CTyp 使用手册

版本: 0.2.0

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。

目录

1. 快速开始

通过以下代码快速使用 CTyp 包的设置：

```typst
#import "@preview/ctyp:0.2.0": ctyp
#let (ctypset, cjk) = ctyp()
#show: ctypset
```

> **ℹ 注意**
>
> 变量名 `ctypset` 和 `cjk` 可以自行设置，无需使用文档中的名字。

在 Typst Web App 环境中，可以直接使用 Noto CJK 字体系列，通过以下代码进行设置：

```typst
#import "@preview/ctyp:0.2.0": *
#let (ctypset, cjk) = ctyp(
  fontset-cjk: "noto"
)
#show: ctypset
```

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。

fontset
字体集合。接受一个字符串选取预定义的字体集合，或一个字典自定义字体集合。
默认值：`fandol`

2.1.1. 格式说明

字体集合应当是如下结构的字典。

```typst
#let fandol-fontset = (
  family: (
    song: (name: "FandolSong", variants: ("bold",)),
    hei: (name: "FandolHei", variants: ("bold",)),
    ... // 其他字体
  ),
  map: (
    text: (cjk: "song", latin: "serif"),
    strong: (cjk: "hei", latin: "serif"),
    .. // 其他元素
  )
)
```

具体字段含义如下：

- `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。

2.1.2. 预定义的集合

CTyp 包提供了以下预定义的字体集合：`fandol`, `fangzheng`, `source`, `noto`, `windows`, `huawen`。安装对应字体后即可使用。字体集合的详细情况参考附录 B。

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。

font-cjk-map
字体映射表。修改字符集合定义的映射表。
默认值：`(:)`

除了能够设置元素的中文字体，还可以通过 `font-latin` 参数设置西文字体映射。西文字体的字形为 `fontset.map.[element].latin` 中所用到的值，一般情况下是 `serif`, `sans`, `mono` 三种。该参数需要为这些字形指定具体的字体。

font-latin
西文字体映射。定义每种西文字形所对应的字体。
默认值：(:)

修改字体映射

将 `strong` 元素从使用**黑体**改为使用宋体

```typst
#let (ctypset, cjk) = ctyp(
  font-cjk-map: (
    strong: (cjk: "song:bold", latin: "sans"),
  ),
  font-latin: (
    sans: "Arial"
  )
)
#show: _ctypset
#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` 元素使用常规黑体**，而非**加粗的黑体**，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。这些函数提供一个参数 `weight` 用于设置字体粗细。

直接使用 CJK 字体

```typst
#let (ctypset, cjk) = ctyp()
#let (song, hei, kai, fang) = cjk
- #song[这是宋体内容]
- #hei(weight: "bold")[这是黑体加粗内容]
- #kai[这是楷体内容]
- #fang[这是仿宋内容]
```

- 这是宋体内容

- **这是黑体加粗内容**
- 这是楷体内容
- 这是仿宋内容

这些函数中的西文字体默认使用 `ctyp()` 函数参数 `font-latin` 中的 `serif` 字体。若要修改西文字体，可以通过这些修改这些函数中的 `latin` 参数来实现，该参数接受 `font-latin` 中的键，或具体的字体名称。

使用 CJK 字体并设置西文字体

```
#fang(latin: "mono")[使用仿宋体和 monospace 西文字体。]
```

Typst

使用仿宋体和 monospace 西文字体。

2.4. 智能引号（Smartquote）

fix-smartquote
设置是否开启智能引号修复。
默认值：`true`

当中文与英文混杂时，如果通过 `set text(lang: "zh")` 设置了，Typst 在英文中也会使用中文的智能引号。为了修复这一问题，该包参考了“Typst 中文社区”中提供的解决方案，将智能引号的字体设置为 Latin 字体。考虑到中文书写时，往往不会使用智能引号，而是依赖输入法进行输入；而使用英文时，却往往需要智能引号，尤其是在纯文本编辑器中。因此，该包默认开启了智能引号的字体设置。

警告

该功能启用时，无论是中文还是英文，智能引号都会使用西文字体。反之，手动引号（使用 `sym.quote.l` 和 `sym.quote.r` 或与之相应的字符）都会使用中文字体。也就是说：

- 中文应当使用“手动引号”，使用“智能引号”则会使用西文字体。
- Users should use “smartquote” in Latin text; otherwise, “manual quotes” will result in CJK characters.

如果不喜欢该功能，可以将 `ctyp()` 函数的参数 `fix-smartquote` 设置为 `false`，则不修复智能引号字体。

3. 列表

3.1. 列表布局优化

fix-list-enum
设置是否修复列表样式。
默认值：`true`

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

```typst
#let (theme, _) = ctyp(
  fix-list-enum: false,
)
#show: theme
```

3.2. 列表布局设置

如果要修改默认的列表布局，可以通过 `ctyp()` 函数的参数 `fix-list-args` 和 `fix-enum-args` 来设置。这两个参数都是一个字典，包含了列表的样式设置，已尽可能兼容 Typst 提供的 `list` 和 `enum` 两个函数的参数。具体参数说明如下：

fix-list-args
列表样式设置。设置列表的样式。
默认值：(:)

参数	含义	默认值
`marker`	符号列表的备选符号，循环使用，遇到编号列后重置。	●, ▸, –
`numberer`	编号列表的编号格式，循环使用，遇到符号列表后重置。	1), a), i)
`tight`	是否使用紧凑布局。	`true`
`indent`	列表整体缩进。表现为左侧的边距。	`0pt`
`body-indent`	列表内容缩进。	`0.5em`
`spacing`	列表项目之间的间隔。如果使用紧凑布局，其值采用 `par.leading`；否则采用 `par.spacing`。	`auto`
`label-sep`	列表符号/编号与内容之间的间隔。	`0pt`
`marker-width`	符号的宽度。	`0.5em`
`number-width`	编号的宽度。	`1.5em`
`debug`	调试模式。	`false`
`..block-args`	捕获所有其他传递到 `block` 函数的参数。	

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。

page-grid
设置页芯大小，优先保证宽度为整字符数，避免过多分散对齐问题。

页面设置

```typst
#import "@local/ctyp:0.2.0": page-grid
#show: page-grid.with(
  width: 45,
  height: 70
)
```

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。

5. 标题编号设置

Typst 提供了实现任何标题编号格式的能力，但这种方法往往会比较复杂。中文文档的标题设置通常也具有一定的规律性，可以通过配置参数来实现。该包提供了 `ctyp()` 函数的参数 `heading-numbering` 来设置标题编号。该参数可以接受多种类型的值：

`heading-numbering`
标题编号设置。
默认值：`none`

- **单值**：用于统一设置所有级别的标题的编号格式。包括以下几种：
  - `none`：无编号。
  - 字符串：接受所有 Typst 支持的编号格式，即 `numbering()` 函数可接受的值。
  - 字典：用于设置编号格式的字典，包含以下字段：
    - `format`：字符串，表示编号格式。也是接受所有 Typst 支持的编号格式。
    - `sep`：间隔，表示编号与标题内容之间的间隔。可以是任何合法的长度值。
    - `align`：对齐方式，可以是 `left`, `center`, `right` 中的一个。默认值为 `left`。
    - `hanging-indent`：悬挂缩进。设置该值可调整标题悬挂缩进长度。默认为编号部分的宽度。
    - `first-line-indent`：首行缩进。设置该值可调整标题首行缩进长度。默认值为 `0em`。
    - `prefix`：前缀。表示编号前的内容，可以是任何内容。
    - `suffix`：后缀。表示编号后的内容，可以是任何内容。
- **数组**：数组中的每个元素都是上面可接受的单值。第 i 个元素（从 1 开始算）用于设置第 i 级标题的编号格式。如果数组长度小于标题级别，则使用数组中的最后一个元素来设置。

中文编号格式示例一

```typst
#let (ctypset_, cjk_) = ctyp(
  heading-numbering: ((
    format: "一、",
    sep: 0em
  ), "1.1")
)
#show: ctypset_
= 这是一级标题
== 这是二级标题
```

一、这是一级标题

1.1 这是二级标题

中文编号格式示例二

Typst

```
#let (ctypset_, cjk_) = ctyp(
  heading-numbering: ((
    format: "一",
    sep: 1em,
    align: center
  ), "1.1")
)
#show: ctypset_
= 这是一级标题
== 这是二级标题
```

一　这是一级标题

1.1 这是二级标题

附录 A　参考

CTyp

- ctyp()
- enumitem()
- page-grid()

ctyp

该函数设置了一个基本的 CJK 文档排版环境。它应用了 CJK 字体，设置了段落样式，并提供了 CJK 文本样式的实用函数。

参数

```
ctyp(
  fontset-cjk: auto str dictionary,
  font-latin: dictionary,
  font-cjk-map: dictionary,
  fix-list-enum: bool,
  fix-list-args: dictionary,
  fix-enum-args: dictionary,
  fix-smartquote: bool,
  fix-first-line-indent: bool,
  reset-strong-delta: int,
  remove-cjk-break-space: bool,
  heading-numbering: none str dictionary array
) -> array
```

fontset-cjk `auto` or `str` or `dictionary`

设置 CJK 字体集合。如果是 `auto`，使用默认的 Fandol 字体集合。如果是 `str`，使用预定义的打包字体集合中的一个。如果是 `dictionary`，则包含字体集合的详细信息，其结构如下：

`family` 一个从字形到字体名称的映射。包含以下字段：

- **`[shape]`** 代表字形的字符串（如 “song”, “hei”）。可以多次重复，以定义多种字形。对于 CJK 字体而言，字形可以有很多，可包括但不限于 “song”, “hei”, “kai”, “fang”。对于每个字形，其值都是一个字典，包含以下字段：
  - **`name`** 字体名称。必须是一个可用的、能被 Typst 识别的字体名称。
  - **`variants`** 一个定义有哪些变体的数组（如 `["bold", "italic"]`）。

`map` 一个元素到字形的映射的字典。包含以下字段

- **`[element]`** 代表元素的字符串（例如 `"text"`, `"strong"`）。可以多次重复以定义多个元素。目前仅支持以下元素 `text`, `emph`, `strong`, `raw`, `heading`。每个元素的值都是一个字典，包含以下字段：
  - **`cjk`** CJK 字体标识，格式为 “shape:variant”。其中 `shape` 是 `family` 中的键，`variant` 是 `variants` 中的值。
  - **`latin`** Latin 字体标识，是预定义的字符串之一：“serif”, “sans”, “mono”。

默认值: `auto`

font-latin `dictionary`

指定西文字体。必须是一个字典，将西文字体形状映射到字体名称。具有以下字段：

`serif` 衬线字体名称。默认为 “Libertinus Serif”。

`mono` 等款字体名称。无默认。

`sans` 无衬线字体名称。默认为 “DejaVu Sans Mono”。

默认值: `(:)`

font-cjk-map `dictionary`

修改 CJK 字体映射表。具有以下字段：

`[element]` 代表元素的字符串（例如 `"text"`, `"strong"`）。可以多次重复以定义多个元素。目前仅支持以下元素 `text`, `emph`, `strong`, `raw`, `heading`。每个元素的值都是一个字典，包含以下字段：

- **`cjk`** CJK 字体标识，格式为 “shape:variant”。其中 `shape` 是 `family` 中的键，`variant` 是 `variants` 中的值。
- **`latin`** Latin 字体标识，是预定义的字符串之一：“serif”, “sans”, “mono”。

默认值: `(:)`

fix-list-enum `bool`

是否修正列表和枚举的样式。如果为 true，将应用 `fix-list-args` 和 `fix-enum-args` 中定义的样式。

默认值: `true`

fix-list-args `dictionary`

接受一个字典，定义列表样式的参数。详细参数见 `enumitem()` 函数。

默认值: `(:)`

fix-enum-args `dictionary`

接受一个字典，定义枚举样式的参数。详细参数见 `enumitem()` 函数。

默认值: `(:)`

fix-smartquote `bool`

是否修正智能引号。如果为 true，将自动将引号转换为智能引号。

默认值: `true`

fix-first-line-indent `bool`

是否修正列表的首行缩进。如果为 true，将应用 `fix-first-line-indent` 中定义的首行缩进。

默认值: `true`

reset-strong-delta `int`

重置粗体的 delta 值为 0。基于此实现在 `font-cjk-map` 中指定元素的字重。

默认值: `0`

remove-cjk-break-space `bool`

是否移除 CJK 字符之间的断行空格。如果为 true，将移除 CJK 字符之间的断行空格。这用于防止 CJK 字符被空格断开。

默认值: `true`

heading-numbering `none` or `str` or `dictionary` or `array`

设置标题编号的样式。可以设置为单值或者数组。

如果是单值，则用于所有级别的标题。接受的合法类型为：

- `none`：无编号。
- `str`：字符串，接受所有 Typst 支持的编号格式，即 `numbering()` 函数可接受的值。
- `dictionary`：用于设置编号格式的字典，包含以下字段：
  - `format`：字符串，表示编号格式。也是接受所有 Typst 支持的编号格式。
  - `sep`：间隔，表示编号与标题内容之间的间隔。可以是任何合法的长度值。
  - `align`：对齐方式，可以是 `left`, `center`, `right` 中的一个。默认值为 `left`。
  - `first-line-indent`: 首行缩进值。可以是任何合法的长度值。默认值为 `0em`。
  - `hanging-indent`: 悬挂缩进值。默认值为 `auto`，通过测量编号宽度自动设置。也可以是任何合法的长度值，此时不在测量编号宽度，而是使用指定的值。
  - `prefix`：前缀，表示编号前的内容。可以是任何合法的内容。
  - `suffix`：后缀，表示编号后的内容。可以是任何合法的内容。

如果是数组，则数组中的每个元素都是上面可接受的单值。当标题层级数大于数组长度时，使用数组中的最后一个元素来设置更高一级的标题。

默认值: `none`

enumitem

自定义列表和枚举布局，修复符号和文字不对齐的问题。

参数

```
enumitem(
  marker: array,
  numberer: array,
  tight: bool,
  indent: length,
  body-indent: length,
  spacing: auto length,
  label-sep: length,
  marker-width: length,
  number-width: length,
  debug: bool,
  ..block-args: arguments,
  children: array
)
```

marker `array`

符号列表可选用的符号。将循环使用。

默认值: `(sym.circle.filled, sym.triangle.r.filled, sym.dash)`

numberer `array`

编号列表可选用的编号格式。将循环使用。

默认值: `("1)", "a)", "i)")`

tight `bool`

是否使用紧凑布局。紧凑布局会使用 `par.leading` 作为列表项目之间的间隔，否则使用 `par.spacing`。

默认值: `true`

indent `length`

列表整体缩进。表现为左侧的边距。

默认值: `0em`

body-indent `length`

列表内容缩进。

默认值: `0.5em`

spacing `auto` or `length`

列表项目之间的间隔。

默认值: `auto`

label-sep `length`

列表符号/编号与内容之间的间隔。

默认值: `0em`

marker-width `length`

符号的宽度。

默认值: `0.5em`

number-width `length`

编号的宽度。

默认值: `1.5em`

debug `bool`

是否开启调试模式。调试模式会在列表项目的边框上显示红色和绿色的边框，以便于调试列表布局。

默认值: `false`

..block-args `arguments`

捕获所有其他传递到 `block` 函数的参数。

children `array`

列表的子元素。

page-grid

根据页芯网格大小设置页面边距

参数

```
page-grid(
  width: int,
  height: int,
  note-left: int length,
  note-right: int length,
  ..args: arguments,
  body: content
)
```

width `int`

页芯宽度，单位为字符数。

默认值: `42`

height `int`

页芯高度，单位为字符数。

默认值: `65`

note-left `int` or `length`

页脚左侧的注释宽度，单位为字符数。

默认值: `auto`

note-right `int` or `length`

页脚右侧的注释宽度，单位为字符数。

默认值: `auto`

..args `arguments`

其他参数，传递给 `page()` 函数中的 `margin` 参数。

body `content`

页面内容。

附录 B　预定义的字体集合

集合名称	字形	字体标识	中文名称
fandol	song	FandolSong	
	hei	FandolHei	
	kai	FandolKai	
	fang	FandolFang R	
fangzheng (方正字库)	song	FZNewShuSong-Z10S	方正新书宋简体
	hei	FZHei-B01S	方正黑体简体
	kai	FZKai-Z03S	方正楷体简体
	fang	FZFangSong-Z02S	方正仿宋体简体
	songhei	FZSongHei-B07S	方正宋黑简体
	cusong	FZCuSong-B09S	方正粗宋简体
	xiaobiaosong	FZXiaoBiaoSong-B05S	方正小标宋简体
	dabiaosong	FZDaBiaoSong-B06S	方正大标宋简体
	dahei	FZDaHei-B02S	方正大黑简体
windows (Windows 字库)	song	SimSun	宋体
	hei	SimHei	黑体
	kai	KaiTi	楷体
	fang	FangSong	仿宋
	lishu	LiSu	隶书
	youyuan	YouYuan	幼圆
huawen (华文字库)	song	STSong	华文宋体
	hei	STXiHei	华文细黑
	kai	STKaiti	华文楷体
	fang	STFangsong	华文仿宋
	zhongsong	STZhongsong	华文中宋

集合名称	字形	字体标识	中文名称
	lishu	STLiti	华文隶书
	caiyun	STCaiyun	华文彩云
	hupo	STHupo	华文琥珀
	xingkai	STXingkai	华文行楷
	xinwei	STXinwei	华文新魏
source (思源字体)	song	Source Han Serif SC	思源宋体
	hei	Source Han Sans SC	思源黑体
noto	song	Noto Serif CJK SC	
	hei	Noto Sans CJK SC	