

chordx

A package to write song lyrics with chord diagrams in Typst.

v0.4.0

July 9, 2024

<https://github.com/ljgago/typst-chords>

Leonardo Javier Gago

Contents

1. Reference	3
1.1. Chart Chords	3
1.1.1. chart-chord	3
1.2. Piano Chords	6
1.2.1. piano-chord	6
1.3. Single Chords	8
1.3.1. single-chord	8

1. Reference

1.1. Chart Chords

- [chart-chord\(\)](#)

1.1.1. chart-chord

Generates a chart chord for stringed instruments.

Parameters

```
chart-chord(
  ..text-params: auto,
  tabs: str,
  fingers: str,
  capos: str,
  fret: int,
  frets-amount: int,
  design: str,
  position: str,
  background: color,
  name: str content
) -> content
```

..text-params `auto`

Embeds the native **text** parameters from the standard library of **typst**. **Optional**.

tabs `str`

Shows the tabs on the chart. **Optional**.

- **x**: mute note.
- **o**: air note.
- **n**: without note.
- **number**: note position on the fret.

The string length of tabs defines the number of strings on the instrument.

Example:

- "x32o1o" - (6 strings - C Guitar chord).
- "ooo3" - (4 strings - C Ukulele chord).

Default: ""

fingers `str`

Shows the finger numbers. **Optional**.

- **n, x, o**: without finger,
- **number**: one finger

Example: "n32n1n" - (Fingers for guitar chord: C)

Default: ""

capos `str`

Adds one or many capos on the chart. **Optional.**

- 1st digit – **fret**: fret position.
- 2nd digit – **start**: lowest starting string.
- 3rd digit – **end**: highest ending string.

Example: "115" \equiv "1,1,5" \Rightarrow "fret,start,end"

With "|" you can add capos:

Example: "115|312" \equiv "1,1,5|3,1,2" \Rightarrow "fret,start,end|fret,start,end"

Default: ""

fret `int`

Shows the fret number that indicates the starting position of the fretboard. **Optional.**

Default: `none`

frets-amount `int`

Sets the frets amount (the grid rows). **Optional.**

Default: `5`

design `str`

Sets the chart design. **Optional.**

- `"sharp"`: chart with sharp corners.
- `"round"`: chart with round corners.

Default: `"sharp"`

position `str`

Sets the chord chart position. **Optional.**

- `"top"`: chord chart in top position.
- `"bottom"`: chord chart in bottom position.

Default: `"top"`

background `color`

Sets the background color of the chord name. **Optional.**

Default: `rgb(0, 0, 0, 0)`

name `str` or `content`

Shows the chord name. **Required.**

Example:

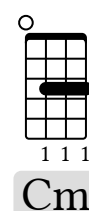
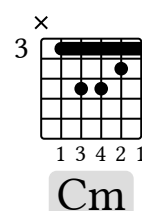
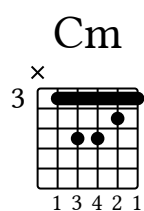
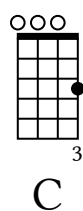
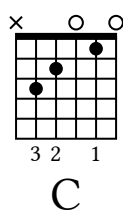
```
#import "@preview/chordx:0.4.0": *

#let chart-chord-sharp = chart-chord.with(size: 18pt)
#let chart-chord-round = chart-chord.with(size: 18pt, design: "round")

// Design "sharp"
#chart-chord-sharp(tabs: "x32o1o", fingers: "n32n1n")[C]
#chart-chord-sharp(tabs: "ooo3", fingers: "ooo3")[C]

// Design "round" with position "bottom"
#chart-chord-round(tabs: "xn332n", fingers: "o13421", fret: 3, capos: "115", position: "bottom")[Cm]
#chart-chord-round(tabs: "onnn", fingers: "n111", capos: "313", position: "bottom")[Cm]

// Design "round" with background color in chord name
#chart-chord-round(tabs: "xn332n", fingers: "o13421", fret: 3, capos: "115", background: silver)[Cm]
#chart-chord-round(tabs: "onnn", fingers: "n111", capos: "313", background: silver)[Cm]
```



1.2. Piano Chords

- [piano-chord\(\)](#)

1.2.1. piano-chord

Generates a piano chord.

Parameters

```
piano-chord(
  ..text-params: auto,
  keys: str,
  layout: str,
  fill: color,
  design: str,
  position: str,
  background: color,
  name: str content
) -> content
```

..text-params `auto`

Embeds the native **text** parameters from the standard library of **typst**. **Optional**.

keys `str`

Keys chord notes from **C1** to **E3** (Depends on your layout). **Optional**.

Example: "**C1**, **E1b**, **G1**" (Cm chord)

Default: ""

layout `str`

Sets the layout and size of the piano, "**C**", "**2C**", "**F**", "**2F**". **Optional**.

- "**C**": the piano layout starts from key **C1** to **E2** (17 keys).
- "**2C**": the piano layout starts from key **C1** to **B2** (24 keys, two octaves).
- "**F**": the piano layout starts from key **F1** to **B2** (19 keys).
- "**2F**": the piano layout stars from key **F1** to **E3** (24 keys, two octaves).

Default: "**C**"

fill `color`

Sets the fill color of the pressed key. **Optional**.

Default: gray

design `str`

Sets the piano design. **Optional.**

- `"sharp"`: piano with sharp corners.
- `"round"`: piano with round corners.

Default: `"sharp"`

position `str`

Sets the chord chart position. **Optional.**

- `"top"`: chord chart in top position.
- `"bottom"`: chord chart in bottom position.

Default: `"top"`

background `color`

Sets the background color of the chord name. **Optional.**

Default: `rgb(0, 0, 0, 0)`

name `str` or `content`

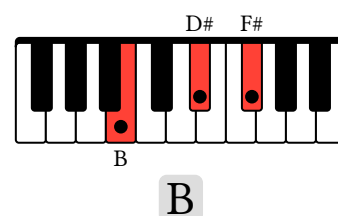
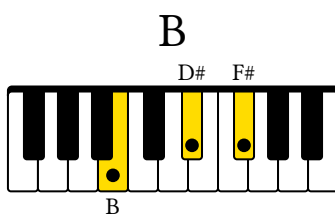
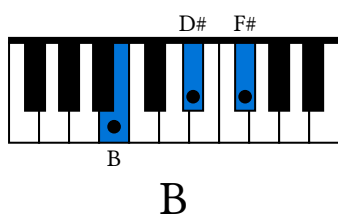
Shows the chord name. **Required.**

Example:

```
#import "@preview/chordx:0.4.0": *

#let piano-chord-sharp = piano-chord.with(layout: "F", size: 18pt)
#let piano-chord-round = piano-chord.with(layout: "F", size: 18pt, design: "round")

#piano-chord-sharp(keys: "B1, D2#, F2#", fill: blue)[B]
#piano-chord-round(keys: "B1, D2#, F2#", fill: yellow, position: "top")[B]
#piano-chord-round(keys: "B1, D2#, F2#", fill: red, background: silver)[B]
```



1.3. Single Chords

- [single-chord\(\)](#)

1.3.1. single-chord

The single chord a chord without diagram used to show the chord name over a word.

Parameters

```
single-chord(
  ..text-params: auto,
  background: color,
  body: content,
  name: content,
  position: content
) -> content
```

..text-params `auto`

Embeds the native **text** parameters from the standard library of **typst**. **Optional**.

background `color`

Sets the background color of the chord. **Optional**.

Default: `rgb(0, 0, 0, 0)`

body `content`

Is the word or words where the chord goes. **Required**.

name `content`

Displays the chord name over the selected words in the body. **Required**.

position `content`

Positions the chord on a specific body character. **Required**.

- `[]`: chord name centered on the body.
- `[number]`: the chord name starts on a specific body character. (First position `[1]`)

Example:

```
#import "@preview/chordx:0.4.0": *

#let chord = single-chord.with(
  font: "PT Sans",
  size: 10pt,
  weight: "semibold",
  background: silver
)

#chord[Jingle][G][2] bells, jingle bells, jingle #chord[all][C][2] the #chord[way!][G][2] \
```



```
#chord[Oh][C][] what fun it #chord[is][G][] to ride \  
In a #chord[one-horse][A7][2] open #chord[sleigh,][D7][3] hey!
```

G **C** **G**
Jingle bells, jingle bells, jingle all the way!

C **G**
Oh what fun it is to ride

A7 **D7**
In a one-horse open sleigh, hey!