

Cuti Demo & 中文文档

简介

Cuti 是一个为了方便用户使用伪粗体和伪斜体而设计的包。

光速上手

在文档顶端加入

```
1 #import "@preview/cuti:0.2.1": show-cn-fakebold
2 #show: show-cn-fakebold
```

宋体、黑体、楷体的加粗就工作了。

请注意: 如果您使用 Source Han Sans / Source Han Serif 等包含 bold 字重的中文字体, 不建议您对整个文档使用 `show: show-cn-fakebold` ——对整个文档使用 `show: show-cn-fakebold` 将导致支持中文粗体的字体也会使用“伪粗体”版的粗体。如下所示:

```
1 #set text(font: "Source Han Serif SC")
2 + Regular: 春江潮水连海平, 海上明月共潮生。
3 + Bold: #text(weight: "bold")[春江潮水连海平, 海上明月共潮生。]
4 #show: show-cn-fakebold
5 + Fakebold: #text(weight: "bold")[春江潮水连海平, 海上明月共潮生。]
```

1. Regular: 春江潮水连海平, 海上明月共潮生。
2. Bold: 春江潮水连海平, 海上明月共潮生。
3. Fakebold: 春江潮水连海平, 海上明月共潮生。

第 2 行中, 使用的是 Source Han Serif SC 的 Bold 字重。但设置 `show: show-cn-fakebold` 后, 第 3 行使用的是伪粗体描边生成 Bold 字重。

Demo

Part 1: font: ("Times New Roman", "SimSun")

Regular: 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Bold(Font Only): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Bold(Fake Only): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Bold(Fake+Font): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Italic(Font Only): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Italic(Fake Only): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。
Italic(Fake+Font): 你说得对, 但是《Cuti》是一个用于伪粗体和伪斜体的包。

Part 2: font: "Source Han Serif SC"

Regular: 前面忘了。同时, 逐步发掘「Typst」的奥妙。
Bold(Font Only): 前面忘了。同时, 逐步发掘「Typst」的奥妙。
Bold(Fake Only): 前面忘了。同时, 逐步发掘「Typst」的奥妙。
Bold(Fake+Font): 前面忘了。同时, 逐步发掘「Typst」的奥妙。

Fakebold 部分

Cuti 利用 text 的 stroke 属性生成伪粗体。该工具通常可用于为宋体、黑体、楷体等字体提供“粗体”。Cuti 使用 0.02857em 作为 stroke 的参数。在 Microsoft Office 中，使用伪粗体会给字符添加一个 0.02857em 的描边。（实际上，精确值可能是 $\frac{1}{35}$ 。）

fakebold

不带其他参数的 #fakebold[] 会为字符添加伪粗体效果。

```
1 - Fakebold: #fakebold[#lorem(5)]
2 - Bold: #text(weight: "bold", lorem(5))
3 - Bold + Fakebold: #fakebold[#text(weight: "bold", lorem(5))]
```

- Fakebold: **Lorem ipsum dolor sit amet.**
- Bold: **Lorem ipsum dolor sit amet.**
- Bold + Fakebold: **Lorem ipsum dolor sit amet.**

#fakebold[] 可以接受与 #text 相同的参数。特别地，若指定 weight 参数，可以用于指定基于某种字重进行描边。如果不指定 weight，基准字重会从上文继承。指定 stroke 参数会被忽略。

```
1 - Bold + Fakebold: #fakebold(weight: "bold")[#lorem(5)]
2 - Bold + Fakebold: #set text(weight: "bold"); #fakebold[#lorem(5)]
```

- Bold + Fakebold: **Lorem ipsum dolor sit amet.**
- Bold + Fakebold: **Lorem ipsum dolor sit amet.**

注: cuti:0.2.0 使用的 base-weight 参数仍保留以保证兼容性。

如果将文字设置为彩色，伪粗体描边也将变成相应的颜色。

```
1 - Blue + Fakebold: #fakebold(fill: blue)[花生瓜子八宝粥，啤酒饮料矿泉水。#lorem(5)]
2 - Gradient + Fakebold: #set text(fill: gradient.conic(..color.map.rainbow));
  #fakebold[花生瓜子八宝粥，啤酒饮料矿泉水。#lorem(5)]
```

- Blue + Fakebold: 花生瓜子八宝粥，啤酒饮料矿泉水。Lorem ipsum dolor sit amet.
- Gradient + Fakebold: 花生瓜子八宝粥，啤酒饮料矿泉水。Lorem ipsum dolor sit amet.

regex-fakebold

#regex-fakebold 设计上是用于多语言、多字体情境的，可以根据参数 reg-exp 内的正则表达式只将匹配到的字符应用伪粗体格式。它也可以接受 #font 相同的参数。

```
1 + RegExp `[a-o]`: #regex-fakebold(reg-exp: "[a-o]")[#lorem(5)]
2 + RegExp `\\p{script=Han}`: #regex-fakebold(reg-exp: "\\p{script=Han}")[衬衫的价格是 9 磅 15 便士。]
3 + RegExp `\\p{script=Han}`: #set text(weight: "bold"); #regex-fakebold(reg-exp:
  "\\p{script=Han}")[衬衫的价格是 9 磅 15 便士。]
```

1. RegExp [a-o]: Lorem ipsum dolor sit amet.
2. RegExp \\p{script=Han}: 衬衫的价格是 9 磅 15 便士。
3. RegExp \\p{script=Han}: 衬衫的价格是 9 磅 15 便士。

在上面的例子 #3 中, 9 和 15 是字体提供的“真”粗体, 而其他字符是用 regular 字重描边得到的伪粗体。

show-fakebold

在多语言、多字体的场景中, 不同的语言通常使用不同的字体, 但是不是所有的字体都自带 bold 字重。需要 strong 或者 bold 效果时, 每次都使用 #fakebold #regex-fakebold 并不方便。所以, 我们提供了用于设置 show 规则 #show-fakebold 函数。

show-fakebold 和 regex-fakebold 有着相同的参数。默认情况下 show-fakebold 使用 "." 作为正则表达式, 也就是说, 所有字符带加粗或 strong 属性都会被伪粗体加粗。

```
1 #show: show-fakebold
2 - Regular: #lorem(10)
3 - Bold: #text(weight: "bold")[#lorem(10)]
```

- Regular: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.
- Bold: **Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.**

正常情况下字体提供的加粗与伪粗体叠加的效果不是我们想要的。一般需要指定正则表达式、指定伪粗体的生效范围。

```
1 #show: show-fakebold.with(reg-exp: "\p{script=Han}")
2 - Regular: 我正在使用 Typst 排版。
3 - Strong: *我正在使用 Typst 排版。*
```

- Regular: 我正在使用 Typst 排版。
- Strong: **我正在使用 Typst 排版。**

show-fakebold 也接受 #font 相同的参数。

cn-fakebold & show-cn-fakebold

这两个都是为中文排版封装的。

cn-fakebold 是 regex-fakebold 的封装。cn-fakebold 会将中文和常见符号进行伪粗体处理, 基准字重为 regular 字重。请注意, 在混排中英文时, 需要另行指定 weight: "bold"。

```
1 #set text(font: ("Linux Libertine", "SimSun"))
2 - Regular: 有时, 我们点击链接, 打开的却是《Never Gonna Give You Up》这首歌。
3 - `cn-fakebold`: #cn-fakebold[《Never Gonna Give You Up》是英国歌手 Rick Astley 演唱的
  歌曲, 于 1987 年发行。]
4 #set text(weight: "bold")
5 - Bold: 《Never Gonna Give You Up》已经成为了一种网络迷因。
6 - Bold + `cn-fakebold`: #cn-fakebold[在 2024 年的今天,《Never Gonna Give You Up》仍有
  独特的魅力。]
```

- Regular: 有时, 我们点击链接, 打开的却是《Never Gonna Give You Up》这首歌。
- cn-fakebold: 《Never Gonna Give You Up》是英国歌手 Rick Astley 演唱的歌曲, 于 1987 年发行。
- Bold: **《Never Gonna Give You Up》** 已经成为了一种网络迷因。
- Bold + cn-fakebold: **在 2024 年的今天, 《Never Gonna Give You Up》** 仍有独特的魅力。

show-cn-fakebold 是 show-fakebold 的封装, 默认的正则范围是中文字符与常见标点符号。

```
1 #show: show-cn-fakebold
2 - Regular: 滚滚长江东逝水，浪花淘尽英雄。 #lorem(5)
3 - Bold: #text(weight: "bold")[滚滚长江东逝水，浪花淘尽英雄。 #lorem(5)]
4 - Strong: *滚滚长江东逝水，浪花淘尽英雄。 #lorem(5)*
```

- Regular: 滚滚长江东逝水，浪花淘尽英雄。 Lorem ipsum dolor sit amet.
- Bold: 滚滚长江东逝水，浪花淘尽英雄。 Lorem ipsum dolor sit amet.
- Strong: 滚滚长江东逝水，浪花淘尽英雄。 Lorem ipsum dolor sit amet.

也可以这样使用：

```
1 #show-cn-fakebold[
2   - Regular: 滚滚长江东逝水，浪花淘尽英雄。 #lorem(5)
3   - Strong: *滚滚长江东逝水，浪花淘尽英雄。 #lorem(5)*
4 ]
```

- Regular: 滚滚长江东逝水，浪花淘尽英雄。 Lorem ipsum dolor sit amet.
- Strong: 滚滚长江东逝水，浪花淘尽英雄。 Lorem ipsum dolor sit amet.

这两个函数也可以接受 #font 相同的参数，以指定中文字符加粗的效果。

Fakeitalic 部分

Cuti 使用的 skew 函数来自 typst issue #2749 (<https://github.com/typst/typst/issues/2749>) by Enivex.

skew 利用 rotate 和 scale 的组合生成伪斜体。Cuti 使用 -0.32175 作为默认的倾斜角度。在 Microsoft Office 中，使用伪粗体会给字符添加一个 $\arctan(\frac{1}{3})$ 的倾斜效果。请注意，由于不同的英文字体拥有不同的倾斜角度，需要自行寻找一个合适的角度。如果使用 Times New Roman 与中易宋体，则默认的角度是比较合适的。

fakeitalic

```
fakeitalic(  
  ang: angle 默认值 -0.32175,  
  content  
)
```

#fakeitalic[] 会为字符添加伪斜体效果。

```
1 - Regular: #lorem(5)  
2 - Italic: #text(style: "italic", lorem(5))  
3 - Fakeitalic: #fakeitalic[#lorem(5)]  
4 - Fakeitalic + Fakebold: #fakeitalic[#fakebold[#lorem(5)]]
```

- Regular: Lorem ipsum dolor sit amet.
- Italic: *Lorem ipsum dolor sit amet.*
- Fakeitalic: *Lorem ipsum dolor sit amet.*
- Fakeitalic + Fakebold: ***Lorem ipsum dolor sit amet.***

可以通过调整 ang 参数调整倾斜的角度。

```
1 - -10deg: #fakeitalic(ang: -10deg)[#lorem(5)]  
2 - -20deg: #fakeitalic(ang: -20deg)[#lorem(5)]  
3 - +20deg: #fakeitalic(ang: 20deg)[#lorem(5)]
```

- -10deg: *Lorem ipsum dolor sit amet.*
- -20deg: *Lorem ipsum dolor sit amet.*
- +20deg: *Lorem ipsum dolor sit amet.*

regex-fakeitalic

```
regex-fakeitalic(  
  reg-exp: str 默认值: "[^ ]",  
  ang: angle ,  
  spacing: relative none 默认值: none ,  
  content  
)
```

#regex-fakeitalic 设计上是用于多语言、多字体情境的，可以根据参数 reg-exp 内的正则表达式只将匹配到的字符应用伪斜体格式。它也可以接受 ang 参数。

```
1 + RegExp `[a-o]`: #regex-fakeitalic(reg-exp: "[a-o]")[#lorem(5)]
```

```
2 + RegExp `\\p{script=Han}`: #regex-fakeitalic(reg-exp: "\\p{script=Han}")[衬衫的价格是  
9 磅 15 便士。]  
3 + RegExp `\\p{script=Han}`: #set text(style: "italic"); #regex-fakeitalic(reg-exp:  
"\\p{script=Han}", ang: -10deg)[衬衫的价格是 9 磅 15 便士。]
```

1. RegExp [a-o]: *Lorem ipsum dolor sit amet.*
2. RegExp \\p{script=Han}: 衬衫的价格是 9 磅 15 便士。
3. RegExp \\p{script=Han}: 衬衫的价格是 9 磅 15 便士。

在上面的例子 #3 中, 9 和 15 是字体提供的“真”斜体, 而其他字符是用 skew 变换得到的伪斜体。

为什么没有 **show-fakeitalic** 与 **show-cn-fakeitalic**

因为斜体远比我想象中要复杂。斜体与直体、伪斜体与真斜体之间需要留间隔, 而我没有找到合适的间隔; 不同字体要写不同的角度, 很难即开即用。

并且, 在 Demo 部分也可以发现: 伪粗体的符号间距会出现问题。