

CTyp 使用手册

版本: 0.1.0

CTyp 是一个用于提供 Typst 中文排版支持的包。该包集成了一些常用的中文排版设置，用于提供快速的中文排版体验。

⚠ 警告

由于 Typst 的中文排版支持仍不完善，该包只能提供非语言级的排版支持。并不保证能够实现所有中文排版需求。

1. 快速开始

通过以下代码快速使用 CTyp 包的设置：

```
1 #import "@local/ctyp:0.1.0": ctyp
2 #let (ctypset, cjk) = ctyp()
3 #show: ctypset
```

📌 注意

变量名 ctypset 和 cjk 可以自行设置，无需使用文档中的名字。

2. 字体设置

2.1. 字体集合

字体集合是一系列字体名称的集合，包括其变体。字体集合通常包含以下元素：宋体、黑体、楷体、仿宋。字体集合至少包含一个元素，但无需包含所有元素。目前 CTyp 包提供 Fandol 字体集合。

字体集合应当是如下结构的字典，

```
1 #let fandol-fontset = (
2   family: (
3     song: (name: "FandolSong", variants: ("bold",)),
4     hei: (name: "FandolHei", variants: ("bold",)),
5     ... // 其他字体
6   ),
7   map: (
8     text: (cjk: "song", latin: "serif"),
9     strong: (cjk: "hei", latin: "serif"),
10    ... // 其他元素
11  )
12 )
```

① 格式说明

- **family**: 一个字典，列出了所用的各种字形，如 `song`, `hei`, `kai`, `fang` 等。
 - ▶ **[key]**: 字形的名称，是一个字典。
 - **name**: 所用的字体的名称。
 - **variants**: 这个字体包含的所有变体。
- **map**: 一个字典，列出各种元素所使用的 CJK 字体和西文字体对应关系。
 - ▶ **[element]**: 元素的名称
 - **cjk**: CJK 字体映射, 格式为 `"family": "variant"`。其中 **family** 是 **family** 中的键, **variant** 是 **variants** 中的值。若指定了不再存在的变体, 认为该字体不存在, 回归使用 **family** 字典中的第一个字体。
 - **latin**: 西文字体映射, 可选项是 `"serif"`, `"sans"`, `"mono"` 或者字体名称。
 - **serif**: 使用衬线字体。
 - **sans**: 使用无衬线字体。
 - **mono**: 使用等宽字体。

2.2. 修改字体映射

在字体集合提供默认字体映射的基础上，可以为特定元素修改其所使用的字体。设置方法是通过 `ctyp()` 函数中的参数 `font-cjk-map()` 来修改，格式参考字体集合字典中的 `map` 字段。需要保证修改后的字体映射仍然能够找到对应的字体。

除了能够设置元素的中文字体，还可以设置对应的西文字体，使得中西文字体能够匹配。

💡 修改字体映射

将 `strong` 元素从使用**黑体**改为使用宋体

```
1 #let (ctypset, cjk) = ctyp(t Typst
2   font-cjk-map: (
3     strong: (cjk: "song:bold", latin: "sans"),
4   ),
5   font-latin: (
6     sans: "Arial"
7   )
8 )
9 #show: _ctypset
10 #strong[加粗的宋体内容]
```

加粗的宋体内容，西文使用 **Sans** 字体

从上面的例子可以看到，默认情况下 `strong` 元素使用**黑体**，而不加粗。也就是和 `#hei(weight: "regular")` 的效果相同。如果使用**加粗的黑体**，则可能过粗。这通常也是 LaTeX 中的默认行为。如果偏好 Microsoft Word 的行为，即使用**黑体**的情况下加粗，那么可以在 `font-cjk-map` 中将对应元素的 `cjk` 字段增加一个 `:bold` 后缀。

注意

为了实现这一效果，`strong` 元素的 `delta` 被设置为 0，并全局生效。如果这导致了异常的行为，可以在 `ctyp()` 函数中设置 `reset-strong-delta` 参数为 300，则恢复默认行为。该值是 Typst 文档中标注的默认值。如果希望使用其他值，可以自行设置。

2.3. 使用 CJK 字体

函数 `ctyp()` 的返回值中的第二个元素 `cjk` 是一个字典。字典的键都来自于字体集合中 `family` 字段的键，也就是字形的名称；值是一个函数，直接使用可以修改内容的字体。

💡 直接使用 CJK 字体

```
1 #let (ctypset, cjk) = ctyp()
2 #let (song, hei, kai, fang) = cjk
3 - #song[这是宋体内容]
4 - #hei[这是黑体内容]
5 - #kai[这是楷体内容]
6 - #fang[这是仿宋内容]
```

 Typst

- 这是宋体内容
- 这是黑体内容
- 这是楷体内容
- 这是仿宋内容

3. 列表

不论是编号列表还是符号列表，在使用中文时，很容易产生列表项目符号与内容基线不平的问题。该包重新设置了列表的样式，使得列表项目符号与内容基线对齐。该功能默认开启。


📄 修复后的列表

- 1) 项目 1
- 2) 项目 2
 - 项目 2.1
 - 项目 2.2
 - 项目 2.3
 - ▶ 项目 2.3.1
 - ▶ 项目 2.3.2
 - ▶ 项目 2.3.3
 - 1) 项目 2.3.3.1
 - 2) 项目 2.3.3.2
 - a) 项目 2.3.3.2.1

如果不喜欢该功能，可以将 `ctyp()` 函数的参数 `fix-list-enum` 设置为 `false`，则不修复列表样式。

💡 不修复列表

```
1 #let (theme, _) = ctyp(  
2   fix-list-enum: false,  
3 )  
4 #show: theme
```


 Typst

4. 页面设置

通常中文环境下，对页面的设置是基于字符数的。该包提供了一个 `page-grid()` 函数，可以根据字符数设置页面的边距。该函数接收 `page()` 函数的 `margin` 参数的所有合法值，但是对于 `width` 和 `height` 参数，必须是整数，表示字符的数量。

💡 页面设置

```
1 #import "@local/ctyp:0.1.0": page-grid  
2 #show: page-grid.with(  
3   width: 45,  
4   height: 70  
5 )
```

 Typst

📌 注意

由于 Typst 的限制，页面设置不能放在 `ctyp()` 函数中。目前采用的是提供单独的 `page-grid()` 函数来设置页面。这也有一些额外的好处，例如可以与其他包结合使用。