

# chordx

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

v0.2.0

August 25, 2023

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

**Leonardo Javier Gago**

## Contents

1. Reference .....	3
1.1. Chart Chords .....	3
1.1.1. new-chart-chords .....	3
1.1.2. chart-chord .....	3
1.2. Piano Chords .....	5
1.2.1. new-piano-chords .....	5
1.2.2. piano-chord .....	6
1.3. Single Chords .....	8
1.3.1. new-single-chords .....	8
1.3.2. single-chord .....	8

# 1. Reference

## 1.1. Chart Chords

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

### 1.1.1. new-chart-chords

Return a new function with default parameters to generate chart chords for stringed instruments.

#### Parameters

```
new-chart-chords(
  frets: integer,
  scale: integer float,
  style: string,
  font: string
) -> function
```

**frets**    integer

Presets the number of frets (rows of the grid). **Optional**.

Default: 5

**scale**    integer or float

Presets the scale. **Optional**.

Default: 1

**style**    string

Sets the chart style. **Optional**.

- "normal": chart with right angles.
- "round": chart with round angles.

Default: "normal"

**font**    string

Sets the name of the text font. **Optional**.

Default: "Linux Libertine"

### 1.1.2. chart-chord

Is the returned function by **new-chart-chords**.

## Parameters

```
chart-chord(
  tabs: string,
  fingers: string,
  capos: string,
  frets: integer,
  fret-number: integer,
  scale: integer,
  name: string content
) -> content
```

**tabs** string

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** string

Shows the finger numbers. **Optional**.

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

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

Default: ""

**capos** string

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

- 1<sup>st</sup> digit – **fret**: fret position.
- 2<sup>nd</sup> digit – **start**: lowest starting string.
- 3<sup>rd</sup> 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: ""

**frets** `integer`Sets the number of frets (rows of the chart grid). **Optional**.

Default: frets

**fret-number** `integer`Shows the fret number that indicates the starting position of the fretboard. **Optional**.Default: `none`**scale** `integer`Sets the scale. **Optional**.

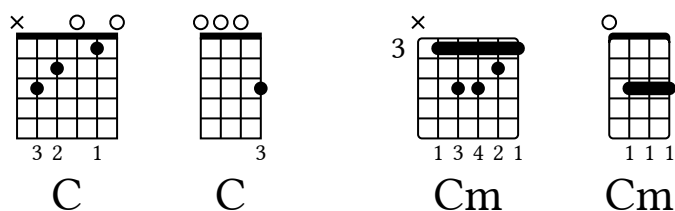
Default: scale

**name** `string` or `content`Shows the chord name. **Required**.**Example:**

```
#import "@preview/chordx:0.2.0": *

#let chart-chord = new-chart-chords(scale: 1.5)
#let chart-chord-round = new-chart-chords(style: "round", scale: 1.5)

#chart-chord(tabs: "x32o1o", fingers: "n32n1n")[C]
#h(2em)
#chart-chord(tabs: "ooo3", fingers: "ooo3")[C]
#h(4em)
#chart-chord-round(tabs: "xn332n", fingers: "o13421", fret-number: 3, capos: "115")[Cm]
#h(2em)
#chart-chord-round(tabs: "onnn", fingers: "n111", capos: "313")[Cm]
```



## 1.2. Piano Chords

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

### 1.2.1. new-piano-chords

Return a new function with default parameters to generate piano chords.

## Parameters

```
new-piano-chords(  
  layout: string,  
  scale: integer | float,  
  color: color,  
  style: string,  
  font: string  
) -> function
```

### layout string

Presets 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 starts from key F1 to E3 (24 keys, two octaves).

Default: "C"

### scale integer or float

Presets the scale. **Optional.**

Default: 1

### color color

Presets the pressed key color. **Optional.**

Default: gray

### style string

Sets the piano style. **Optional.**

- "normal": piano with right angles.
- "round": piano with round angles.

Default: "normal"

### font string

Sets the name of the text font. **Optional.**

Default: "Linux Libertine"

## 1.2.2. piano-chord

Is the returned function by **new-piano-chords**.

## Parameters

```
piano-chord(
  keys: string,
  color: color,
  layout: string,
  scale: integer float,
  name: string content
) -> content
```

**keys**    string

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

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

Default: ""

**color**    color

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

Default: color

**layout**    string

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: layout

**scale**    integer or float

Sets the scale. **Optional**.

Default: scale

**name**    string or content

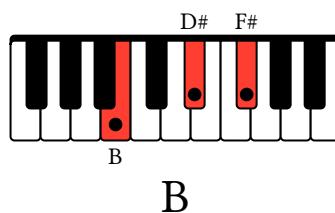
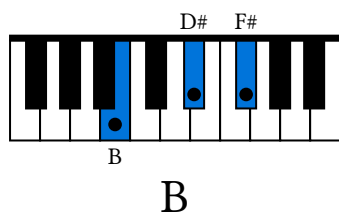
Shows the chord name. **Required**.

## Example:

```
#import "@preview/chordx:0.2.0": *

#let piano-chord = new-piano-chords(layout: "F", scale: 1.5)
#let piano-chord-round = new-piano-chords(layout: "F", scale: 1.5, style: "round")

#piano-chord(keys: "B1, D2#, F2#", color: blue)[B]
#h(4em)
#piano-chord-round(keys: "B1, D2#, F2#", color: red)[B]
```



### 1.3. Single Chords

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

#### 1.3.1. new-single-chords

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

##### Parameters

```
new-single-chords(..text-params: auto) -> function
```

**..text-params** `auto`

Are the same parameters of **text** from the standard library of **typst**. **Required.**

#### 1.3.2. single-chord

Is the returned function by **new-single-chords**.

##### Parameters

```
single-chord(
  body: content,
  name: content,
  position: content
)
```

**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 over a specific character in the body. **Required.**

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

##### Example:



```
#import "@preview/chordx:0.2.0": *  
  
#let chord = new-single-chords(style: "italic", weight: "semibold")  
  
#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!