

keyle

v0.1.0

2024-08-09

MIT

This package provides a simple way to style keyboard shortcuts in your documentation.

magicwenli <YXNIAN@OUTLOOK.COM>

<https://github.com/magicwenli/keyle>

Table of contents

I. About

II. Usage

II.1. Importing 3

II.2. Quick Start 3

III. Themes

III.1. Built-in Themes 4

III.2. Custom Themes 4

IV. Symbols

IV.1. Mac Keyboard Symbols 5

IV.2. Linux Biolinum Keyboard Symbols 5

V. Available commands

VI. Index

Part I.

About

KEYLE is a library that allows you to create HTML `<kbd>` like keyboard shortcuts simple and easy.

The name, `keyle`, is a combination of `key` and `theme`.

This project is inspired by [auth0/kbd](#) and [dogezen/badgery](#). Send them respect!

Part II.

Usage

II.1. Importing

KEYLE is imported using

```
#import "@preview/keyle:0.1.0"
```

II.2. Quick Start

Generate a keyboard renderer with `#config()` function. Section Section III.1 lists available themes.

```
#let kbd = keyle.config()
#kbd("Ctrl","Shift","Alt","Del")

#let kbd = keyle.config(theme: keyle.themes.biolinum, delim: keyle.biolinum-key.delim_plus)
#kbd("Ctrl","Shift","Alt","Del")
```

Ctrl + Shift + Alt + Del

Ctrl + ⇧ + Alt + Del

There are `compact` and `delim` options to make the output more compact and change the delimiter between keys.

You can either use them in `#config()` function or directly in generated `kbd` function.

```
#let kbd = keyle.config(compact: true, delim: "-")
#kbd("Ctrl","Shift","Alt","Del")

#let kbd = keyle.config()
#kbd("Ctrl","Shift","Alt","Del", compact: true, delim: "-")
```

Ctrl-Shift-Alt-Del

Ctrl-Shift-Alt-Del

Part III.

Themes

III.1. Built-in Themes

Themes function are available at `#keyle.themes` dictionary.

`#keyle.themes.standard`

Home

`#keyle.themes.deep-blue`

HOME

`#keyle.themes.type-writer`

HOME

`#keyle.themes.biolinum`

Home

III.2. Custom Themes

You can create your own theme by defining a function that takes a string and returns a themed content.

```
// https://www.radix-ui.com/themes/playground#kbd
#let radix_kdb(content) = box(
  rect(
    inset: (x: 0.5em),
    outset: (y: 0.05em),
    stroke: rgb("#1c2024") + 0.3pt,
    radius: 0.35em,
    fill: rgb("#fcfcfd"),
    text(fill: black, font: ("Segoe UI (Custom)", "Roboto", "Helvetica Neue", "Open Sans (Custom)", "sans-serif", "Apple
Color Emoji", "Segoe UI Emoji"), content),
  ),
)
#let kbd = keyle.config(theme: radix_kdb)
#kbd("⌘ D") #kbd("^↑ F")
```

⌘ D

^↑ F

Part IV.

Symbols

IV.1. Mac Keyboard Symbols

KEYLE provides symbols for Mac keyboard. You may need install **Fira Code** font to see the symbols correctly.

- [Fira Code @ Google Fonts](#)
- [Apple Mac Keyboard Symbols](#)

```
#let kbd = keyle.config(theme: keyle.themes.standard)
#let mac-key-font = (
  "Fira Code",
  "FiraCode",
  "FiraCode Nerd Font Mono",
)
#grid(columns: (2fr, 1fr, 2fr, 1fr), rows: 2em, align: horizon, .keyle.mac-key.pairs().map(item => (
  raw("#keyle.mac-key." + item.at(0)),
  kbd(text(font: mac-key-font, item.at(1))),
)).flatten())
```

#keyle.mac-key.command



#keyle.mac-key.shift



#keyle.mac-key.option



#keyle.mac-key.control



#keyle.mac-key.return



#keyle.mac-key.delete



#keyle.mac-key.forward-delete



#keyle.mac-key.escape



#keyle.mac-key.left



#keyle.mac-key.right



#keyle.mac-key.up



#keyle.mac-key.down



#keyle.mac-key.pageup



#keyle.mac-key.pagedown



#keyle.mac-key.home



#keyle.mac-key.end



#keyle.mac-key.tab-right



#keyle.mac-key.tab-left



































IV.2. Linux Biolinum Keyboard Symbols

KEYLE provides symbols for Linux Biolinum Keyboard font.

- [Linux Biolinum Keyboard](#)

4.2 Linux Biolinum Keyboard Symbols

```
#let kbd = keyle.config(theme: keyle.themes.biolinum)
#grid(columns: (5fr, 3fr, 5fr, 3fr), rows: 2em, align: horizon, ..keyle.biolinum-key.pairs().map(item => (
  raw("#keyle.biolinum-key:" + item.at(0)),
  kbd(item.at(1)),
)).flatten())
```

#keyle.biolinum-key.Strg		#keyle.biolinum-key.Alt	
#keyle.biolinum-key.Ctrl		#keyle.biolinum-key.Shift	
#keyle.biolinum-key.Tab		#keyle.biolinum-key.Enter	
#keyle.biolinum-key.Capslock		#keyle.biolinum-key.Home	
#keyle.biolinum-key.Del		#keyle.biolinum-key.Ins	
#keyle.biolinum-key.End		#keyle.biolinum-key.Space	
#keyle.biolinum-key.Esc		#keyle.biolinum-key.PageUp	
#keyle.biolinum-key.PageDown		#keyle.biolinum-key.Back	
#keyle.biolinum-key.Pad_0		#keyle.biolinum-key.Pad_1	
#keyle.biolinum-key.Pad_2		#keyle.biolinum-key.Pad_3	
#keyle.biolinum-key.Pad_4		#keyle.biolinum-key.Pad_5	
#keyle.biolinum-key.Pad_6		#keyle.biolinum-key.Pad_7	
#keyle.biolinum-key.Pad_8		#keyle.biolinum-key.Pad_9	
#keyle.biolinum-key.Pad_Div		#keyle.biolinum-key.Pad_Add	
#keyle.biolinum-key.Pad_Sub		#keyle.biolinum-key.Pad_Mul	
#keyle.biolinum-key.Pad_Enter		#keyle.biolinum-key.delim_plus	+
#keyle.biolinum-key.delim_minus	-	#keyle.biolinum-key.Win	
#keyle.biolinum-key.GNU		#keyle.biolinum-key.Tux	
#keyle.biolinum-key.Mac_Cmd		#keyle.biolinum-key.Mac_Opt	
#keyle.biolinum-key.Mac_FDel		#keyle.biolinum-key.Mac_Del	

Part V.

Available commands

<code>#config()</code>	<code>#theme-func-deep-blue()</code>	<code>#theme-func-type-writer()</code>
<code>#theme-func-biolinum()</code>	<code>#theme-func-standard()</code>	

#theme-func-standard(`<sym>`) → `content`

Theme function to render keys in a standard style.

— Argument —

`<sym>`

`str`

The key symbol to render.

#theme-func-deep-blue(`<sym>`) → `content`

Theme function to render keys in a deep blue style.

— Argument —

`<sym>`

`str`

The key symbol to render.

#theme-func-type-writer(`<sym>`) → `content`

Theme function to render keys in a type writer style.

— Argument —

`<sym>`

`str`

The key symbol to render.

#theme-func-biolinum(`<sym>`) → `content`

Theme function to render keys in a Linux Biolinum Keyboard style.

You need to have the font installed on your system.

— Argument —

`<sym>`

`str`

The key symbol to render.

#config(`<theme>`: `"themes.standard"`, `<compact>`: `false`, `<delim>`: `"+"`) → `function`

Config function to generate keyboard rendering helper function.

— Argument —

`<theme>`: `"themes.standard"`

`function`

The theme function to use.

— Argument —

`<compact>`: `false`

`bool`

Whether to render keys in a compact format.

Argument

`<delim>: "+"`

`str`

The delimiter to use when rendering keys in compact format.

Part VI.

Index

C

[#config](#) 7

T

[#theme-func-biolinum](#) 7

[#theme-func-deep-blue](#) 7

[#theme-func-stardard](#) 7

[#theme-func-type-writer](#) 7