominal value: it will be 3/2 for triplets, namely three notes being played in place of two.

If the optional duration *tuplet-span* is specified, it is used instead of `tupletSpannerDuration` for grouping the tuplets. For example,

```
\tuplet 3/2 4 { c8 c c c c c }
```

results in two groups of three tuplets, each group lasting for a quarter note.

`\tupletSpan [tuplet-span (duration)] ⇒ music`

Set `tupletSpannerDuration` to the duration *tuplet-span*.

This context property is the length into which `\tuplet` without an explicit `tuplet span` argument of its own will group its tuplets. To revert to the default of not subdividing the contents of a `\tuplet` command without an explicit `tuplet span` argument, use

```
\tupletSpan \default
```

`\tweak prop (key list or symbol) value (any type) music (music) ⇒ music`

Add a tweak to *music*.

Layout objects created by *music* get their property *prop* set to *value*. If *prop* has the form *Grob.property*, like with

```
\tweak Accidental.color #red cis'
```

an indirectly created grob (Accidental is caused by NoteHead) can be tweaked; otherwise only directly created grobs are affected.

prop can contain additional elements in which case a nested property (inside of an alist) is tweaked.

If *music* is an event-chord, every contained rhythmic-event is tweaked instead.

`\undo music (music) ⇒ music`

Convert `\override` and `\set` in *music* to `\revert` and `\unset`, respectively.

Any reverts and unsets already in *music* cause a warning. Non-property-related music is ignored.

`\unfolded music (music) ⇒ music`

Mask *music* until the innermost enclosing repeat is unfolded.

`\unfoldRepeats [types (symbol list or symbol)] music (music) ⇒ music`

Unfold `\repeat`.

This forces `\repeat volta`, `\repeat tremolo` or `\repeat percent` commands in *music* to be interpreted as `\repeat unfold`, if specified in the optional symbol-list *types*. The default for *types* is an empty list, which forces any of those commands in *music* to be interpreted as `\repeat unfold`. Possible entries are *volta*, *tremolo* or *percent*. Multiple entries are possible.

`\versus music (music) ⇒ music`

Prepend character U+2123 (VERSICLE) to the lyrics represented by *music*.

`\voices ids (list of indices or symbols) music (music) ⇒ music`

Specify voice order in simultaneous music.

This takes the key list *ids* of numbers (indicating the use of ‘\voiceOne’...) or symbols (indicating voice names, typically converted from strings by argument list processing) and assign the following \-separated music in *music* to contexts according to that list. Named rather than numbered contexts can be used for continuing one voice (for the sake of spanners and lyrics), usually requiring a \voiceOne-style override at the beginning of the passage and a \oneVoice override at its end.

The default

```
<< ... \ \ ... \ \ ... >>
```

construct would correspond to

```
\voices 1,2,3 << ... \ \ ... \ \ ... >>
```

`\void arg (any type) ⇒ void`

Accept a Scheme argument *arg* and return a void expression.

Use this if you want to have a Scheme expression evaluated because of its side effects but its return value being ignored.

`\volta volta-numbers (number list) music (music) ⇒ music`

Mark *music* as being limited to the volte given in *volta-numbers*.

This gets used when the innermost enclosing repeat is unfolded. Volta number begins at 1 and increases by 1 with each repetition.

`\vshape offsets (list) item (key list or music) ⇒ music`

Like \shape, but additionally show control points for ease of tweaking.

`\withMusicProperty sym (symbol) val (any type) music (music) ⇒ music`

Set music property *sym* to *val* in *music*.

`\withRelativeDir file-name (string) ⇒ any type`

Prepend directory of current input file to string *file-name*.

Use this for markup commands that include files, and where such files should be found relative to the input file. Example:

```
\markup { \image #X #3 \withRelativeDir "test.png" }
```

`\xNote note (music) ⇒ music`

Print *note* with a cross-shaped note head.

`\= id (index or symbol) event (post-event) ⇒ post-event`

Assign an ID to a spanner or an item.

This sets the spanner-id or id property of *event* to the given *id*, which is a non-negative integer or a symbol.

For spanners this can be used to tell LilyPond how to connect overlapping or parallel slurs or phrasing slurs within a single Voice context.

```
\fixed c' { c\=1( d\=2( e\=1) f\=2) }
```



For items this can be used, for example, to tell LilyPond how to connect a FingerGlideSpanner with non-matching fingers.

```
\fixed c' { c\glide \= #'foo -1 d\= #'foo -2 }
```



`\% count (number) music (music) ⇒ music`

This is the same as `\repeat percent count music`.

`* count (number) music (music) ⇒ music`

This is the same as `\repeat unfold count music`.

A.21 Identificatori delle modifiche di contesto

I seguenti comandi possono essere usati come modificatori di contesto all'interno di un blocco `\layout` o `\with`.

`\EnableGregorianDivisiones`

Configure division commands such as `\section` to create `Divisio` grobs rather than `BarLine` grobs. This does not affect measure bar lines or the properties of the grobs themselves.

- Sets translator property `caesuraTypeTransform` to `caesura-to-divisio`.
- Sets translator property `doubleRepeatBarType` to `'()`.
- Sets translator property `endRepeatBarType` to `'()`.
- Sets translator property `fineBarType` to `""`.
- Sets translator property `sectionBarType` to `""`.
- Sets translator property `startRepeatBarType` to `'()`.
- Sets translator property `underlyingRepeatBarType` to `""`.
- Sets translator property `doubleRepeatSegnoBarType` to `"S-||"`.
- Sets translator property `endRepeatSegnoBarType` to `"S-||"`.
- Sets translator property `fineSegnoBarType` to `"S-||"`.
- Sets translator property `fineStartRepeatSegnoBarType` to `"S-||"`.
- Sets translator property `segnoBarType` to `"S-||"`.
- Sets translator property `startRepeatSegnoBarType` to `"S-||"`.

`\RemoveAllEmptyStaves`

Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`, including those in the first system.

- Sets grob property `remove-empty` in Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno* to `#t`.
- Sets grob property `remove-first` in Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno* to `#t`.

`\RemoveEmptyStaves`

Remove staves which are considered to be empty according to the list of interfaces set by `keepAliveInterfaces`.

- Sets grob property `remove-empty` in Sezione “VerticalAxisGroup” in *Guida al Funzionamento Interno* to `#t`.

A.22 Convenzioni per i nomi

Si usano delle convenzioni per i nomi in modo che sia più facile avere in mente la natura di un oggetto.

Object/property type	Convenzione del nome	Esempi
Parole chiave	aaaa	\score, \addlyrics
Funzioni musicali e scorciatoie	aaaa o aaaaAaaaAaaa	\time, \stemUp
Funzioni di markup	aaaa o aaaa-aaaa-aaaa	\bold, \with-dimensions
Variabili del foglio	aaaa o aaaa-aaaa-aaaa	indent, page-breaking
Espressioni musicali	Aaaa o AaaaAaaaAaaa	NoteEvent, SequentialMusic
Classi musicali	aaaa o aaaa-aaaa-aaaa	note-event, rhythmic-event
Proprietà musicali	aaaa o aaaa-aaaa-aaaa	force-accidental, iterator-ctor
Contesti	Aaaa o AaaaAaaaAaaa	Staff, GrandStaff
Proprietà di contesto	aaaa o aaaaAaaaAaaa	stanza, instrumentName
Incisori	Aaaa_engraver o Aaaa_aaaa_aaaa_engraver	Clef_engraver, Note_heads_engraver
Oggetti di formattazione	Aaaa o AaaaAaaaAaaa	Slur, NoteHead
Proprietà degli oggetti di formattazione	aaa o aaa-aaa-aaa	direction, beam-thickness
Interfacce	aaa-aaa-interface	grob-interface, break-aligned-interface
Funzioni Scheme definite in Guile	aaaa o aaaa-aaaa-aaaa	map, for-each
Funzioni Scheme definite in LilyPond	ly:aaaa o ly:aaaa-aaaa-aaaa o aaaa o aaaa-aaaa-aaaa ³	ly:round-filled-box, music-map

A.23 Tipi di predicati predefiniti

Predicates return #t (true) if their argument is of the named type and #f (false) if it isn't.

A.23.1 R5RS primary predicates

Primary predicates can be applied to any expression. They can be used on their own as predicates for LilyPond functions. The predicates here are part of the Scheme standard R5RS.

Type predicate	Description
boolean?	boolean
char?	character
complex?	complex number
eof-object?	end-of-file object
input-port?	input port
integer?	integer
list?	list (use cheap-list? for faster processing)
null?	null
number?	number
output-port?	output port
pair?	pair
port?	port
procedure?	procedure
rational?	rational number
real?	real number
string?	string

³ Per ragioni storiche, alcune ma non tutte le funzioni definite da LilyPond iniziano col prefisso ly:.

symbol?	symbol
vector?	vector

A.23.2 R5RS secondary predicates

Secondary predicates are only applicable to specific expressions (for example, to numbers). They will throw a type error when applied to expressions they are not intended for. The predicates here are part of the Scheme standard R5RS.

Type predicate	Description
char-alphabetic?	alphabetic character
char-lower-case?	lower-case character
char-numeric?	numeric character
char-upper-case?	upper-case character
char-whitespace?	whitespace character
even?	even number
exact?	exact number
inexact?	inexact number
negative?	negative number
odd?	odd number
positive?	positive number
zero?	zero

A.23.3 Guile predicates

These predicates are defined by Guile but are not part of a Scheme standard.

Type predicate	Description
hash-table?	hash table

A.23.4 LilyPond scheme predicates

These predicates are only available within LilyPond and defined in Scheme.

Type predicate	Description
alist?	association list (list of pairs)
boolean-or-symbol?	boolean or symbol
cheap-list?	list (use this instead of list? for faster processing)
color?	color
exact-rational?	an exact rational number
fraction?	fraction, as pair
grob-list?	list of grobs
grouped-number-list?	list of non-empty lists of numbers
index?	non-negative, exact integer
index-or-markup?	index or markup
key?	index or symbol
key-list?	list of indices or symbols
key-list-or-music?	key list or music
key-list-or-symbol?	key list or symbol
ly:skyline-pair?	pair of skylines
markup?	markup
markup-command-list?	markup command list
markup-list?	markup list
moment-pair?	pair of moment objects
musical-length?	non-negative exact rational, fraction (as pair), moment, or +inf.0

<code>musical-length-as-moment?</code>	non-negative moment with no grace part
<code>musical-length-as-number?</code>	non-negative exact rational or <code>+inf.0</code>
<code>non-negative-number?</code>	non-negative number
<code>number-list?</code>	number list
<code>number-or-grob?</code>	number or grob
<code>number-or-number-pair?</code>	number or pair of numbers
<code>number-or-pair?</code>	number or pair
<code>number-or-string?</code>	number or string
<code>number-pair?</code>	pair of numbers
<code>number-pair-list?</code>	list of number pairs
<code>optionally-grouped-beat-structure?</code>	a number list or a list of them
<code>positive-exact-rational?</code>	a positive, exact rational number
<code>positive-fraction?</code>	positive, finite fraction, as pair
<code>positive-musical-length?</code>	positive exact rational, fraction (as pair), moment, or <code>+inf.0</code>
<code>positive-musical-length-as-moment?</code>	positive moment with no grace part
<code>positive-musical-length-as-number?</code>	positive exact rational or <code>+inf.0</code>
<code>positive-number?</code>	positive number
<code>rational-or-procedure?</code>	an exact rational or procedure
<code>rhythmic-location?</code>	rhythmic location
<code>sane-simple-time-signature?</code>	simple time signature
<code>sane-time-signature?</code>	time signature
<code>scale?</code>	non-negative rational, fraction, or moment
<code>scheme?</code>	any type
<code>string-or-music?</code>	string or music
<code>string-or-pair?</code>	string or pair
<code>string-or-symbol?</code>	string or symbol
<code>symbol-key-alist?</code>	alist, with symbols as keys
<code>symbol-list?</code>	symbol list
<code>symbol-list-or-music?</code>	symbol list or music
<code>symbol-list-or-symbol?</code>	symbol list or symbol
<code>time-signature?</code>	time signature
<code>void?</code>	void

A.23.5 LilyPond exported predicates

These predicates are only available within LilyPond and usually defined in C++.

Type predicate	Description
<code>ly:book?</code>	book
<code>ly:context?</code>	context
<code>ly:context-def?</code>	context definition
<code>ly:context-mod?</code>	context modification
<code>ly:dimension?</code>	dimension, in staff space
<code>ly:dir?</code>	direction
<code>ly:dispatcher?</code>	dispatcher
<code>ly:duration?</code>	duration
<code>ly:event?</code>	post-event
<code>ly:font-metric?</code>	font metric
<code>ly:grob?</code>	graphical (layout) object
<code>ly:grob-array?</code>	array of grobs
<code>ly:grob-properties?</code>	grob properties
<code>ly:input-location?</code>	input location

ly:item?	item
ly:iterator?	iterator
ly:lily-lexer?	lily-lexer
ly:lily-parser?	lily-parser
ly:listener?	listener
ly:moment?	moment
ly:music?	music
ly:music-function?	music function
ly:music-list?	list of music objects
ly:music-output?	music output
ly:note-scale?	note scale
ly:otf-font?	OpenType font
ly:output-def?	output definition
ly:page-marker?	page marker
ly:pango-font?	Pango font
ly:paper-book?	paper book
ly:paper-system?	paper-system Prob
ly:pitch?	pitch
ly:prob?	property object
ly:score?	score
ly:skyline?	skyline
ly:source-file?	source file
ly:spanner?	spanner
ly:spring?	spring
ly:stencil?	stencil
ly:stream-event?	stream event
ly:transform?	coordinate transform
ly:translator?	translator
ly:translator-group?	translator group
ly:unpure-pure-container?	unpure/pure container

Appendice B Schema riassuntivo

Sintassi	Descrizione	Esempio
<code>1 2 8 16</code>	durate	
<code>c4. c4..</code>	punti di aumentazione	
<code>c d e f g a b</code>	scala	
<code>fis bes</code>	alterazione	
<code>\clef treble \clef bass</code>	chiavi	
<code>\time 3/4 \time 4/4</code>	indicazione di tempo	
<code>r4 r8</code>	pausa	
<code>d ~ d</code>	legatura di valore	
<code>\key es \major</code>	armatura di chiave	
<code>note'</code>	alzare l'ottava	
<code>note,</code>	abbassare l'ottava	

`c(d e)`

legatura di portamento



`c\ (c(d) e\)`

legatura di frase



`a8[b]`

travatura



`<< \new Staff ... >>`

più righe



`c-> c-.`

articolazioni



`c2\mf c\s fz`

dinamiche



`a\< a a\!`

crescendo



`a\> a a\!`

decrescendo



`< >`

accordo



`\partial 8`

anacrusi



`\tuplet 3/2 {f g a}`

terzine



`\grace`

abbellimenti



`\lyricmode { twinkle }`

inserimento del testo
vocale

twinkle

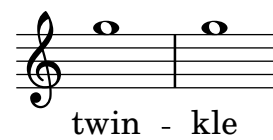
`\new Lyrics`

stampa del testo vocale

twinkle

`twin -- kle`

trattino nel testo vocale



`\chordmode { c:dim f:maj7 }`

accordi



`\new ChordNames`

mostrare i nomi degli
accordi

C[°] F^Δ

`<<{e f} \ {c d}>>`

polifonia



`s4 s8 s16`

pause spaziatrici

Appendix C GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements.”

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendice D Indice

Oltre a tutti i comandi e le parole chiave di LilyPond, questo indice elenca i termini musicali e le espressioni che si riferiscono a ognuno di essi, corredati di collegamenti alle relative sezioni del manuale.

!		/	
!	6	/	435
\!	132	/+	436
"		:	
" "	119	:	177
#		<	
#f (false)	822	\<	132
#t (true)	822	<...>	178
		<>	179, 344
		<	178
%		=	
%	494, 498	=	10
\%	821	\=	142, 820
%{ ... %}	494, 498		
,		>	
'	1	\>	132
		>	178
(?	
(.....	142	?	6
\(.....	145		
)		[
)	142	\[.....	456
\)	145	[.....	101
*]	
*	821	\]	456
]	101
,		^	
,	1	^	435, 643
—		—	
-	129, 452, 643	-	288, 643
-!	793		
-+	794	 	
--	793	118, 119
-	793		
->	793	~	
-^	793	~	58
-_	793		
•			
•	50		

1

15ma	24
15mb	24

8

8va	24
8vb	24

A

a capo	567
a capo, gestione con una voce apposita	569
a capo, glissando	151
a capo, testo	267
abbellimento	122, 129, 422
abbellimento al termine di una nota	122
abbellimento, e spaziatura rigorosa	125
abbellimento, e testo cantato	313
abbellimento, modifica delle impostazioni di formattazione	123, 124
abbellimento, modifica manuale	123, 124
abbellimento, sincronizzazione	126
\abs-fontsize	262, 713
\abs-hspace	725
\abs-vspace	726
\absolute	802
\accent	129, 793
‘accent’, articolazione	793
accento	130
“accento”, articolazione	129
\accentus	468, 795
‘accentus’, articolazione gregoriana	468
‘accentus’, articolazione Gregoriana	795
\accepts	620, 621, 623
acciacatura	122
\acciacatura	122, 803
acciacatura su più note	126
\accidental	755
\accidentalStyle	29, 803
AccidentalSuggestion	131
accollatura	203
accompagnamento, ritmo, mostrare	86, 87
accordatura non occidentale	485
accordatura per banjo	410
accordatura per liuto	411
accordatura personalizzata	379
accordatura predefinita, per strumenti con tasti ..	378
accordi ripetuti, soppressione	400
accordi, diagrammi automatici	403
accordi, ripetizione	362
accordi, ripetizioni con volte, sotto	443
accordo	178, 431
accordo alterato	434
accordo che attraversa i righi	350
accordo di settimana	432
accordo di settimana maggiore, formattazione	442
accordo esteso	434
accordo jazz	439
accordo trasversale ai righi	350
accordo vuoto	124, 134, 179, 344
accordo, alterazioni in	36
accordo, altezza relativa	179
accordo, diagramma per strumenti a tasto	382

accordo, diteggiatura	236
accordo, e legatura di valore	59
accordo, e ottava relativa	4
accordo, fondamentale	432
accordo, glissando, in intavolatura	376
accordo, inversione	436
accordo, modalità per	432
accordo, modificare una nota dell’	632
accordo, nome	432, 437
accordo, nome alternativo	439
accordo, nome, personalizzazione	439
accordo, nomi, eccezioni	442
accordo, power chord	409
accordo, qualità	432
accordo, ripetizione	180
accordo, soppressione delle ripetizioni	438
accordo, sopprimere ripetizioni	400
accordo, suddivisione sui righi con \autoChange ..	348
accordo, voicing	436
add-grace-property	124
add-stem-support	237
\addChordShape	396, 803
adding white background, to text	787
\addInstrumentDefinition	803
additionalPitchPrefix	440
\addlyrics	282, 284, 285
\addQuote	222, 803
\addToTagGroup	803
\aeolian	22
aerofono, strumento	419
\after	133, 803
\afterGrace	122, 803
afterGraceFraction	801
aggiungere diagramma dei tasti personalizzato ...	395
aggiunta, negli accordi	434
agogo	796
Aiken shape note head, thin variant	46
Aiken, testa di nota	44
\aikenHeads	44
\aikenHeadsMinor	45
\aikenThinHeads	44
\aikenThinHeadsMinor	45
aiuto, nuvoletta	247
al niente, forcilla	137
alfa, trasparenza	241
\alias	620
\align-on-other	726
alignAboveContext	216, 624
alignBelowContext	216, 302, 624
alist	799
allineamento del basso continuo	450
allineamento orizzontale del testo	264
allineamento orizzontale, testo vocale	298
allineamento su un oggetto	663
allineamento sulla cadenza	127
allineamento verticale del testo	265
allineamento verticale, dinamiche	138
allineamento verticale, segno testuale	138
allineamento, numeri di battuta	117
allineamento, testo, comandi	267
allineare il markup	264
allineare il testo	264
\allowPageTurn	573, 803
\allowVoltaHook	803
alterazione	6, 29

alterazione automatica.....	29	\appendToTagMarkup.....	804
alterazione di cortesia.....	6	\applyContext.....	610, 804
alterazione di sicurezza.....	6	\applyMusic.....	804
alterazione di un quarto di tono.....	8	\applyOutput.....	804
alterazione gregoriana.....	467	appoggiatura.....	122
alterazione mensurale.....	463	\appoggiatura.....	122, 804
alterazione moderna.....	32	\approximatePitch.....	804
alterazione nella notazione di Kiev.....	477	araba, esempio di musica.....	490
alterazione su più voci.....	32	araba, improvvisazione.....	489
alterazione tra parentesi.....	6	araba, modello di musica.....	490
alterazione, basso continuo, posizione.....	448	araba, musica.....	486
alterazione, cadenze.....	79	\arabicStringNumbers.....	359
alterazione, di cortesia.....	6	arabo, nomi delle note in.....	487
alterazione, di sicurezza.....	6	arcata in giù.....	356
alterazione, e legatura di valore.....	6	“arcata in giù”, indicazione di arcata.....	129
alterazione, e note simultanee.....	36	“arcata in su”, indicazione di arcata.....	129
alterazione, musica ficta.....	463	arcata in su.....	356
alterazione, musica in tempo libero.....	79	arcata, giù.....	794
alterazione, negli accordi.....	36	arcata, indicazione di.....	356
alterazione, stile <i>modern-cautionary</i>	31	arcata, su.....	794
alterazione, stile moderno.....	31	archi da orchestra.....	355
alterazione, tra parentesi.....	6	archi, scrivere per.....	355
alterazioni, glifi alternativi.....	36	armatura di chiave.....	6, 22
alterazioni, gruppi di glifi.....	708	armatura di chiave gregoriana.....	467
alterazioni, nascondere, su note legate all’inizio del sistema.....	7	armatura di chiave mensurale.....	463
\alterBroken.....	671, 803	armatura di chiave, impedire i segni di bequadro..	23
\alternative.....	161	armatura di chiave, non tradizionale.....	23
altezza.....	1	armatura di chiave, visibilità dopo un cambio esplicito.....	655
altezza isolata.....	50	armature di chiave arabe.....	487
altezza naturale.....	6	armonici artificiali.....	357
altezza predefinita.....	50	armonici naturali.....	356
altezza relativa, accordi.....	179	armonici su strumento con tasti.....	408
altezza, trasposizione di.....	11	armonici, flageolet.....	129, 794
altezze, trasposizione “intelligente”.....	12	armonici, in intavolatura.....	371
alto, chiave di.....	17, 709	armonici, intavolatura.....	364
Amazing Grace, esempio per cornamusa.....	423	armonico, testa di nota.....	41
ambito delle altezze.....	38	arpa.....	354
ambitus.....	38	arpa, diagramma del pedale.....	354
ambitus, distanza rispetto alla linea.....	39	arpa, pedale.....	354
ambitus, per voce.....	38	arpeggio.....	154
ambitus, posizionamento.....	40	\arpeggio.....	154
ambitus, su più voci.....	39	arpeggio attraverso il rigo, stile della parentesi...	158
ambitusAfter.....	40, 803	arpeggio e legature di valore.....	61
AmbitusLine.....	39	arpeggio spezzato.....	154
anacrusi.....	78	arpeggio, attraverso i righi.....	155
anacrusi, in una ripetizione.....	163	arpeggio, attraverso le voci.....	157
analisi musicologica.....	250	arpeggio, parentesi, trasversale ai righi.....	350
analisi, parentesi, con etichetta.....	251	arpeggio, simbolo speciale.....	155
analizzatore sintattico.....	801	arpeggio-direction.....	154
angolazione, con travature, modifica.....	90	\arpeggioArrowDown.....	154, 804
annidamento dei righi.....	207	\arpeggioArrowUp.....	154, 804
annidamento, ripetizione.....	169	\arpeggioBracket.....	155, 804
annotated moving by spacing, in text.....	777	\arpeggioNormal.....	154, 804
\annotated-moving.....	777	\arpeggioParenthesis.....	155, 804
annotated-spacing.....	601	\arpeggioParenthesisDashed.....	155, 804
annotazione.....	260	\arrow-head.....	269, 744
annotazione su pausa multipla.....	67	articolazione “espressivo”.....	134
antica, chiave.....	17	articolazione aperta.....	420, 794
‘aperta’, articolazione.....	420	articolazione chiusa.....	420
“aperto”, articolazione.....	129	articolazione gregoriana.....	468
aperto, high hat (charleston).....	129	articolazione Gregoriana, accentus.....	468, 795
apice.....	263	articolazione Gregoriana, circulus.....	468, 795
\append-to-tag.....	778	articolazione Gregoriana, episem finis.....	468
\appendToTag.....	530, 804	articolazione Gregoriana, episem initium.....	468

articolazione Gregoriana, ictus	468, 795
articolazione Gregoriana, semicirculus	468, 795
articolazione semiaperta	794
articolazione, accent	793
articolazione, accento	129
articolazione, aperto	129
articolazione, chiuso	129
articolazione, espressivo	129, 793
articolazione, marcato	129, 793
articolazione, portato	129, 793
articolazione, semiaperto	129
articolazione, staccatissimo	129, 793
articolazione, staccato	129, 793
articolazione, stopped	794
articolazione, tenuto	129, 793
articolazione, valori predefiniti, modifica	130
articolazioni	793
\articulate	549
articulate, script	549
articulate.ly	549
articulation-event	225
artificiali, armonici	357
\ascendens	469, 474
\assertBeamQuant	804
\assertBeamSlope	804
associatedVoice	284
associatedVoice	282, 314
assoluta, ottava	1
assoluto	1
\atLeft	804
\atRight	804
\auctum	469, 474
audio	537
aug	433
\augmentum	475, 805
auto-first-page-number	562
\auto-footnote	778
autoBeaming	91, 606
\autoBeamOff	88, 350
\autoBeamOn	88
\autoBreaksOff	567
\autoBreaksOn	567
\autoChange	347, 805
autoChange, e musica relativa	348
\autoLineBreaksOff	567
\autoLineBreaksOn	567
automaticBars	658
\autoPageBreaksOff	570
\autoPageBreaksOn	570

B

Bézier, curva, punti di controllo	666
backslashed digit	779
\backslashed-digit	779
Balloon_engraver	247
\balloonGrobText	247, 805
\balloonLengthOff	247
\balloonLengthOn	247
\balloonText	247, 805
banjo, accordatura	410
banjo, diagramma dei tasti	383
banjo, intavolatura	358, 410
banjo, tablatura	358

banjo-c-tuning	410
banjo-double-c-tuning	410
banjo-double-d-tuning	410
banjo-modal-tuning	410
banjo-open-d-tuning	410
banjo-open-dm-tuning	410
banjo-open-g-tuning	410
\bar	105, 112, 805
bar line, in markup	755
\bar-line	755
barCheckSynchronize	118
baritono, chiave di	17, 709
BarNumber	112
\barNumberCheck	119, 805
barNumberVisibility	112
barré, indicazione	382
barra, gambo con	124
barrata, testa di nota	41
Bartók, pizzicato	357
bartype	112
base-shortest-duration	591
\bassFigureExtendersOff	447
\bassFigureExtendersOn	447
\bassFigureStaffAlignmentDown	449
\bassFigureStaffAlignmentNeutral	449
\bassFigureStaffAlignmentUp	449
basso	796
basso acustico	796
basso cifrato	445
basso continuo	445
basso continuo, allineamento	450
basso continuo, alterazione, posizione	448
basso continuo, linee di estensione	448
basso numerato	445
basso, chiave di	17, 709
batteria	412, 414
battiti per minuto	75
battito, disposizione delle travature	97
battito, raggruppamento	98
battuta del ritornello	105
battuta in levare	78
battuta, controlli	118
battuta, numero	112
battuta, numero, a intervalli regolari	114
battuta, numero, nella prima misura	113
battuta, numero, stile del	114
battuta, stanghetta manuale	105
battuta, stanghetta	105
\beam	744
\beamExceptions	91, 805
beatBase	91, 96
beatStructure	91, 96
bemolle	6
bemolle, doppio	6
\bendAfter	148, 805
\bendHold	366, 805
bending, nella notazione per intavolatura	366
\bendStartLevel	366, 805
bequadro, impedirne l'inserimento nelle armature di chiave	23
bequadro, supplementare, impedire	7
binari ferroviari	148
binding-offset	559
bisbigliando	354
Bison	801

<code>\blackTriangleMarkup</code>	440
<code>blank-after-score-page-penalty</code>	561
<code>blank-last-page-penalty</code>	561
<code>blank-page-penalty</code>	561
blocco di commento	494, 498
blocco MIDI	539
BNF	801
<code>\bold</code>	261, 713
bongo.....	796
<code>\book</code>	494, 497
<code>\bookOutputName</code>	496, 805
<code>\bookOutputSuffix</code>	496, 805
<code>\bookpart</code>	495, 497, 571
<code>bookTitleMarkup</code>	507
<code>bottom-margin</code>	555
bounding box, of glyph.....	788
<code>\box</code>	268, 714
brace, in markup.....	781
bracket.....	352
<code>\bracket</code>	140, 744
<code>\bracket</code>	268
brano a più voci.....	319
<code>\break</code>	567
<code>break-align-symbols</code>	663
<code>break-visibility</code>	654
breakable.....	89
<code>\breakAlignInsert</code>	805
<code>breakbefore</code>	504
<code>\breathe</code>	147, 806
<code>BreathingSign</code>	147
<code>\breve</code>	49, 63
breve, nota, stile alternativo	50

C

cabasa	796
cadenza	79, 127
cadenza, allineamento su	127
cadenza, alterazioni.....	79
cadenza, interruzioni di linea	81
cadenza, interruzioni di pagina.....	81
cadenza, numeri di battuta	79
cadenza, stanghette.....	79
cadenza, travature.....	79
<code>\cadenzaOff</code>	79
<code>\cadenzaOn</code>	79
<code>\caesura</code>	467, 806
<code>callback</code>	799
cambiare i tipi di carattere	261
cambiare il numero di punti di aumentazione.....	51
cambiare il separatore dell'accordo.....	443
cambiare il simbolo del rigo.....	645
cambiare il simbolo di respiro.....	147
cambiare il tempo senza il segno di metronomo....	77
cambiare manualmente il rigo.....	344
cambio automatico di rigo	347
canali MIDI.....	545
cantante, nome del	312
cantico	330, 336
canto	330, 336
canzoniere	439
capotasto	387
<code>\caps</code>	714
carattere non-ASCII.....	531

carattere riservato, stampare	260
carattere speciale.....	531
carattere speciale, in modalità markup.....	260
carattere tipografico, famiglia di	799
carta, formato.....	553
<code>\cavum</code>	469, 474
<code>\center-align</code>	264, 726
<code>\center-column</code>	266, 727
centering column of text.....	727
centrare il testo sulla pagina.....	266
cesura.....	148
<code>\change</code>	344
changing direction of text column.....	728
<code>\char</code>	779
charleston semiaperto	416
charleston, aperto	129
charleston, pedale	129, 416
charleston, semiaperto.....	129
check-consistency	559
chiamata, segno di	119
chiave	6, 17
chiave antica	17
chiave con notina (citazione in corpo piccolo)	18
chiave della notazione di Kiev	476
chiave delle percussioni	412
chiave di alto.....	709
chiave di baritono.....	17, 709
chiave di basso.....	17, 709
chiave di contralto.....	17
chiave di Do.....	17
chiave di Fa	17
Chiave di Fa	709
chiave di Kiev.....	459
chiave di mezzosoprano	17, 709
chiave di Petrucci	709
chiave di Sol.....	17, 709
chiave di Sol tenore.....	709
chiave di soprano	17, 709
chiave di subbasso.....	17
chiave di tenore.....	17, 709
chiave di tenore per coro.....	18
chiave di varbaritono	17
chiave di violino	17, 709
chiave francese	17
chiave gregoriana.....	466
chiave mensurale	459, 709
chiave mensurale nera	459
chiave per musica antica.....	459
chiave Petrucci	459
chiave traspositrice	18
chiave trasposta, visibilità della.....	658
chiave, intavolatura.....	380
chiave, moderntab	380
chiave, modifica delle proprietà	20
chiave, stile di.....	459
chiave, stili di	709
chiave, tablatura	380
chiave, visibilità della trasposizione	658
chiave, visibilità dopo un cambio esplicito.....	655
chiavi dell'intavolatura	709
chiavi della musica antica.....	709
chiavi delle percussioni	709
chitarra, accordi, tabella.....	86
chitarra, intavolatura	358
chitarra, ritmo di accompagnamento, mostrare. 86, 87	

chitarra, tablatura.....	358	colore rgb.....	242
chitarra, testa di nota.....	41	colore x11.....	241, 243
‘chiusa’, articolazione.....	420	colore, codici simili a css.....	241
“chiuso”, articolazione.....	129	colori semitrasparenti.....	241
choral.....	33	colori, elenco dei.....	690
choral, stile delle alterazioni.....	33	coloring text.....	787
choral-cautionary.....	33	\column.....	266, 727
choral-cautionary, stile delle alterazioni.....	33	\column-lines.....	789
chordChanges.....	400, 438	combinare le parti, modifica del testo.....	198
\chordmode.....	5, 14, 393, 641	combinare musica contrassegnata con etichette...	530
chordNameExceptions.....	441, 442	combinatore delle parti.....	194
chordNameLowercaseMinor.....	440	combinatore delle parti, modifica del testo.....	198
ChordNames.....	393	combinazione automatica delle parti.....	194
ChordNames, con stanghette.....	443	\combine.....	269, 727
chordNameSeparator.....	441, 443	comma, intervallo.....	491
chordNoteNamer.....	440	commento.....	494, 498
chordPrefixSpacer.....	442	commento multirigo.....	494, 498
\chordRepeats.....	362, 806	common-shortest-duration.....	591
chordRootNamer.....	440	Completion_heads_engraver.....	84
\chords.....	438, 641	Completion_rest_engraver.....	84
Christian Harmony, testa di nota.....	44	\compound-meter.....	757
\circle.....	268, 745	\compressMMRests.....	66, 68, 806
circling text.....	745	comprimere la musica.....	57
\circulus.....	468, 795	\concat.....	728
‘circulus’, articolazione gregoriana.....	468	concatenating text.....	728
‘circulus’, articolazione Gregoriana.....	795	condensare le pause normali.....	70
citare le voci.....	222	conduzione, segni.....	98
citazione di un contesto.....	607	conga.....	796
citazione in corpo più piccolo, togliere.....	230	\consists.....	613, 620
citazione in corpo piccolo, chiave.....	18	contatore della ripetizione con segno percentuale..	174
citazione, nel testo vocale.....	281	contemporaneo, glissando.....	150
claves.....	796	contenitore impuro, Scheme.....	672
\clef.....	17, 806	contenitore puro, Scheme.....	672
clip-regions.....	534	contesto implicito.....	624
closure.....	799	contesto, ciclo di vita.....	611
cluster.....	183	contesto, conservazione.....	611
cluster di note.....	183	contesto, creazione e citazione.....	607
\cm.....	644	contesto, definire nuovo.....	620
coda.....	121	contesto, modifica proprietà di.....	615
\coda.....	129, 756, 795	contesto, ordine di disposizione.....	623
coda (uncinata).....	461	\context.....	607, 615
coda mensurale.....	461	\context nel blocco \layout.....	615
“coda”, segno.....	129, 795	context-spec-music.....	191
coda, segno, sulla stanghetta.....	256	\contextPropertyCheck.....	806
‘coda’, segno, variante.....	795	continuazione di una nota.....	452
coda, variante di, segno.....	129	conto della ripetizione con segno percentuale,	
\codaMark.....	806	visibilità.....	175
codette.....	102	controlli del numero di battuta.....	118
collisione.....	188	controlli di battuta.....	118
collisione di note.....	188	controlling general text alignment.....	730
collisione trasversale ai righi.....	345	controllo dell’ottava.....	10
collisione, ignorare.....	183	controllo della misura.....	118
collisione, numero di battuta.....	118	controllo delle altezze.....	10
collisione, voci trasversali ai righi.....	345	controllo di battuta, con ripetizioni.....	163
collisione,ignorare.....	193	controlpitch.....	10
colonna, testo.....	259, 266	copyright, segno.....	533
color.....	241	corda vuota, indicazione di.....	356
color, suitable for dichromatism.....	693	corda, bending, intavolatura.....	366
colorare un oggetto.....	241, 653	corda, numero di.....	359
colorare una nota.....	241	cornamusa.....	422
colorare una voce.....	187	cornamusa scozzese.....	422
colorata, nota.....	241	cornamusa, esempio.....	423
colorato, oggetto.....	241	coro, rigo per.....	203
colore.....	241	corona.....	121, 129, 794
colore in un accordo.....	243	corona breve.....	129, 794

corona Henze	129, 794
corona lunga	129, 794
corona, sulla stanghetta	256
correzione dell'ottava	10
countPercentRepeats	174
cowbell	796
\cr	132
creating a table	790
creating empty text object	782
creating horizontal space, in text	725, 732
creating text fraction	780
creating vertical space, in text	726, 741, 786
creazione del rigo	202
\cresc	134
crescendo	132
crescendo-event	225
crescendoSpanner	139
crescendoText	139
\crescHairpin	135
\crescTextCresc	135
cross	41
\crossStaff	350, 806
css, codici colore	241
\cueClef	226, 806
\cueClefUnset	806
\cueDuring	226, 806
\cueDuringWithClef	226, 806
CueVoice	226
cuica	796
currentBarNumber	112, 128
curva Bézier, punti di controllo	666
custodes	457
\customTabClef	758
custos	457

D

D.S. al Fine	121
\dashBang	130
\dashDash	130
\dashDot	130
\dashHat	130
\dashLarger	130
\dashPlus	130
\dashUnderscore	130
dead note, percussioni	419
\deadNote	42, 807
\deadNotesOff	42
\deadNotesOn	42
decorazione del testo	268
\decr	132
\decreasc	134
decrescendo	132
decrescendoSpanner	139
decrescendoText	139
default	29, 30
default-staff-staff-spacing	574
\default	119, 513
\defaultchild	624
\defaultTimeSignature	71
\defineBarLine	110, 807
definire tastiera predefinita	398
definire una stanghetta	110
definizione di output	605
definizioni di contesto per il MIDI	543

delimitatore di inizio del sistema	203
delimitatore di inizio del sistema, annidato	207
\deminutum	469, 474
\denies	620, 622, 623
\descendens	469, 474
destra, mano, posizionamento diteggiatura	406
diagramma degli accordi	382, 392
diagramma degli accordi per chitarra	86
diagramma degli accordi, automatico	403
diagramma dei tasti	382, 392
diagramma dei tasti per mancini	385
diagramma dei tasti personalizzato	389
diagramma dei tasti, aggiungere diteggiatura	404
diagramma dei tasti, aggiungere personalizzato ..	395
diagramma dei tasti, automatico	403
diagramma dei tasti, cambiare orientamento	389
diagramma dei tasti, insieme ai nomi degli accordi ..	393
diagramma dei tasti, mandolino	392
diagramma dei tasti, personalizzato	390
diagramma dei tasti, trasposizione	394
diagramma dei tasti, ukulele	392
diagramma del pedale dell'arpa	354
diagramma per legni, elenchi delle chiavi	430
diagramma per legni, elenco	428
diagrammi dei tasti personalizzati	382
diagrammi per legni, modifica	429
diamante, testa di nota	41
dichromatism, suitable colors	693
diesis	6
diesis, doppio	6
dim	433
\dim	134
dimension, of bounding box	788
dimensione	645
dimensione del testo	262
dimensione del tipo di carattere	262
dimensione del tipo di carattere (notazione)	231
dimensione standard del tipo di carattere (notazione)	235
\dimHairpin	135
diminuendo	132
\dimTextDecr	135
\dimTextDecresc	135
\dimTextDim	135
dinamica, indicazione, nuova	140
dinamiche	132
dinamiche assolute	132
dinamiche centrate nella musica per pianoforte ..	343
dinamiche del testo, stile	139
dinamiche editoriali	140
dinamiche nel MIDI	540
dinamiche personalizzate nel MIDI	540
dinamiche, allineamento verticale	138
dinamiche, centrarle nella musica per pianoforte ..	343
dinamiche, estensore del testo, personalizzazione ..	255
dinamiche, linea di estensione, nascondere	139
dinamiche, parentesi	140
dinamiche, posizionamento verticale	135
dinamiche, testo, stile	139
\dir-column	728
direzione verticale dei grob forzata	643
direzione verticale predefinita (-)	643
direzione, automatica, del gambo, rispetto alla linea centrale	245
direzione, giù (↓)	643

direzione, predefinita (-).....	643
direzione, predefinita, del gambo, rispetto alla linea centrale	244
direzione, sù (^).....	643
diritto d'autore.....	533
\discant.....	773
discanto, simbolo, fisarmonica	352
disegnare il simbolo del rigo	645
disegnare legature di valore e di portamento	667
disegnare manualmente le legature di valore.....	61
disegnare un oggetto grafico	268
\displayLilyMusic.....	550, 807
\displayMusic.....	807
\displayScheme.....	807
disposizione automatica delle travature, proprietà delle indicazioni di tempo.....	72
disposizione delle travature che segue strettamente il battito.....	97
disposizione delle travature, proprietà predefinite delle indicazioni di tempo.....	72
distanza assoluta	644
distanza proporzionale.....	644
distanza, tra i righi.....	574
dita, cambio.....	235
diteggiatura	235
diteggiatura del pollice	236
diteggiatura per accordo	236
diteggiatura, aggiungere al diagramma dei tasti..	404
diteggiatura, dentro al rigo	237
diteggiatura, e pausa multipla	70
diteggiatura, grafico	421
diteggiatura, mano destra, per strumento a tasti .	405
diteggiatura, mano destra, posizionamento.....	406
diteggiatura, orientamento.....	236
diteggiatura, pollice	129, 794
diteggiatura, scivolare	237
diteggiatura, simboli per legni	420
diteggiatura, supporto per i gambi.....	237
dito che pizzica, orientamento	236
“dito del piede”, segno del pedale	129
divisio	467
\divisioMaior.....	467
\divisioMaxima.....	467
\divisioMinima.....	467
divisione delle note	84
divisione delle pause.....	84
divisione, voce.....	321
divisiones	467
Do, chiave di	17
dodecaphonic	34
dodecaphonic, stile delle alterazioni	34
dodecaphonic-first	35
dodecaphonic-first, stile delle alterazioni	35
dodecaphonic-no-repeat.....	34
dodecaphonic-no-repeat, stile delle alterazioni	34
doppie, legature di portamento, per accordi legati.	144
doppio bemolle	6
doppio diesis	6
doppio punto, nota	50
\dorian.....	22
dorico	22
\dotsDown.....	50
\dotsNeutral	50
\dotsUp.....	50
\doubleflat	759

\doublesharp.....	759
doubleSlurs	144
‘down pralltriller’, ornamento	793
‘down-bow’, indicazione di arcata	794
\downbow.....	129, 356, 794
\downmordent	129, 793
\downprall.....	129, 793
\draw-circle.....	269, 745
\draw-dashed-line	746
\draw-dotted-line	746
\draw-hline	746
\draw-line.....	269, 747
\draw-squiggle-line	747
drawing beam, within text.....	744
drawing box, with rounded corners	748
drawing box, with rounded corners, around text .	753
drawing circle, within text	745
drawing dashed line, within text	746
drawing dotted line, within text	746
drawing ellipse, around text	748
drawing line, across a page	746
drawing line, within text	747
drawing oval, around text	749
drawing path	750
drawing polygon	751
drawing solid box, within text	748
drawing squiggled line, within text.....	747
drawing triangle, within text	754
\dropNote.....	436, 807
\drummode.....	202, 412, 641
drumPitchNames.....	417
drumPitchTable.....	417
\drums	412, 641
DrumStaff	202
drumStyleTable.....	416
durata isolata	49
durata predefinita	49
durata, di una nota.....	49
durata, linea	452
durata, scalare	57
\dwn.....	487
\dynamic.....	140, 715
dynamic-event	225
\dynamicDown.....	135
DynamicLineSpanner	135, 138
\dynamicNeutral.....	135
\dynamicUp	135

E

\easyHeadsOff	42
\easyHeadsOn	42
eccezione, nomi dell’accordo	442
editoriali, dinamiche	140
edizione medicea	454
edizione vaticana	454
effetti nel MIDI	548
elenca i tipi di carattere disponibili	277
elenco dei colori	690
elenco dei diagrammi per legni.....	428
elenco delle chiavi nei diagrammi per legni	430
\ellipse.....	748
Emmentaler, font.....	693
\EnableGregorianDivisiones.....	821

<code>\enablePerStaffTiming</code>	807
encapsulated postscript, output.....	535
enclosing text, in box.....	714
enclosing text, in box with rounded corners.....	753
<code>\endcr</code>	132
<code>\enddecr</code>	132
<code>\endSpanners</code>	651, 807
eoio.....	22
‘episem finis’, articolazione gregoriana.....	468
‘episem initium’, articolazione gregoriana.....	468
<code>\episemFinis</code>	468
<code>\episemInitium</code>	468
EPS image.....	749
EPS, output.....	535
<code>\epsfile</code>	269, 748
equalizzazione MIDI.....	540
equalizzazione, strumento, cambiare predefinita nel MIDI.....	542
esempio di musica araba.....	490
esempio per cornamusa.....	423
espandere la musica.....	57
espressione nel MIDI.....	548
espressione, markup.....	260
espressivo.....	134
<code>\espressivo</code>	129, 134, 793
“espressivo”, articolazione.....	129, 793
estensione.....	38
estensore.....	292
estensore del testo.....	254
estensore del testo, dinamiche, personalizzazione.....	255
estensore del testo, formattazione.....	255
estensore spezzato, modifica.....	671
estensore, modifica.....	671
<code>\etc</code>	676
etichetta.....	527
etichette, gruppi di.....	530
<code>\eventChords</code>	807
evento segnaposto.....	179
<code>explicitClefVisibility</code>	655
<code>explicitKeySignatureVisibility</code>	655
extent, of actual inking.....	788
extent, of bounding box.....	788
extra-offset.....	574
<code>\eyeglasses</code>	779
<code>Ez_numbers_engraver</code>	43

F

<code>\f</code>	132
Fa, chiave di.....	17, 709
false (#f).....	822
famiglia di font.....	263
famiglia di tipi di carattere.....	263
famiglia di tipi di carattere, impostare.....	277
<code>\featherDurations</code>	104, 807
feature, OpenType font.....	719
<code>\fermata</code>	129, 759, 794
fermata su pausa multipla.....	67
Ferneyhough, forcella.....	138
Feta, font.....	693
<code>\ff</code>	132
<code>\fff</code>	132
<code>\ffff</code>	132
<code>\fffff</code>	132

fiati.....	419
<code>\figured-bass</code>	715
<code>figuredBassAlterationDirection</code>	448
<code>figuredBassPlusDirection</code>	448
<code>\figuremode</code>	445, 642
<code>\figures</code>	445, 642
<code>\fill-line</code>	266, 728
<code>\fill-with-pattern</code>	729
<code>\filled-box</code>	269, 748
finale alternativo.....	161
finale alternativo, con legature di valore.....	163
finale alternato, ripetizioni.....	171
finali alternativi, e testo vocale.....	303
finalis.....	467
<code>\finalis</code>	467
fine ripetizione.....	169
<code>\finger</code>	235, 715, 807
<code>fingeringOrientations</code>	236
first-page-number.....	562
<code>\first-visible</code>	779
fisarmonica.....	352
fisarmonica, simbolo di discanto.....	352
fisarmonica, simbolo di registro.....	352
<code>\fixed</code>	2, 807
<code>\flageolet</code>	129, 794
“flageolet”, armonici.....	129, 794
<code>\flat</code>	759
Flex.....	800
<code>\flexa</code>	475
floor tom tom.....	796
followVoice.....	348
fondamentale dell’accordo.....	432
font.....	274, 799
font (notazione), dimensione standard.....	235
font Emmentaler.....	693
font feature, OpenType.....	719
font Feta.....	693
font non testuali, nel markup.....	274
font Parmesan.....	693
font, cambiare.....	261
font, dimensione.....	262
font, famiglia.....	263
font, impostare.....	277
font, impostare la dimensione.....	565
font, modificarli per l’intero documento.....	277
font, ridimensionamento.....	231
font-interface.....	235, 274
font-size.....	231, 235
<code>\fontCaps</code>	715
<code>fontSize</code>	231
<code>\fontSize</code>	262, 716
<code>\footnote</code>	512, 779, 808
<code>Forbid_line_break_engraver</code>	55
forcella.....	132
forcella allargata (flared-hairpins).....	138
forcella angolata.....	659
forcella continua (costante-hairpins).....	138
forcella Ferneyhough.....	138
forcella, al niente.....	137
forcella, allineare le estremità alle direzioni di NoteColumn.....	136
forcella, lunghezza minima.....	136
forcella, spostare le estremità della.....	137
forcella, terminata sulle stanghette.....	136
forget.....	35

forget, stile delle alterazioni	35
forma di un accordo, per strumenti con tasti	396
formato carta	553
formato carta, orientamento	554
formato carta, orizzontale (landscape)	554
formato del segno di chiamata	120
formato pagina	553
formattare un estensore del testo	255
formattazione del gruppo irregolare	52
formattazione dell'accordo di settimana maggiore	442
formattazione della pagina	595
formattazione della terzina	52
formattazione mensurale	206
formattazione, nel testo vocale	281
formattazione, notine	226
formattazione, oggetto	800
four-string-banjo	410
\fp	132
\fraction	780
frammento	226
frammento musicale	534
frammento, citare	222
francese, chiave	17
frase, legatura di	145
fraseggio, nel testo vocale	289
\freeBass	773
\frenchChords	439
\fret-diagram	382, 768
fret-diagram, diagramma dei tasti	382
fret-diagram-interface	389
\fret-diagram-terse	385, 768
fret-diagram-terse, diagramma dei tasti conciso ..	385
\fret-diagram-verbose	387, 769
fret-diagram-verbose, diagramma dei tasti ridondante	387
FretBoards	392
frigio	22
\fromproperty	780
Funk, testa di nota	44
\funkHeads	44
\funkHeadsMinor	45

G

G, chiave di	709
G2, chiave di	709
gambo	244
gambo barrato	124
gambo che attraversa i rigli	350
gambo invisibile	244
gambo orizzontale	370
gambo trasversale ai rigli	350
gambo, direzione	244
gambo, direzione automatica rispetto alla linea centrale	245
gambo, direzione predefinita rispetto alla linea centrale	244
gambo, giù	244
gambo, in intavolatura	370
gambo, neutrale	244
gambo, su	244
\general-align	265, 730
\germanChords	439
gestione del tempo	128
GG, chiave di	709

ghost note	243
ghost note, percussioni	419
giù, direzione (↓)	643
giustificato, testo	267
glide	237
glifi di alterazione alternativi	36
glifi, gruppi, alterazioni	708
glifo	799
glifo musicale	121
glissando	149
\glissando	149
glissando che vanno a capo	151
glissando contemporaneo	150
glissando, accordi in intavolatura	376
glissando, attraverso le ripetizioni	152
glissando, e ripetizioni	169
glissando, segni di tempo	151
\glissandoMap	150
glissato, in intavolatura	375
glyph	799
glyph, bounding box	788
\grace	122, 808
grado dell'accordo, alterazione	435
graffa verticale	203
graffa, varie dimensioni	274
grafica inclusa	269
grafica, inclusione	268
grafico della diteggiatura	421
grammatica di LilyPond	801
gregoriana, alterazione	467
gregoriana, armatura di chiave	467
gregoriana, articolazione	468
Gregoriana, articolazione, accentus	795
gregoriana, articolazione, accentus	468
Gregoriana, articolazione, circulus	795
gregoriana, articolazione, circulus	468
gregoriana, articolazione, episem finis	468
gregoriana, articolazione, episem initium	468
Gregoriana, articolazione, ictus	795
gregoriana, articolazione, ictus	468
gregoriana, articolazione, semicirculus	468
Gregoriana, articolazione, semicirculus	795
gregoriana, musica, trascrizione moderna	336
gregoriano, legatura di neumi quadrati	469
GregorianTranscriptionStaff	202
Grid_line_span_engraver	248
Grid_point_engraver	248
gridInterval	248
griglia	248
grob	625, 800
grob e whiteout	653
grob, direzione verticale forzata predefinita	643
grob, posizione verticale	643
grob, proprietà	630
grob, sovrascrivere	653
grob, visibilità del	652
grob-interface	800
\grobdescriptions	808
grow-direction	104
“gruppetto barrato”, ornamento	129
“gruppetto Haydn”, ornamento	129
“gruppetto rovesciato”, ornamento	129
“gruppetto”, ornamento	129
gruppo di rigli	203
gruppo di rigli, con parentesi quadra all'inizio ...	205

gruppo irregolare	51
gruppo irregolare con travature, interruzione di linea.....	55
gruppo irregolare, formattazione del	52
gruppo irregolare, inserirne multipli.....	52
gruppo irregolare, modifiche del numero del	53
gruppo irregolare, posizionamento della parentesi quadra	52
gruppo irregolare, raggruppamento.....	52
gruppo ritardato, ornamento.....	131
Guida al funzionamento interno	605
guide.....	457
giuro.....	796

H

‘half open’, articolazione.....	794
half-open high hat.....	796
\halfopen.....	129, 416, 794
\halign.....	264, 731
hammer on.....	377
handclap.....	796
Harmonia Sacra, testa di nota	44
\harmonic.....	42, 356, 364
\harmonicByFret.....	364, 808
\harmonicByRatio.....	364, 808
\harmonicNote.....	808
\harmonicsOff.....	356
\harmonicsOn.....	356, 808
\harp-pedal.....	770
‘Haydn turn’, ornamento	793
\haydnturn.....	129, 793
\hbracket.....	268, 749
\hcenter-in.....	732
\header.....	497
\heel.....	129, 794
‘heel’, segno di pedale	794
\heelcircle.....	129, 794
Henze, corona.....	794
\henzelongfermata.....	129, 794
\henzeshortfermata	129, 794
\hide.....	652, 808
\hideKeySignature	422
\hideNotes.....	240
\hideSplitTiedTabNotes.....	363
\hideStaffSwitch.....	348
high bongo.....	796
high conga.....	796
high hat.....	796
high hat semiaperto	416
high hat, aperto.....	129
high hat, half-open	796
high hat, open	796
high hat, pedale	129, 416, 796
high hat, semiaperto	129
high timbale	796
horizontal-shift	560
Horizontal_bracket_engraver.....	250
HorizontalBracketText.....	251
horizontally centering text.....	726
\hspace.....	732
hufnagel.....	454, 455
\huge.....	231, 264, 716
hyperlink, as QR code.....	784

I

\ictus.....	468, 795
‘ictus’, articolazione Gregoriana	795
‘ictus’, articolazione gregoriana	468
\if.....	767
\iij.....	470
\IIJ.....	470
\ij.....	470
\IJ.....	470
image.....	749
\image.....	749
immagine, inclusione	269
immutabile, oggetto	800
immutable	800
importing stencil, into text	786
\improvisationOff.....	47, 86
\improvisationOn.....	47, 86
improvvisazione	47
improvvisazione araba.....	489
\in.....	644
\incipit.....	478, 808
incipit, aggiungere.....	478
incisore, includere nel contesto.....	620
\inclinatum	469, 474
inclinazione, estensore dell’ottava, modifica	26
\include.....	498, 524
includere le impostazioni	531
inclusione di file	524
incorniciatura del testo.....	268
indent	220, 560, 595
indicare la posizione e il barré per uno strumento con tasti.....	407
indicazione di arcata	356
indicazione di arcata, in giù	129
indicazione di arcata, in sù	129
indicazione di armonici, nell’intavolatura.....	364
indicazione di barré	382
indicazione di corda vuota	356
indicazione di diteggiatura per accordo	236
indicazione di No Chord in ChordNames.....	437
indicazione di tempo	71
indicazione di tempo composto	83
indicazione di tempo doppia	81
indicazione di tempo polimetrico.....	81
indicazione di tempo, impostazioni predefinite.....	72
indicazione di tempo, mensurale	460
indicazione di tempo, mostrare solo il numeratore ..	74
indicazione di tempo, multipla	614
indicazione di tempo, proprietà, ripristinare i valori predefiniti	73
indicazione di tempo, stile	71, 460
indicazione di tempo, visibilità.....	71
indicazione dinamica nuova	140
indicazione dinamica, più di un segno su una nota ..	133
indicazione manuale di ripetizione	169
indicazione metronomica.....	75
indicazione polimetrica	81
indicazione testuale.....	256
indicazioni di tempo arabe.....	489
indicazioni dinamiche multiple, su una nota	133
informazioni sul tempo e ripetizioni.....	169
ingrandimento del tipo di carattere	231
\inherit-acceptability.....	622, 809
\initialContextFrom.....	809

inizializzazione del rigo	202
inizio del sistema	203
inizio ripetizione	169
inner-margin	559
inno	319, 330, 336
inno, e misure parziali	340
input-language	810
insegnamento (<i>teaching</i>), stile delle alterazioni	35
inserimento del testo	281
inserimento di gruppi irregolari multipli	52
inserting music, into text	764
inserting PostScript directly, into text	752
inserting URL link, into text	755
\inStaffSegno	164, 809
\instrumentSwitch	809
intavolatura	202, 358
intavolatura e armonici	364
intavolatura per banjo	358, 410
intavolatura per chitarra	358
intavolatura per liuto	411
intavolatura personalizzata	378
intavolatura, accordature personalizzate	379
intavolatura, accordature predefinite	378
intavolatura, banjo	378
intavolatura, base	360
intavolatura, basso	378
intavolatura, basso elettrico	378
intavolatura, chiave	380
intavolatura, chiavi di	709
intavolatura, chitarra	378
intavolatura, contrabbasso	378
intavolatura, e armonici	371
intavolatura, e bending	366
intavolatura, e gambi	370
intavolatura, e polifonia	370
intavolatura, e slide	375
intavolatura, e travature	370
intavolatura, glissando di accordi	376
intavolatura, mandolino	378
intavolatura, martellato	377
intavolatura, microtono	381
intavolatura, predefinita	360
intavolatura, quarto di tono	381
intavolatura, strappato	377
intavolatura, ukulele	378
intavolatura, viola	378
intavolatura, violino	378
intavolatura, violoncello	378
interfaccia	800
interfaccia dell'oggetto grafico	800
interface, layout	625
Internals Reference	605
interruzione di linea	105
interruzione di linea manuale	567
interruzione di linea regolare	569
interruzione di linea, cadenze	81
interruzione di linea, musica in tempo libero	81
interruzione di linea, travature	89
interruzione di pagina	595
interruzione di pagina, cadenze	81
interruzione di pagina, musica in tempo libero	81
interruzione, di pagina o linea, gestione con una voce apposita	569
interruzione, nella musica in tempo libero	81
intervallo intermedio	486

intervallo, comma	491
intestazione	499
\inversion	14, 809
inversione	14
inversione di accordo	436
inversione modale	16
\invertChords	436, 809
invisibile, gambo	244
invisibile, nota	240
\ionian	22
ionio	22
isolata, altezza	50
isolata, durata	49
isolati, segni percentuali, nelle ripetizioni	175
\italianChords	439
\italic	261, 716

J

jazz, accordo	439
\jump	809
\justified-lines	273, 789
\justify	267, 733
\justify-field	733
\justify-line	734
\justify-string	735
justifying lines of text	789
justifying text	733

K

\keep-with-tag	781
\keepWithTag	527, 809
\key	22, 45, 809
Kiev, chiave	459
Kiev, notazione di, alterazione	477
Kiev, notazione di, legatura	477
Kiev, notazione, chiave	476
\kievanOff	476
\kievanOn	476
\kievanStaff	475
KievanVoice	475
\killCues	230, 809
kirchenpausen	68

L

\label	521, 809
laissez vibrer	59
\laissezVibrer	59
\language	810
\languageRestore	810
\languageSaveAndChange	810
\large	231, 264, 716
\larger	262, 264, 716
last-bottom-spacing	557
\layout	497, 563, 605, 615
layout interface	625
layout, file	565
layout-set-staff-size	565
lead sheet	439
\left-align	264, 735
left-aligning text	735
\left-brace	781

<code>\left-column</code>	736	<code>\ligature</code>	810
<code>left-margin</code>	558	ligature, in text.....	728
legato discendente.....	377	<code>\line</code>	736
legatura.....	456	line-width.....	558, 595
legatura di frase.....	142, 145	<code>\linea</code>	469, 474
legatura di frase puntata.....	145	linea del cambio rigo.....	348
legatura di frase tratteggiata.....	145	linea del rigo, fermare e avviare.....	210
legatura di frase, definizione dei modelli di tratteggio.....	146	linea del rigo, modificare.....	210
legatura di frase, metà continua e metà tratteggiata.....	146	linea di commento.....	494, 498
legatura di Kiev.....	477	linea di commento, singola.....	494, 498
legatura di neumi quadrati.....	469	linea di durata.....	452
legatura di portamento.....	142	linea di estensione, dinamiche, nascondere.....	139
legatura di portamento continua.....	143	linea trasversale ai rigi.....	348
legatura di portamento multipla.....	142	linea verticale tra i rigi.....	248
legatura di portamento punteggiata.....	143	linea, cambio di rigo.....	348
legatura di portamento simultanea.....	142	linea, interruzione, gestione con una voce apposita.....	569
legatura di portamento tratteggiata.....	143	linea, per indicare la durata.....	452
legatura di portamento, definizione dei modelli di tratteggio.....	144	linee della griglia, modificare aspetto.....	249
legatura di portamento, definizione dei modelli di tratteggio per il fraseggio.....	146	linee di estensione del basso continuo.....	448
legatura di portamento, definizione del modello di tratteggio.....	143	<code>\lineprall</code>	129, 793
legatura di portamento, e ripetizioni.....	169	lingua, nomi delle altezze in un'altra.....	8
legatura di portamento, frase puntata.....	145	lingua, nomi delle note in un'altra.....	8
legatura di portamento, frase tratteggiata.....	145	lista di associazioni.....	799
legatura di portamento, fraseggio multiplo.....	145	liuto, accordatura.....	411
legatura di portamento, fraseggio simultaneo.....	145	liuto, intavolatura.....	411
legatura di portamento, fraseggio, definizione dei modelli di tratteggio.....	146	livello del chorus nel MIDI.....	548
legatura di portamento, metà tratteggiata e metà continua.....	143	<code>\locrian</code>	22
legatura di portamento, modificare.....	666	locrio.....	22
legatura di portamento, posizionamento manuale.....	142	<code>\longa</code>	49, 63
legatura di portamento, sopra le note.....	142	<code>\longfermata</code>	129, 794
legatura di portamento, sotto le note.....	142	<code>\lookup</code>	781
legatura di portamento, testo all'interno.....	144	low bongo.....	796
legatura di portamento, tratto metà continuo e metà tratteggiato.....	146	low conga.....	796
legatura di valore.....	58	low timbale.....	796
legatura di valore punteggiata.....	60	<code>\lower</code>	265, 736
legatura di valore tratteggiata.....	60	'lower mordent', ornamento.....	793
legatura di valore, aspetto.....	60	lowering text.....	736
legatura di valore, con arpeggi.....	61	lunghezza di una nota.....	49
legatura di valore, disegno manuale.....	61	lunghezza minima, forcilla.....	136
legatura di valore, e alterazione.....	6	ly:minimal-breaking.....	572
legatura di valore, e parentesi della volta.....	59	ly:one-line-auto-height-breaking.....	572
legatura di valore, finali alternativi.....	163	ly:one-line-breaking.....	572
legatura di valore, in un accordo.....	59	ly:one-page-breaking.....	572
legatura di valore, laissez vibrer.....	59	ly:optimal-breaking.....	571
legatura di valore, modificare.....	666	ly:page-turn-breaking.....	572
legatura di valore, nel testo vocale.....	288	<code>\lydian</code>	22
legatura di valore, nelle ripetizioni.....	163	<code>\lyricmode</code>	281, 282, 642
legatura di valore, posizionamento.....	60	<code>\lyrics</code>	642
legatura di valore, ripetizione.....	59	<code>\lyricsto</code>	282, 284, 285
legature di frase multiple.....	145		
legature di frase simultanee.....	145		
legature di portamento doppie per accordi legati.....	144		
legature di portamento, stile.....	143		
legature mensurali bianche.....	464		
legni, simboli di diteggiatura.....	420		
lexer.....	800		
lidio.....	22		
		M	
		m.....	433
		maggiore.....	22
		magnification->font-size.....	231, 565
		<code>\magnify</code>	262, 717
		magnifying text.....	717
		<code>\magnifyMusic</code>	231, 810
		<code>\magnifyStaff</code>	566, 810
		magstep.....	231, 565, 644
		maj.....	433
		<code>\major</code>	22
		majorSevenSymbol.....	440, 442
		makam.....	491

makamlar	491	MensuralStaff	202, 458
make-dynamic-script	140	MensuralVoice	458
make-pango-font-tree	277	Mensurstriche	479
\makeClusters	183, 810	\mergeDifferentlyDottedOff	188
\makeDefaultStringTuning	810	\mergeDifferentlyDottedOn	188
mancino, diagramma dei tasti	385	\mergeDifferentlyHeadedOff	188
maqam	486	\mergeDifferentlyHeadedOn	188
maracas	796	merging text	727, 737
marcato	130	metadati MIDI	511
\marcato	129, 793	metadati PDF	511
“marcato”, articolazione	129, 793	metronomo, segno, sotto il rigo	77
marginale di rilegatura	559	metronomo, segno, testo personalizzato	78
marginale, testo che va oltre	254	mezzosoprano, chiave di	17, 709
\mark	119, 256, 810	\mf	132
Mark_engraver	257	microtono	9
\markalphabet	781	microtono, intavolatura	381
\markLengthOff	76, 257	\midi	497, 605
\markLengthOn	76, 257	MIDI	27, 537
\markletter	782	MIDI, bilanciamento stereo	548
markup	260	MIDI, blocco	539
\markup	256, 258, 259, 260, 642	MIDI, canali	545
markup condizionale	510	MIDI, definizioni di contesto	543
markup multilinea	266	MIDI, dinamiche	540
markup, allineare	264	MIDI, dinamiche personalizzate	540
markup, centrare sulla pagina	266	MIDI, effetti	548
markup, decorazione	268	MIDI, equalizzazione	540
markup, due colonne	259	MIDI, espressione	548
markup, espressione	260	MIDI, livello del chorus	548
markup, image	749	MIDI, metadati	511
markup, incorniciatura	268	MIDI, modificare l'equalizzazione predefinita di uno strumento	542
markup, multipagina	273	MIDI, notazione non supportata	538
markup, nel testo	282	MIDI, notazione supportata	537
markup, notazione musicale dentro	270	MIDI, panning, posizione	548
markup, partitura dentro	272	MIDI, riverbero	548
markup, rhythm	763	MIDI, strumento	549
markup, sintassi	260	MIDI, tracce	545
markup, testo a capo	267	MIDI, trasposizione	27
markup, testo giustificato	267	MIDI, un canale per voce	546
markup, testo, comando di allineamento	267	MIDI, uso delle ripetizioni	545
markup, testo, dentro le legature di portamento ..	144	MIDI, volume	540
markup, testo, padding	268	midiBalance	548
markup-markup-spacing	557	midiChannelMapping	545
markup-system-spacing	557	midiChorusLevel	548
\markuplist	259, 273	midiDrumPitches	417
\markupMap	810	midiExpression	548
martellato	377	midiPanPosition	548
max-systems-per-page	560	midiReverbLevel	548
\maxima	49, 63	min-systems-per-page	560
Measure_grouping_engraver	98	minimum-Y-extent	574
measureBarType	112	minimumFret	361, 404
measureLength	91, 128	\minor	22
measurePosition	78, 128	minorChordModifier	441
Medicaea, Editio	455	minore	22
melisma	289, 292	mirroring markup	754
\melisma	289	misolidio	22
melisma, con travature	89	misura	71
\melismaEnd	289	misura parziale	78
melodia alternativa, passare a	314	misura parziale, negli inni	340
melodia, mostrare i ritmi	86	misura, numero	112
mensurale	454, 455	misura, numero, stile del	114
mensurale, chiave	459, 709	misura, raggruppamento	98
mensurale, coda	461	misura, ripetizione	173
mensurale, formattazione	206	misura, sottoraggruppamento	98
mensurale, notazione, signum congruentiae	795	misura, stanghetta manuale	105
mensurali, legature bianche	464		

misura, stanghette.....	105
mixed.....	352
\mixolydian.....	22
\mm.....	644
modale, inversione.....	16
modale, trasposizione.....	15
\modalInversion.....	16, 810
modalità accordo.....	432
modalità markup, caratteri speciali.....	260
modalità markup, testo tra virgolette.....	260
\modalTranspose.....	15, 811
mode.....	801
modello di musica araba.....	490
modern.....	31
<i>modern</i> , stile delle alterazioni.....	31, 32
modern-cautionary.....	31
<i>modern-cautionary</i> , stile delle alterazioni.....	31
modern-voice.....	32
modern-voice-cautionary.....	32
<i>modern-voice-cautionary</i> , stile delle alterazioni.....	32
moderntab, chiave.....	380
modifica automatica del rigo.....	347
modifica di una proprietà.....	628
modifica manuale del rigo.....	344
modifica, ripristinare.....	631
modificare il nome di uno strumento.....	221
modificare in un solo istante.....	631
modificare un abbellimento.....	123, 124
modificatore, nell'accordio.....	432
modo.....	22
modo ecclesiastico.....	22
\mordent.....	129, 793
'mordent, lower', ornamento.....	793
'mordent', ornamento.....	793
'mordent, upper', ornamento.....	793
"mordente inferiore", ornamento.....	129
"mordente inverso inferiore", ornamento.....	129
"mordente inverso lungo", ornamento.....	129
"mordente inverso superiore", ornamento.....	129
"mordente inverso", ornamento.....	129
"mordente superiore", ornamento.....	129
"mordente", ornamento.....	129
movimenti, molteplici.....	494
\mp.....	132
multi-measure rest, within text, by duration.....	762
multi-measure rest, within text, by number of measures.....	759
\multi-measure-rest-by-number.....	759
multilinea, markup.....	266
multilinea, testo.....	266
MultiMeasureRestScript.....	67
MultiMeasureRestText.....	67
multipagina, testo.....	273
multipla, legatura di portamento.....	142
musica a quattro battute.....	569
musica antica, chiave per.....	459
musica antica, chiavi di.....	709
musica araba.....	486
musica di insieme.....	323
musica ficta.....	463
musica gregoriana, trascrizione moderna.....	336
musica in tempo libero.....	79, 128
musica in tempo libero, alterazioni.....	79
musica in tempo libero, interruzioni di linea.....	81
musica in tempo libero, interruzioni di pagina.....	81

musica in tempo libero, numeri di battuta.....	79
musica in tempo libero, stanghette.....	79
musica in tempo libero, travature.....	79
musica mensurale, segno.....	460
musica mensurale, trascrizione di.....	206
musica orchestrale.....	323
musica ottomana.....	491
musica parallela.....	199
musica per pianoforte, centrare le dinamiche.....	343
musica per principianti.....	42
musica polifonica.....	188
musica relativa, e autoChange.....	348
musica religiosa.....	330
musica rinascimentale.....	206
musica turca.....	491
musica, dentro il blocco markup.....	270
musical, teatro.....	322
musicale, suggerimento.....	325
\musicglyph.....	121, 760
\musicLength.....	811
\musicMap.....	811
musicologia, analisi.....	250
musicQuotes.....	801
mutable.....	801
mute bongo.....	796
mute conga.....	796
mute timbale.....	796

N

\n.....	132
N.C. (No Chord), simbolo.....	437
\name.....	620
nascondere alterazioni, su note legate all'inizio del sistema.....	7
nascondere i righi.....	217
nascondere i righi antichi.....	217
nascondere i righi ritmici.....	217
nascondere i tetragrammi.....	217
nascondere la linea di estensione delle dinamiche.....	139
nascosta, nota.....	240
\natural.....	760
naturali, armonici.....	356
neo-modern.....	33
<i>neo-modern</i> , stile delle alterazioni.....	33
neo-modern-cautionary.....	33
<i>neo-modern-cautionary</i> , stile delle alterazioni.....	33
neo-modern-voice.....	34
<i>neo-modern-voice</i> , stile delle alterazioni.....	34
neo-modern-voice-cautionary.....	34
<i>neo-modern-voice-cautionary</i> , stile delle alterazioni.....	34
neomensurale.....	455
nera, chiave mensurale.....	459
nessun accordo, simbolo.....	437
neumi quadrati, legatura gregoriana.....	469
\new.....	607
\newSpacingSection.....	592
niente, al, forcella.....	137
no-reset.....	35
no-reset, stile delle alterazioni.....	35
\noBeam.....	101
\noBreak.....	567
nome accordo, insieme ai diagrammi dei tasti.....	393
nome alternativo di un accordo.....	439
nome altezza, altre lingue.....	8

nome del cantante.....	312
nome del personaggio.....	323
nome delle note in turco.....	491
nome di un accordo.....	432, 437
nome di un'altezza.....	1
nome nota, altre lingue.....	8
nome nota, olandese.....	6
nome nota, predefinito.....	6
nome strumento.....	219, 549
nome strumento, abbreviato.....	219
nome strumento, aggiunta in altri contesti.....	221
nome strumento, centrare.....	219
nome strumento, complesso.....	219
nome strumento, modifica.....	221
nomi delle note in arabo.....	487
nomi delle note, mostrare.....	246
\nonArpeggiato.....	321, 350
nonstaff-nonstaff-spacing.....	574
nonstaff-relatedstaff-spacing.....	574
nonstaff-unrelatedstaff-spacing.....	574
\noPageBreak.....	570, 811
\noPageTurn.....	573, 811
\normal-size-sub.....	717
\normal-size-super.....	263, 717
\normal-text.....	717
\normal-weight.....	718
\normalsize.....	231, 264, 718
nota a piè di pagina.....	512
nota a piè di pagina basata su un evento.....	513
nota a piè di pagina basate sul tempo.....	515
nota a piè di pagina nel testo separato.....	518
nota a piè di pagina, nelle espressioni musicali ...	512
nota che attraversa i righi.....	350
nota colorata.....	241
nota colorata in un accordo.....	243
nota di basso, per accordi.....	435
nota doppiamente puntata.....	50
nota fantasma.....	243
nota fantasma, percussioni.....	419
nota in corpo più piccolo.....	222, 226
nota invisibile.....	240
nota muta, percussioni.....	419
nota nascosta.....	240
nota più piccola.....	226
nota puntata.....	50
nota puntata, cambiare il numero di punti.....	51
nota puntata, spostamento orizzontale.....	192
nota smorzata su strumento con tasti.....	408
nota spaziatrice.....	65
nota tra parentesi.....	243
nota trasparente.....	240
nota trasversale ai righi.....	350
nota trasversale al rigo.....	344
nota, altezza predefinita.....	50
nota, continuazione.....	452
nota, divisione.....	84
nota, durata.....	49
nota, durata predefinita.....	49
nota, lunghezza.....	49
nota, nome predefinito.....	6
nota, spaziatura orizzontale.....	592
nota, spostamento.....	188
nota, spostamento orizzontale.....	192
nota, trasposizione di.....	11
notazione grafica.....	269
notazione semplificata.....	42
notazione semplificata, usando numeri.....	43
notazione, dentro il blocco markup.....	270
notazione, dimensione del tipo di carattere.....	231
notazione, spiegare la.....	247
\note.....	760
note a forma variabile.....	44
note head, Aiken, thin variant.....	46
note simultanee e alterazioni.....	36
note, nomi, mostrare.....	246
note, within text, by duration.....	760
note, within text, by log and dot-count.....	761
\note-by-number.....	761
note-event.....	225
Note_heads_engraver.....	84
Note_name_engraver.....	246
\notemode.....	642
noteNameFunction.....	246
NoteNames.....	246
noteNameSeparator.....	246
notina.....	222, 226
notina, chiave.....	18
notina, formattazione.....	226
notina, togliere.....	230
\null.....	265, 782
NullVoice.....	308
\number.....	718
numeri di battuta, alternativi, nelle ripetizioni ...	167
numeri di battuta, con lettere, nelle ripetizioni ...	167
numeri di battuta, controlli.....	118
numeri di corda, orientamento.....	236
numeri di strofa.....	310
\numericTimeSignature.....	71
numero del gruppo irregolare, modifiche del.....	53
numero del gruppo irregolare, non predefinito.....	53
numero della misura, e ripetizioni.....	169
numero di battuta.....	112, 128
numero di battuta, a intervalli regolari.....	114
numero di battuta, allineamento.....	117
numero di battuta, cadenze.....	79
numero di battuta, collisione.....	118
numero di battuta, disposizione a distanza regolare.....	112
numero di battuta, musica in tempo libero.....	79
numero di battuta, nella prima misura.....	113
numero di battuta, stile.....	114
numero di battuta, togliere.....	117
numero di chiamata, sotto il rigo.....	77
numero di corda.....	356, 359
numero di corda vs. numero di diteggiatura.....	359
numero di misura.....	112
numero di misura, stile.....	114
numero di pagina, in numeri romani.....	562
numero di pagina, indicare il primo.....	562
numero di pagina, numerazione automatica.....	562
numero di pagina, sopprimere.....	562
numero di ripetizione, modificare.....	169
numero, in notazione semplificata.....	43
nuova spaziatura nel corso di un brano.....	592
nuovo contesto.....	607
nuovo rigo.....	202
nuvoletta.....	247
nuvoletta di aiuto.....	247

O

<code>\octaveCheck</code>	10, 811
<code>offset</code>	634
<code>\offset</code>	634, 811
oggetti variabili	801
oggetto colorato	241
oggetto grafico	800
oggetto grafico incorporato	268
oggetto grafico, disegnare	268
oggetto grafico, includere	268
oggetto immutabile	800
oggetto Scheme	802
oggetto, colorare	653
oggetto, rotazione	659
oggetto, sovrascrivere	653
oggetto, visibilità del	652
<code>\omit</code>	652, 811
<code>\on-the-fly</code>	510, 782
<code>\once</code>	629, 631, 636, 671, 811
<code>\oneVoice</code>	184
<code>\open</code>	129, 356, 420, 794
open bongo	796
open conga	796
open high hat	796
open timbale	796
'open', articolazione	794
OpenType, font feature	719
opera	322
operazione, inversione	14
operazione, inversione modale	16
operazione, modale	15
operazione, retrogradazione	15
operazione, trasposizione	15
oratorio	319
orchestra, archi	355
orchestra, musica	323
orchestra, notazione per	323
ordine di stampa	653
ordine verticale degli script	131
organo, segno del pedale	129, 794
orientamento dei diagrammi dei tasti, modifica	389
orientamento dei numeri di corda	236
orientamento del dito che pizzica	236
orientamento delle diteggiature	236
<code>\oriscus</code>	469, 474
orizzontale, allineamento, testo vocale	298
orizzontale, spaziatura	590
ornamento	122
ornamento, down pralltriller	793
ornamento, gruppetto	129
ornamento, gruppetto barrato	129
ornamento, gruppetto Haydn	129
ornamento, gruppetto rovesciato	129
ornamento, gruppo ritardato	131
ornamento, Haydn turn	793
ornamento, lower mordent	793
ornamento, mordent	793
ornamento, mordent, lower	793
ornamento, mordent, upper	793
ornamento, mordente	129
ornamento, mordente inferiore	129
ornamento, mordente inverso	129
ornamento, mordente inverso inferiore	129
ornamento, mordente inverso lungo	129

ornamento, mordente inverso, superiore	129
ornamento, mordente superiore	129
ornamento, pralltriller	793
ornamento, pralltriller, down	793
ornamento, pralltriller, long	793
ornamento, pralltriller, up	793
ornamento, reverse turn	793
ornamento, signum congruentiae	129
ornamento, slash turn	793
ornamento, trill	793
ornamento, trillo	129
ornamento, turn	793
ornamento, up pralltriller	793
ornamento, upper mordent	793
ossia	213, 218
ossia, posizionamento	216
ottava	24
<code>\ottava</code>	24, 811
ottava assoluta	1
ottava relativa	2
ottava relativa, e accordi	4
ottava relativa, e trasposizione	5
ottava, controllo	10
ottava, estensore, modifica dell'inclinazione	26
ottava, per una sola voce	26
ottava, testo	25
ottavation	25
ottavation-numbers	25
ottavation-ordinals	25
ottavation-simple-ordinals	25
ottavationMarkups	25
ottavazione	24
ottomana, musica	491
outer-margin	559
output, definizione	605
output-count	801
output-def	801
output-suffix	801
outside-staff-horizontal-padding	589
outside-staff-padding	589
outside-staff-priority	589
<code>\oval</code>	749
<code>\overlay</code>	737
<code>\override</code>	630, 634, 782
<code>\override-lines</code>	789
<code>\overrideProperty</code>	634, 811
<code>\overrideTimeSignatureSettings</code>	72, 812
overriding property, within text markup	782
<code>\overtie</code>	720

P

<code>\p</code>	132
<code>\pad-around</code>	268, 737
<code>\pad-markup</code>	268, 737
<code>\pad-to-box</code>	268, 738
<code>\pad-x</code>	268, 738
<code>\pad-x-left</code>	738
<code>\pad-x-right</code>	738
padding	627
padding intorno al testo	268
padding text	737
padding text horizontally	738
page breaking, manual	570
page-breaking	561

page-breaking-system-system-spacing.....	561	parte a due.....	194
page-count.....	561	parte solista.....	194
\page-link.....	782	\partial.....	78, 161, 163, 813
page-number-type.....	562	partitura completa.....	322, 323
\page-ref.....	521, 783	partitura polimetrica.....	614
page-spacing-weight.....	562	partitura senza i righi vuoti.....	217
\pageBreak.....	570, 812	partitura vocale.....	322
\pageTurn.....	573, 812	partitura vocale, aggiungere suggerimenti.....	327
pageTurnMinimumRepeatLength.....	573	partitura, dentro il blocco markup.....	272
pageTurnMinimumRestLength.....	573	parziale, misura.....	78
pagina, formato.....	553	parziale, misura, negli inni.....	340
pagina, interruzione.....	595	\path.....	750
pagina, interruzione, gestione con una voce		path, drawing.....	750
apposita.....	569	\pattern.....	783
pagina, orientamento.....	554	pausa.....	63
\palmMute.....	812	pausa antica.....	462
\palmMuteOn.....	812	pausa d'intero.....	63, 66
Pango.....	274	pausa di breve.....	63
panning, posizione nel MIDI.....	548	pausa di lunga.....	63
\paper.....	497, 553	pausa di maxima.....	63
paper-height.....	555	pausa ecclesiastica.....	68
paper-width.....	558	pausa intera per una misura intera.....	66
parallela, musica.....	199	pausa invisibile.....	65
\parallelMusic.....	199, 812	pausa multipla.....	63, 66
parentesi.....	250	pausa multipla con testo a margine.....	67
parentesi del gruppo irregolare, visibilità.....	54	pausa multipla, attaccare fermata.....	67
parentesi dell'arpeggio trasversale ai righi.....	350	pausa multipla, attaccare testo.....	67
parentesi della volta.....	169	pausa multipla, contrazione.....	67
parentesi della volta, accorciare.....	165	pausa multipla, e diteggiature.....	70
parentesi della volta, con testo.....	171	pausa multipla, espansione.....	67
parentesi della volta, e legatura di valore.....	59	pausa multipla, numerazione.....	68
parentesi della volta, in altri righi.....	166	pausa multipla, posizionamento.....	69
parentesi di analisi musicologica.....	250	pausa multipla, script.....	67
parentesi di analisi, con etichetta.....	251	pausa multipla, testo a margine.....	70
parentesi di raggruppamento delle note.....	250	pausa spaziatrice.....	65
parentesi graffa, annidamento di.....	207	pausa, collisioni di.....	70
parentesi orizzontale.....	250	pausa, divisione.....	84
parentesi quadra.....	243	pausa, inserire le durate.....	63
parentesi quadra verticale.....	203	pausa, segno.....	147
parentesi quadra, all'inizio di un gruppo di righi.....	205	pausa, specificare la posizione verticale.....	63
parentesi quadra, annidamento di.....	207	pausa, spostamento automatico.....	188
parentesi uncinata (o angolare).....	178	pausa, stile.....	63
parentesi, all'inizio di un rigo singolo.....	205	pause, condensare.....	70
parentesi, fraseggio.....	250	PDF, metadati.....	511
parentesi, stile nell'arpeggio attraverso il rigo.....	158	pedal high hat.....	796
parentesi, verticale, contrassegnare voci divise.....	321	pedale del pianoforte.....	351
\parenthesize.....	243, 750, 812	pedale dell'arpa.....	354
parlato.....	329	pedale di risonanza.....	351
parlato, testa di nota.....	41	pedale di risonanza, stile del.....	352
Parmesan, font.....	693	pedale high hat (charleston).....	416
parser.....	801	pedale sostenuto.....	351
parser variable.....	801	pedale tonale.....	351
\partCombine.....	194, 308, 812	pedale, high hat (charleston).....	129
\partCombine e testo cantato.....	308	pedale, indicazione, misto.....	352
\partCombine e testo vocale.....	197	pedale, indicazione, parentesi.....	352
\partCombineApart.....	196	pedale, indicazione, testo.....	352
\partCombineAutomatic.....	196	pedale, segno.....	129, 794
\partCombineChords.....	196	pedale, segno, heel.....	794
\partCombineDown.....	813	pedale, segno, organo.....	794
\partCombineForce.....	813	pedale, segno, toe.....	794
partCombineListener.....	801	pedalSustainStyle.....	352
\partCombineSoloI.....	196	pedice.....	263
\partCombineSoloII.....	196	percent.....	173
\partCombineUnisono.....	196	percentuale, ripetizione.....	173
\partCombineUp.....	813	percussioni.....	412, 414

percussioni personalizzate	416	portato	130
percussioni, chiave	412	\portato	129, 793
percussioni, chiavi di	709	“portato”, articolazione	129, 793
percussioni, dead note	419	posizionamento della parentesi quadra del gruppo irregolare	52
percussioni, ghost note	419	posizionamento verticale delle dinamiche	135
percussioni, nota fantasma	419	posizionamento, diteggiatura mano destra	406
percussioni, nota morta	419	posizionamento, testo vocale	295
percussioni, nota muta	419	posizione verticale dei grob	643
percussioni, nota non suonata	419	posizione, alterazione del basso continuo	448
percussioni, varie	796	posizione, ossia	216
personaggio, nome	323	posizione, pausa multipla	69
personalizzazione del nome di un accordo	439	posizione, testo vocale	216
\pes	475	post-events	793
Petrucchi	454, 455	postscript	269
Petrucchi, chiave	459, 709	\postscript	269, 752
\phrasingSlurDashed	145	power chord	409
\phrasingSlurDashPattern	146, 813	\pp	132
\phrasingSlurDotted	145	\ppp	132
\phrasingSlurDown	145	\pppp	132
\phrasingSlurHalfDashed	146	\ppppp	132
\phrasingSlurHalfSolid	146	\prall	129, 793
\phrasingSlurNeutral	145	\pralldown	129, 793
\phrasingSlurSolid	145	\prallmordent	129, 793
\phrasingSlurUp	145	\prallprall	129, 793
\phrygian	22	‘pralltriller, down’, ornamento	793
piè di pagina	499	‘pralltriller, long’, ornamento	793
piano	32	‘pralltriller’, ornamento	793
<i>piano</i> , stile delle alterazioni	32	‘pralltriller, up’, ornamento	793
piano-cautionary	32	\prallup	129, 793
<i>piano-cautionary</i> , stile delle alterazioni	32	\preBend	366, 814
pianoforte, musica, centrare le dinamiche	343	\preBendHold	366, 814
pianoforte, pedale	351	predefinedDiagramTable	401
pianoforte, rigo	343	\predefinedFretboardsOff	403
pianoforte, rigo per	203	\predefinedFretboardsOn	403
PianoStaff	343, 347	predefinita, altezza	50
piatti, vari	796	predefinito, offset	634
piatto China	796	prima volta	161
piatto crash	796	principianti, musica	42
piatto ride	796	print-all-headers	562
piatto splash	796	print-first-page-number	562
Pitch_squash_engraver	86	print-page-number	562
\pitchedTrill	159, 813	printAccidentalNames	246
pitchnames	801, 810	printNotesLanguage	246
pizzicato a schiocco	357	printOctaveNames	246
pizzicato alla Bartók	357	prob	802
place-fret	769	\property-recursive	783
placing horizontal brackets, around text	749	\propertyOverride	814
placing parentheses, around text	750	\propertyRevert	814
placing vertical brackets, around text	744	\propertySet	814
PNG image	749	\propertyTweak	814
\pointAndClickOff	813	\propertyUnset	814
\pointAndClickOn	813	proprietà	628
\pointAndClickTypes	813	proprietà condivisa	800
polifonia su un rigo singolo	184	proprietà dei grob	630
polifonia, in intavolatura	370	proprietà dell’oggetto	802
polifonia, testo cantato condiviso	308	proprietà di contesto predefinita, modifica	615
polifonia, voci ulteriori	191	proprietà immutabile	800
polimetria	614	proprietà variabili	801
polimetria, indicazione	81	\pt	644
“pollice”, diteggiatura	129, 236, 794	pull off	377
\polygon	751	puntata, nota, spostamento orizzontale	192
\polymetric	813	punte delle travature	102
\popContextProperty	814	punteggiatura, nel testo vocale	281
portamenti indeterminati verso il basso (cadute) e verso l’alto	148	punti di controllo, modifica con \tweak	633

punto coronato	794
punto di aumentazione, cambiare numero	51
punto di controllo, curva Bézier	666
punto, nota	50
\push-to-tag	783
\pushContextProperty	815
\pushToTag	530, 815
\pushToTagMarkup	815
\put-adjacent	739
putting space around text	737

Q

q, ripetizione accordo	180, 362
\qr-code	784
QR code	784
quarto di tono	6
quarto di tono, alterazione	8
quarto di tono, intavolatura	381
\quilisma	469, 474
quotedCueEventTypes	225
quotedEventTypes	225
\quoteDuring	222, 226, 815

R

r	63
R	66
ragged-bottom	555
ragged-last	559, 595
ragged-last-bottom	555
ragged-right	559, 595
raggruppamento di gruppi irregolari	52
\raise	265, 739
\raiseNote	436, 815
raising text	739
Ratisbona, Editio	455
\reduceChords	87, 815
referencing page label, in text	788
referencing page number, in text	782, 783
registro della fisarmonica	352
relativa, ottava	2
\relative	2, 5, 14, 348, 815
relativo	2
\remove	613
remove-grace-property	124
\remove-with-tag	784
\RemoveAllEmptyStaves	217, 821
\RemoveEmptyStaves	217, 821
\removeFromTagGroup	815
\removeWithTag	527, 815
\repeat	161
\repeat percent	173
\repeat tremolo	176
repeatCommands	169
repeatCountVisibility	175
\repeatTie	59, 163, 304
\replace	720
\replace-with-tag	785
\replaceWithTag	815
\replaceWithTagMarkup	815
\resetRelativeOctave	5, 815
\resetTagGroup	815
\resetTagGroups	816

respiro, segno	147
respiro, simbolo, elenco di	796
respiro, simbolo, modifica	147
\responsum	470, 816
\rest	63, 762
rest, within text, by duration	762
rest, within text, by log and dot-count	762
\rest-by-number	762
rest-event	225
restNumberThreshold	68
restrainOpenStrings	361
retrogradazione, trasformazione	15
\retrograde	15, 816
'reverse turn', ornamento	793
\reverseturn	129, 793
\revert	631
\revertTimeSignatureSettings	73, 816
\rfz	132
rgb, colore	242
rgb-color	242
\rhythm	763
rhythm, in text	763
RhythmicStaff	202
ricopiate, ripetizioni	171
ride bell	796
ridimensionamento dei righi	213
righe, gruppo di	203
\right-align	264, 739
right-aligning text	739
\right-brace	785
\right-column	740
right-margin	558
\rightHandFinger	405, 816
rigo del pianoforte	343
rigo Gregoriano per trascrizione	202
rigo multiplo	203
rigo per batteria	202
rigo per coro	203
rigo per percussioni	202
rigo per pianoforte	203
rigo ritmico	202
rigo temporaneo	213, 217
rigo vuoto	217
rigo, annidato	207
rigo, batteria	202
rigo, cambio automatico	347
rigo, cambio del, linea	348
rigo, cambio manuale	344
rigo, distanza	574
rigo, impostare la dimensione	565
rigo, nascondere	217
rigo, nuovo	202
rigo, percussioni	202
rigo, ridimensionamento del	213
rigo, segno di metronomo sotto	77
rigo, simbolo del	210
rigo, singolo	202
rigo, singolo, con parentesi	205
rigo, strumenti a tastiera	343
rigo, strumenti dotati di tasti	343
rilegatura	559
rimozione, negli accordi	435
rinascimentale, musica	206
ripetere il testo vocale, con finali alternativi	303
ripetere una legatura di valore	59

ripetizione alternata	171
ripetizione annidata	169
ripetizione con percentuale	173
ripetizione con volte, sotto gli accordi	443
ripetizione della misura	173
ripetizione e glissando	152
ripetizione normale	161
ripetizione ricopiata	171
ripetizione, alternativa	171
ripetizione, ambigua	169
ripetizione, breve	173
ripetizione, con anacrusi	163
ripetizione, con controlli di battuta	163
ripetizione, con finali alternativi	161
ripetizione, con legature di valore	163
ripetizione, con segno	164
ripetizione, contatore con segno percentuale	174
ripetizione, doppia, stile per la volta	167
ripetizione, e glissandi	169
ripetizione, e legatura di portamento	169
ripetizione, e numero della misura	169
ripetizione, e testo vocale	299
ripetizione, fine	169
ripetizione, informazioni sul tempo	169
ripetizione, inizio	169
ripetizione, manuale	169
ripetizione, numeri di battuta alternativi	167
ripetizione, numeri di battuta con lettere	167
ripetizione, percentuale, visibilità del conto	175
ripetizione, tremolo	176
ripetizione, unfold	171
ripetizione, usando q	362
ripetizione, uso di q	180
ripetizioni con segni percentuali isolati	175
ripetizioni nel MIDI	545
ripetuti, accordi, soppressione	438
ripristinare le proprietà predefinite di un'indicazione di tempo	73
ripristinare una modifica	631
riquadro intorno all'oggetto	645
risonanza, pedale di	351
ritmo di accompagnamento, mostrare	86
ritmo, mostrare la melodia	86
ritocco	632
ritornello	107
riverbero nel MIDI	548
\romanStringNumbers	356, 359
romboidale, testa di nota	356
\rotate	740
rotating text	740
rotazione di un oggetto	659
\rounded-box	268, 753
rullante	796
rullante acustico	796
rullante elettrico	796

S

s	65
sù, direzione (˘)	643
Sacred Harp, testa di nota	44
\sacredHarpHeads	44
\sacredHarpHeadsMinor	45
saltare le note nel teso vocale	303
\sans	721
SATB	319
scalable vector graphics, output	535
scalare la durata	57
\scale	754
\scaleDurations	57, 81, 816
scaling markup	754
scaling text	741
scambio di rigo	348
scelta della dimensione del tipo di carattere (notazione)	231
Scheme, contenitore impuro	672
Scheme, contenitore puro	672
Scheme, oggetto	802
schiocco, pizzicato	357
scivolare, diteggiatura	237
\score	493, 497, 764
\score-lines	789
score-markup-spacing	557
score-system-spacing	557
scoreTitleMarkup	507
script	793
script su pausa multipla	67
script, ordine verticale	131
scritta	253
scrivere la musica in parallelo	199
seconda volta	161
\sectionLabel	816
segni di tempo, per glissando	151
segni per la conduzione	98
segno	107, 121
\segno	129, 764, 795
segno del pedale	129
segno del pedale dell'organo	129
segno del pedale, dito del piede	129
segno del pedale, tallone	129
segno di bequadro	6
segno di bequadro, supplementare, impedire	7
segno di chiamata	119
segno di chiamata manuale	120
segno di chiamata personalizzato	120
segno di chiamata, formato	120
segno di chiamata, stile	120
segno di coda, variante	129
segno di metronomo con testo personalizzato	78
segno di modifica dell'ottava	1
segno di spunta	147
segno di tremolo	177
segno separatore del sistema	209
segno testuale, allineamento verticale	138
segno, coda	129, 795
segno, coda, variante	795
segno, con ripetizioni	164
segno, di chiamata, formato	120
segno, di chiamata, stile	120
segno, numero di chiamata, sotto il rigo	77
“segno”, segno	129, 795

segno, snappizzicato	129, 794	simultanea, legatura di portamento	142
segno, su ogni rigo	257	sincronizzazione degli abbellimenti	126
segno, sulla stanghetta	256	\single	516, 637, 816
segno, variante coda	795	sintassi di LilyPond	801
\segnoMark	816	sintassi di markup	260
segui voce	348	sistema	203
self-alignment-X	574	sistema, delimitatore di inizio annidato	207
semai, forma	489	sistema, segno separatore	209
semi-bemolle	6, 9	\skip	65
semi-diesis	6, 9	\skip	65, 303, 816
“semiaperto”, articolazione	129	skipTypesetting	534
semiaperto, high hat (charleston)	129	‘slash turn’, ornamento	793
semibemolle, simbolo, aspetto	487	slashChordSeparator	441
\semicirculus	468, 795	slashed digit	785
‘semicirculus’, articolazione gregoriana	468	\slashed-digit	785
‘semicirculus’, articolazione Gregoriana	795	\slashedGrace	122, 816
\semiflat	765	\slashSeparator	562
\semiGermanChords	439	\slashturn	129, 793
\semisharp	765	slide, in intavolatura	375
semitrasparenza	241	slur-event	225
separatore dell'accordo, modifica	443	\slurDashed	143
\serif	721	\slurDashPattern	143, 816
sesqui-bemolle	9	\slurDotted	143
sesqui-diesis	9	\slurDown	142
\sesquiflat	765	\slurHalfDashed	143
\sesquisharp	765	\slurHalfSolid	143
\set	91, 628, 634	\slurNeutral	142
set-global-fonts	278	\slurSolid	143
set-global-staff-size	565	\slurUp	144
settimana maggiore, accordo, formattazione	442	\small	231, 264, 721
setting extent of text object	788	\smallCaps	722
setting horizontal text alignment	731	\smaller	262, 264, 722
setting subscript, in standard font size	717	smob	802
setting superscript, in standard font size	717	\snappizzicato	129, 794
\sf	132	“snappizzicato”, segno	129, 794
\sff	132	Sol tenore, chiave di	709
\sfz	132	Sol, chiave di	17, 709
\shape	667, 816	Solesmes	455
\sharp	765	soppressione degli accordi ripetuti	438
\shiftDurations	816	sopprimere accordi ripetuti	400
\shiftOff	188	soprano, chiave di	17, 709
\shiftOn	188	sos	351
\shiftOnn	188	sostenuto, pedale	351
\shiftOnnn	188	\sostenutoOff	351
short-indent	220, 560	\sostenutoOn	351
\shortfermata	129, 794	\sourcefileline	498
show-available-fonts	277	\sourcefilename	498
showFirstLength	534, 801	Southern Harmony, testa di nota	44
\showKeySignature	422	\southernHarmonyHeads	44
showLastLength	534, 801	\southernHarmonyHeadsMinor	45
\showStaffSwitch	348	sovrascrivere un oggetto	653
sidestick	796	\sp	132
‘signum congruentiae’, notazione mensurale	795	spacing	591
‘signum congruentiae’, ornamento	129	Span_stem_engraver	350
\signumcongruentiae	129, 795	spanner	802
sillabe, durata automatica	284	spaziatura del testo vocale	297
simboli di settimana maggiore	442	spaziatura nuova nel corso di un brano	592
simbolo arabo per semibemolle	487	spaziatura orizzontale	590
simbolo del rigo	210	spaziatura orizzontale, sovrascrivere la	672
simbolo del rigo, impostazione del	645	spaziatura rigorosa e abbellimenti	125
simbolo di respiro, modifica	147	spaziatura verticale	574, 595
simbolo non musicale	269	spaziatura, visualizzazione della formattazione ...	601
simbolo speciale di arpeggio	155	spazio bianco	498
\simple	721	spazio, dentro i sistemi	574
simple text string, with tie characters	766	spazio, nel testo vocale	281, 288

spazio, tra i righi	574	stencil.....	802
spezzato, arpeggio.....	154	\stencil.....	786
spostamento automatico di una pausa	188	stencil, soppressione	652
spostamento orizzontale di una nota	192	stereo, bilanciamento nel MIDI.....	548
spostamento relativo (offset).....	634	stile alternativo delle note brevi.....	50
spostare una nota	188	stile del pedale di risonanza	352
spostare una voce	188	stile del segno di chiamata.....	120
\spp.....	132	stile delle alterazioni <i>default</i>	30
Sprechgesang.....	329	stile delle alterazioni <i>modern</i>	31, 32
spunta, segno di.....	147	stile delle alterazioni <i>modern-cautionary</i>	31
staccatissimo.....	130	stile delle alterazioni <i>neo-modern</i>	33
\staccatissimo.....	129, 793	stile delle alterazioni <i>neo-modern-cautionary</i>	33
“staccatissimo”, articolazione	129, 793	stile delle alterazioni per insegnamento (<i>teaching</i>)	35
staccato.....	130	stile delle alterazioni <i>voice</i>	31
\staccato.....	129, 793	stile delle alterazioni, choral	33
“staccato”, articolazione	129, 793	stile delle alterazioni, choral-cautionary	33
stacking text, in a column	727	stile delle alterazioni, <i>dodecaphonic</i>	34
staff brace, in markup.....	781	stile delle alterazioni, dodecaphonic-first	35
staff-affinity.....	574	stile delle alterazioni, dodecaphonic-no-repeat	34
staff-padding.....	237	stile delle alterazioni, forget	35
\staff-space.....	644	stile delle alterazioni, <i>modern-voice-cautionary</i>	32
staff-staff-spacing.....	574	stile delle alterazioni, <i>neo-modern-voice</i>	34
Staff.midiInstrument.....	549	stile delle alterazioni, <i>neo-modern-voice-cautionary</i>	34
Staff_collecting_engraver.....	257	stile delle alterazioni, no-reset	35
Staff_symbol_engraver.....	217	stile delle alterazioni, <i>piano</i>	32
staffgroup-staff-spacing	574	stile delle alterazioni, <i>piano-cautionary</i>	32
\staffHighlight.....	817	stile delle dinamiche del testo.....	139
StaffSymbol.....	210	stile delle teste di nota	707
stampa del nome di un accordo	437	stile di alterazione.....	29
stampare carattere riservato	260	stile di alterazione predefinito.....	29
stampare carattere speciale	260	stile di chiave	459
stampo.....	802	stile di indicazione del pedale	352
stampo, soppressione	652	stile di voce	187
stanghetta	105	stile moderno dell’alterazione	31
stanghetta di chiusura.....	105	stile, doppia ripetizione per la volta.....	167
stanghetta doppia	105	stile, legatura di portamento.....	143
stanghetta invisibile	105	stile, pause	63
stanghetta manuale	105	stile, teste di nota	41
stanghetta predefinita, modifica	112	stopAcciaccaturaMusic.....	124
stanghetta, cadenze.....	79	stopAppoggiaturaMusic.....	124
stanghetta, definire.....	110	stopGraceMusic.....	124
stanghetta, evitata dal testo vocale	298	\stopGroup	250
stanghetta, forcella, terminare a	136	\stopped	129, 416, 420, 794
stanghetta, in ChordNames	443	‘stopped’, articolazione	794
stanghetta, musica in tempo libero.....	79	\stopStaff	210, 213, 217
stanghetta, simboli sulla	256	\stopTrillSpan.....	158
stanghetta, soppressione	658	\storePredefinedDiagram.....	396, 401, 817
stanghetta, tra i righi.....	206	strappato	377
start-repeat	169	strato.....	653
startAcciaccaturaMusic.....	124	strictBeatBeaming	97
startAppoggiaturaMusic.....	124	\string-lines.....	790
startGraceMusic.....	124	stringNumberOrientations	236
\startGroup	250	\stringTuning.....	379, 817
\startStaff	210, 213	stringTunings	378, 392
\startTrillSpan.....	158	strofa, numeri di	310
\stdBass.....	773	strokeFingerOrientations	236, 406
\stdBassIV	774	\stropha.....	469, 474
\stdBassV	775	strumenti a tastiera, rigo	343
\stdBassVI	776	strumenti dotati di tasti, rigo.....	343
stem-spacing-correction	591	strumento a fiato.....	419
\stemDown	244	strumento a tasti, diteggiatura della mano destra.....	405
stemLeftBeamCount	102	strumento aerofono.....	419
\stemNeutral	244	strumento con tasti, accordature predefinite.....	378
stemRightBeamCount	102	strumento con tasti, armonici.....	408
\stemUp.....	244	strumento con tasti, forma di un accordo	396

strumento con tasti, indicare la posizione e il barré .	407
strumento con tasti, nota smorzata .	408
strumento traspositore .	11, 27
strumento, nome .	219, 549
strumento, nome complesso .	219
\strut .	786
strumento, equalizzazione nel MIDI, cambiare predefinita .	542
\styledNoteHeads .	817
\sub .	263, 722
subbasso, chiave di .	17
subdivideBeams .	96
subscript text .	722
suddividere le travature .	96
suggerimento musicale .	325
suggerimento, nella partitura vocale .	327
suggestAccidentals .	131, 463
\super .	263, 722
superscript text .	722
sus .	435
\sustainOff .	351
\sustainOn .	351
SVG, output .	535
swing .	763
system-count .	560
system-separator-markup .	562
system-system-spacing .	557
systems-per-page .	560

T

\tabChordRepeats .	362, 817
\tabChordRepetition .	817
tabella alternativa, tastiera .	401
\tabFullNotation .	361
tablatura .	202, 358
tablatura e armonici .	364
tablatura per banjo .	358, 410
tablatura per chitarra .	358
tablatura per liuto .	411
tablatura personalizzata .	378
tablatura, accordature personalizzate .	379
tablatura, accordature predefinite .	378
tablatura, banjo .	378
tablatura, base .	360
tablatura, basso .	378
tablatura, basso elettrico .	378
tablatura, chiave .	380
tablatura, chiavi di .	709
tablatura, chitarra .	378
tablatura, contrabbasso .	378
tablatura, e armonici .	371
tablatura, e bending .	366
tablatura, e gambi .	370
tablatura, e polifonia .	370
tablatura, e slide .	375
tablatura, e travature .	370
tablatura, glissando di accordi .	376
tablatura, mandolino .	378
tablatura, martellato .	377
tablatura, microtono .	381
tablatura, predefinita .	360
tablatura, quarto di tono .	381
tablatura, strappato .	377

tablatura, ukulele .	378
tablatura, viola .	378
tablatura, violino .	378
tablatura, violoncello .	378
\table .	790
\table-of-contents .	524, 791
TabStaff .	202, 360
TabVoice .	360
tag .	527
\tag .	527, 786, 817
\tag-list .	791
\tagGroup .	530, 817
\tagGroupRef .	817
tagliata, testa di nota .	47
taglio addizionale .	210
taglio addizionale, funzionamento interno .	210
taglio addizionale, modificare .	210
“tallone”, segno del pedale .	129
tam tam .	796
tamburello .	796
tante voci .	188
\taor .	422
taqasim .	489
tasti, diagramma personalizzato .	389
tasti, diagramma personalizzato, aggiungere .	395
tasti, diagramma, trasposizione .	394
tasti, diagrammi automatici .	403
tasti, strumento a, diteggiatura della mano destra .	405
tastiera del diagramma dei tasti, personalizzata .	397
tastiera, definire predefinita .	398
tastiera, diagramma .	382
tastiera, diagramma personalizzato .	382
tastiera, tabella alternativa .	401
tasto .	361
tasto, strumento con, armonici .	408
tasto, strumento con, nota smorzata .	408
teaching .	35
teaching, stile delle alterazioni .	35
\teeny .	231, 264, 723
tempo .	71, 75
\tempo .	75
tempo composto, indicazione .	83
tempo polimetrico .	81
tempo polimetrico, con travature .	81
tempo, in una partitura .	128
tempo, modifica, senza il segno di metronomo .	77
tempo, stile .	71
tempo, with rhythm .	763
\temporary .	636, 671, 817
tenere la musica etichettata .	527
tenore, chiave di .	17, 709
tenuto .	130
\tenuto .	129, 793
“tenuto”, articolazione .	129, 793
terzina .	51
terzina, formattazione della .	52
testa di nota .	231
testa di nota antica .	460
testa di nota romboidale .	356
testa di nota, Aiken .	44
testa di nota, armonico .	41
testa di nota, barrata .	41
testa di nota, chitarra .	41
testa di nota, Christian Harmony .	44
testa di nota, esercizio .	42

testa di nota, facile da suonare.....	42	testo, dimensione.....	262
testa di nota, forma.....	44	testo, elemento non vuoto.....	253
testa di nota, Funk.....	44	testo, esteso su più pagine.....	273
testa di nota, Harmonia Sacra.....	44	testo, fuori dal margine.....	254
testa di nota, improvvisazione.....	47	testo, in colonne.....	259, 266
testa di nota, notazione semplificata.....	42	testo, incorniciatura.....	268
testa di nota, parlato.....	41	testo, indicazione.....	256
testa di nota, romboidale.....	41	testo, inserimento.....	281
testa di nota, Sacred Harp.....	44	testo, nella parentesi della volta.....	171
testa di nota, Southern Harmony.....	44	testo, padding.....	268
testa di nota, speciale.....	41	testo, posizionamento.....	295
testa di nota, stile.....	707	testo, \skip.....	65
testa di nota, stili.....	41	testo, sulla stanghetta.....	256
testa di nota, tagliata.....	47	testo, tenerlo dentro il margine.....	254
testa di nota, Walker.....	44	text.....	352
teste di nota antiche.....	476	text column, left-aligned.....	736
testo a capo automaticamente.....	267	text column, right-aligned.....	740
testo a due colonne.....	259	text, with a tie above.....	720, 723
testo a margine.....	260	text, with a tie below.....	723, 724
testo a margine, pausa multipla.....	70	\text-accidental.....	765
testo al livello superiore.....	258	\text-doubleflat.....	766
testo cantato, condiviso tra voci.....	308	\text-doublesharp.....	766
testo cantato, su abbellimenti.....	313	\text-flat.....	766
testo condizionale.....	510	\text-natural.....	766
testo dell'ottava.....	25	\text-sharp.....	766
testo formattato su più pagine.....	273	\textEndMark.....	818
testo giustificato.....	267	\textLengthOff.....	68, 70, 254
testo indipendente.....	258	\textLengthOn.....	68, 70, 138, 254
testo multilinea.....	266	\textMark.....	818
testo separato.....	258	\textSpannerDown.....	255
testo su pausa multipla.....	67	\textSpannerNeutral.....	255
testo tra virgolette.....	253	\textSpannerUp.....	255
testo tra virgolette, in modalità markup.....	260	\thumb.....	129, 236
testo vocale alternato.....	307	'thumb', diteggiatura.....	794
testo vocale diviso.....	307	\tie.....	723
testo vocale e melodie.....	284	tie, above text.....	720, 723
testo vocale, allineamento a una melodia.....	282	tie, below text.....	723, 724
testo vocale, allineamento con una melodia sporadica.....	612	TieColumn.....	61
testo vocale, allineamento orizzontale.....	298	\tied-lyric.....	766
testo vocale, aumentare lo spazio tra.....	297	\tieDashed.....	60
testo vocale, citazione.....	281	\tieDashPattern.....	60, 818
testo vocale, con travatura.....	91	\tieDotted.....	60
testo vocale, e markup.....	282	\tieDown.....	60
testo vocale, e note legate.....	304	\tieHalfDashed.....	60
testo vocale, evitare le stanghetta.....	298	\tieHalfSolid.....	60
testo vocale, formattazione.....	281	\tieNeutral.....	60
testo vocale, mantenere dentro il margine.....	298	\tieSolid.....	60
testo vocale, posizionamento.....	216	\tieUp.....	60
testo vocale, punteggiatura.....	281	tieWaitForNote.....	61
testo vocale, ripetizioni.....	299	timbales.....	796
testo vocale, ripetizioni con finali alternativi.....	303	\time.....	71, 91, 818
testo vocale, saltare le note.....	303	\timeAbbrev.....	83, 818
testo vocale, spazi.....	281	\times.....	818
testo vocale, tenerlo dentro il margine.....	254	timeSignature.....	81
testo vocale, uso delle variabili.....	293	\tiny.....	231, 264, 723
testo, allineamento.....	264	tipi di carattere disponibili, elenco.....	277
testo, allineamento orizzontale.....	264	tipi di carattere, famiglia.....	263
testo, allineamento verticale.....	265	tipo di carattere.....	274, 693, 799
testo, altre lingue.....	253	tipo di carattere (notazione), dimensione standard.....	235
testo, assegnato a una voce.....	184	tipo di carattere, cambiare.....	261
testo, centrare sulla pagina.....	266	tipo di carattere, dimensione.....	262
testo, comando di allineamento.....	267	tipo di carattere, impostare la dimensione.....	565
testo, decorazione.....	268	tipo di carattere, modificarli per l'intero documento.....	277
testo, dentro le legature di portamento.....	144	tipo di carattere, ridimensionamento.....	231

tipo di carattere, trovare disponibilità	277	travatura manuale	88
titolo	499	travatura manuale, abbellimenti	101
\tocItem	524, 818	travatura manuale, scorciatoia per impostare la	
\tocItemWithDotsMarkup	522	direzione	101
\toe	129, 794	travatura trasversala al rigo	344
‘toe’, segno del pedale	794	travatura, cadenze	79
togliere i numeri di battuta	117	travatura, con melisma	89
togliere la musica etichettata	527	travatura, con tempi polimetrici	81
togliere le citazioni in corpo più piccolo	230	travatura, con testo vocale	91
tom	796	travatura, estremità, in una partitura	99
tom alto	796	travatura, estremità, in voci multiple	99
tom basso	796	travatura, gruppo irregolare, interruzione di linea ..	55
tom medio	796	travatura, in intavolatura	370
tonale, pedale	351	travatura, interruzioni di linea	89
top-margin	555	travatura, musica in tempo libero	79
top-markup-spacing	557	travatura, \partCombine con \autoBeamOff	90
top-system-spacing	557	travatura, personalizzazione delle regole	88
toplevel-bookparts	801	travatura, punta	102
toplevel-scores	801	travatura, suddivisione	96
tracce MIDI	545	travature manuali	101
transizione delle vocali	292	tre corde	351
\translate	265, 740	\treCorde	351
\translate-scaled	265, 741	tremolo	176
translating text	740, 741	tremolo	176
\transparent	786	tremolo tra due righe	177
\transpose	5, 11, 14, 818	tremolo, segno	177
\transposedCueDuring	229, 818	triade	432
\transposition	27, 222, 819	\triangle	269, 754
trascrizione di musica mensurale	206	triangolo	796
trascrizione moderna, della musica gregoriana	336	\trill	129, 158, 793
trascrizione, da mensurale a moderno	483	‘trill’, ornamento	793
trasformazione retrograda	15	trillo	158
trasparente, nota	240	trillo con alterazione	159
trasparenti, rendere gli oggetti	652	trillo con altezza	159
trasparenza, semi	241	trillo con notina	159
trasporre	11	trillo con notina e alterazione	159
traspositore, strumento	11	“trillo”, ornamento	129
trasposizione	11	trovare i tipi di carattere disponibili	277
trasposizione del diagramma dei tasti	394	true (#t)	822
trasposizione dell’ottava	18	\tuplet	51, 81, 819
trasposizione modale	15	\tupletDown	52
trasposizione opzionale dell’ottava	18	\tupletNeutral	52
trasposizione, altezze, “intelligente”	12	TupletNumber	53
trasposizione, chiave	18	\tupletSpan	52, 819
trasposizione, delle altezze	11	tupletSpannerDuration	52
trasposizione, delle note	11	\tupletUp	52
trasposizione, e ottava relativa	5	turca, musica	491
trasposizione, MIDI	27	turco, nome delle note in	491
trasposizione, strumento	27	\turn	129, 793
trasversale ai righe, accordo	350	‘turn’, ornamento	793
trasversale ai righe, gambo	350	\tweak	632, 634, 819
trasversale ai righe, linea	348	\tweak, modificare i punti di controllo	633
trasversale ai righe, nota	350	tweak, relazione con \override	634
trasversale ai righe, parentesi dell’arpeggio	350	two-sided	559
trasversale, nota	344	\type	620
trasversale, travatura	344	\typewriter	723
tratteggio, modelli, legatura di portamento ..	143, 144		
tratti di suddivisione della travatura, direzione ..	97		
trattino	292		
travatura a raggiera	104		
travatura angolare	89		
travatura con angolazione	89		
travatura con angolazione, modifica	90		
travatura convergente o divergente	104		
travatura del tremolo	176		

U

U.C.....	351
ukulele, diagramma dei tasti.....	383
una corda.....	351
\unaCorda.....	351
\underline.....	261, 724
underlining text.....	724
\undertie.....	724
\undo.....	637, 819
unfold.....	171
unfold, finali alternativi.....	171
unfold, ripetizione.....	171
\unfolded.....	819
\unfoldRepeats.....	545, 819
\unHideNotes.....	240
Unicode.....	532
unione delle parti.....	194
unire le note.....	188
universal-color.....	693
\unless.....	767
\unset.....	629
‘up pralltriller’, ornamento.....	793
‘up-bow’, indicazione di arcata.....	794
\upbow.....	129, 356, 794
\upmordent.....	129, 793
‘upper mordent’, ornamento.....	793
\upprall.....	129, 793
\upright.....	725
URL link, as QR code.....	784
UTF-8.....	531

V

varbaritono, chiave di.....	17
\varcoda.....	129, 767, 795
\varheel.....	129, 794
variabile dell’analizzatore sintattico.....	801
variabile globale.....	801
variabile Scheme.....	801
variabili.....	498
variabili, oggetti e proprietà.....	801
variabili, uso delle.....	526
variante del segno di “coda”.....	129
variante, ‘coda’, segno.....	795
\vartoe.....	129, 794
Vaticana, Editio.....	455
VaticanaStaff.....	202, 465
VaticanaVoice.....	465
\vcenter.....	741
\verbatim-file.....	786
\version.....	498
\versus.....	470, 819
VerticalAxisGroup.....	574
verticale, direzione forzata, dei grob.....	643
verticale, spaziatura.....	574, 595
vertically centering text.....	741
\verylongfermata.....	129, 794
\veryshortfermata.....	129, 794
vibraslap.....	796
violino, chiave di.....	17, 709
\virga.....	469, 474
virgolette, nel testo vocale.....	288
\virgula.....	467
visibilità dell’indicazione di tempo.....	71

visibilità dell’oggetto.....	652
visibilità della chiave trasposta.....	658
visibilità della parentesi del gruppo irregolare.....	54
vocale, partitura, aggiungere suggerimenti.....	327
vocale, transizione.....	292
voce.....	184
voce apposita per gestire le interruzioni.....	569
voce, ambitus.....	38
voce, citare.....	222, 226
voce, con ottavazione.....	26
voce, divisa.....	321
voce, \partCombine con \autoBeamOff.....	90
voce, seguire.....	348
voce, spostamento.....	188
voce, stile di.....	187
voce, ulteriore, nella musica polifonica.....	191
voci, multiple.....	188
Voice.....	184
voice.....	29, 31
voice, stile delle alterazioni.....	31
\voiceFour.....	184
\voiceFourStyle.....	187
\voiceNeutralStyle.....	187
\voiceOne.....	184
\voiceOneStyle.....	187
\voices.....	186, 820
\voiceThree.....	184
\voiceThreeStyle.....	187
\voiceTwo.....	184
\voiceTwoStyle.....	187
\void.....	550, 820
volta.....	161
\volta.....	820
volta della ripetizione, modificare.....	169
volta, parentesi.....	169
volta, parentesi accorciate.....	165
volta, prima.....	161
volta, seconda.....	161
volta, stile della doppia ripetizione.....	167
\volta-number.....	725
Volta_engraver.....	166
volume MIDI.....	540
\vshape.....	820
\vspace.....	741
vuoto, accordo.....	124, 344

W

Walker, testa di nota.....	44
\walkerHeads.....	44
\walkerHeadsMinor.....	45
whichBar.....	112
whistle.....	796
whiteout.....	653
\whiteout.....	787
\whiteTriangleMarkup.....	440
\with.....	613, 618
\with-color.....	241, 787
\with-dimension.....	788
\with-dimension-from.....	788
\with-dimensions.....	788
\with-dimensions-from.....	788
\with-link.....	788
\with-outline.....	788

`\with-string-transformer` 725
`\with-true-dimension` 788
`\with-true-dimensions` 789
`\with-url` 755
`\withMusicProperty` 820
`\withRelativeDir` 820
`woodblock` 796
`\woodwind-diagram` 771
`\wordwrap` 267, 742
`\wordwrap-field` 742
`\wordwrap-lines` 273, 792

`\wordwrap-string` 743

X

`x11, colore` 241
`x11-color` 241, 243, 691
`X-offset` 574
`\xNote` 41, 820
`\xNotesOff` 41
`\xNotesOn` 41