

# markdown 转 pdf typst 模板

2024/12/21

## 目录

1. 模板说明 .....	1
1.1. 支持的样式 .....	2
1.2. 使用工具的版本 .....	2
1.3. 模板包目录下的内容 .....	2
1.4. 模板中的默认设计 .....	4
1.4.1. pdf 页面 .....	4
1.4.2. 字体 .....	5
1.4.3. 默认字体大小 .....	7
1.4.4. 缩进 .....	7
1.4.5. 代码高亮颜色主题 .....	8
1.4.6. 样式修改 .....	8
1.5. pandoc 模板程序使用方法 .....	8
1.5.1. 运行 .....	10
1.6. themekit 使用方法 .....	10
1.6.1. 运行 .....	11
1.6.2. 与 sublime text 集成 .....	11
2. front matter .....	13
2.1. 文章参数 .....	13
2.2. 页面布局样式参数 .....	14
2.3. 内容插入参数 .....	16
3. 限制 .....	16
3.1. !! pandoc 的限制 .....	17
3.1.1. 重复脚注引用的问题 .....	17
3.2. typst 的限制 .....	17
3.2.1. 首行缩进 .....	17
3.2.2. 图片文件 .....	17
3.2.3. 样式主题文件 .....	18
3.2.4. 斜体 .....	18
3.2.5. 表格行数 .....	18
4. 技巧 .....	18
4.1. 缩小 pdf 文件体积 .....	18

4.2. typst 导出 png 图片时不够清晰 .....	19
4.3. 隐藏章节标题 .....	19
4.4. 隐藏页码 .....	19
4.5. 长页(不分页) .....	19
5. 效果示例 .....	20
5.1. 段落与换行 .....	20
5.2. 强调 .....	21
5.3. 有序列表 .....	22
5.4. 无序列表 .....	24
5.5. 任务列表 .....	25
5.6. 忽略 Markdown 格式 .....	25
5.7. emoji .....	25
5.8. 链接 .....	26
5.9. 图片 .....	27
5.10. 脚注 .....	29
5.11. 定义列表 .....	30
5.12. 数学公式 .....	31
5.13. 代码和语法突出显示 .....	32
5.14. 表格 .....	34
5.14.1. 表格标题 table caption .....	37
5.14.2. simple tables .....	38
5.14.3. 多行表格 .....	38
5.14.4. grid tables .....	39
5.15. 表格宽度与列宽 .....	41
5.16. 引用 .....	44
5.17. alerts .....	45
5.18. 水平线 .....	48
5.19. 内联 HTML .....	48
5.20. 标题 .....	49
6. (演示需要, 请忽略) h1 标题 .....	49
6.1. h2 标题 .....	49
6.1.1. h3 标题 .....	49
7. (演示需要, 请忽略) Alt-H1 .....	49
7.1. Alt-H2 .....	49

# 1. 模板说明

本模板用于 pandoc markdown 转 pdf 文档。故而，本模板有一定的预置性。则本模板会影响生成 pdf 文档中的排版布局。

1. **本模板适用于**：模板中预置样式，适用于个人或者团队编写技术文档，说明文档，用于交流或者分享，体现个性化。也可用于学习如何对 typst 中的排版与格式进行编程。
2. **本模板暂不适用于**：不针对学术、论文这在严格格式的领域，如缺少 [摘要](#)、[引文](#)、[科学数学图表](#)等专用类型。不过你可以基于本模板进行修改。（需要懂 typst 语法）

## 优势

1. 用于快速将 markdown 转换为 pdf。要求的工具 pandoc 与 typst 两个命令行工具。
2. 支持常见的样式，用于替代 word 或者其它重量级写作工具，来编写轻量级文档如（工作总结、个人日记、团队任务计划等一般常见的短中篇文章）
3. 使用 typst 来替代 latex，避免大炮打蚊子，typst 是当前生成 pdf 文档最轻量、效率最高的方案
4. 无须将 markdown 先转为 html 或 docx，再用浏览器或者 word/wps 来转换成 pdf。
5. 脱机使用，在转换生成 pdf 文件时，无须联网。

## 劣势

因使用场景设定，不支持专业的写作要求，无法生成论文、学术报告、书籍出版对排版与格式有更高要求的写作物。因为模板设定在脱机环境下使用，因此，模板没有使用 typst universe 上扩展包。不过，这个扩展可以由你自己完成。

### Caution

本模板的初衷，是为了可在 sublime text 编辑器中编写 markdown 文档后，快速生成一份风格 **自以为是** 的 pdf 文档，用于信息交流或者分享。因此，它并不为了成为写作工具而存在。它仅定义了一些常用的文档结构类型的 typst 排版布局和渲染排版，不覆盖 pandoc 所有的 markdown 属性，当前具体支持的样式见 [支持的样式](#)

如需要更加全面的写作支持，可使用套件 quarto（支持 Scholarly Writing、Presentations、Dashboards、Books、Manuscripts、出版 多个领域的用户）。

## 1.1. 支持的样式

- [段落与换行](#)
  - [强调 \(斜体粗体\)](#)
  - [无序列表](#)
  - [有序列表](#)
  - [任务列表](#)
  - [表情符号 emoji](#)
  - [链接](#)
  - [图片](#)
  - [脚注](#)
  - [定义列表](#)
  - [数学公式](#)
  - [代码高亮 fenced code block](#)
  - [表格](#)
  - [表格宽度与列宽](#)
  - [引用 quotelblocktext](#)
  - [alerts](#)
  - [内联 HTML](#)
  - [水平线](#)
  - [标题](#)
- 

## 1.2. 使用工具的版本

1. pandoc 3.5 下载与安装说明, <https://pandoc.org/installing.html>
2. typst 0.12.0 下载与安装说明, <https://github.com/typst/typst/releases/tag/v0.12.0>

先下载 pandoc、typst, 并将其放至可以查找到的系统环境变量 PATH 中。

- macos/linux 操作系统: 可使用类似于 brew、apt、mise、asdf 等命令行软件工具进行安装。
- windows 操作系统: 可使用 scoop、Chocolatey 等命令行软件工具进行安装。或者下载后, 分别将 pandoc 与 typst 可执行程序所在目录, 追加到系统变量 PATH 。

## 1.3. 模板包目录下的内容

```
|—— pandoc
|   |—— example.sh
```



模板包含两种发行形式, `pandoc` 与 `themekit`, 它们分别是两种不同组织方式, 但使用它们所转换得到的 `pdf` 文件、`png` 图片效果是相同的。

- 使用 `pandoc` 包中的模板, 可生成 `typst` 独立文件, 方便用于独立调试或者修改 `typst` 结果文件。但对 `themes` 代码高亮目录的位置有要求。
- `themekit` 要求先安装成 `typst local package`, 安装方法见 `howto.themekit.pdf` 中的说明。`themekit` 包中的 `pandoc typst` 模板使用引用 `typst package` 的方式使用各种 `rules` 定义, 方便用于生成目标格式文件。

#### `pandoc` 模板包:

1. `pandoc` 目录下是一个个独立的 `pandoc typst` 模板。每个模板文件, 分别设置了不同的颜色与主题。
2. `themes` 目录下的 `.tmTheme` 文件, 是 `sublime text 3.1` 之前使用的编程语言高亮颜色主题文件, 本身是 `XML` 格式。`typst` 使用它来定义代码高亮的颜色主题。这些主题文件, 在 `pandoc typst` 模板中被引用。

#### `themekit` 模板包:

1. `styled` 存放了按页面大小、预定颜色、高亮主题的 `typst show` 或 `set` 的规则, 每个文件之间均独立, 它们的 `themekit` 的 `lib.typst` 中被重新导出。
2. `pandoc` 目录存放了 `styled` 中规则的引用, 可用于在 `pandoc` 命令行 `--template` 参数。每个文件对应一种高亮主题。

#### 说明或帮忙文档:

1. `pandoc.vs.themekit.pdf`, 介绍了两个模板包的差别。
2. `howto.pandoc.pdf`, `howto.themekit.pdf` 介绍模板包如何上手使用方法。
3. `userguide`, 即本用户帮助说明文档, 相较于 `howto`, 本文档更加详细。

#### i Tip

模板文件的命名为 `[phone.]MODE.THEMES.typst`。若以 `phone` 开头, 生成的 `pdf` 页宽 300 磅, 便于在手机上阅读。`MODE` 为 `dark` 或 `light`, 对应页面深色或浅色背景, `THEMES` 即为代码高亮颜色主题,

## 1.4. 模板中的默认设计

### 1.4.1. pdf 页面

本模板包中, `pdf` 页面大小有两个尺寸, 分别为 `A4` 与 `phone`。`A4` 用于正常的文档, `phone` 尺寸可用于在分享到移动设备时, 便于阅读。具体尺寸如下:

页面	mode	尺寸	边框留白
a4	dark/ light	595.28 磅 × 841.89 磅	上下左右 2.5 厘米
phone	dark/ light	300 磅 × 650 磅	上左右 12 磅, 下 24 磅

**深色**页面背景颜色值 `#10141c`，字体颜色值 `#fbdbb6`，**浅色**页面背景颜色值 `#fcfcfc`，字体颜色值 `#5c6166`。

### 1.4.2. 字体

文档中有两种字体，一种是全文使用的统一字体，另一种用于显示代码的等宽字体，叫 mono 字体。

- 主字体: 模板中使用了 `霞鹜文楷 GB` 做为主字体，它显示的中文更加优雅。
- 等宽字体: 本模板包使用 `霞鹜文楷等宽 GB` 作为等宽字体，西文效果与中文效果字迹比较一致。若喜欢 `fira code` 等编程字体的，可以修改。

霞鹜文楷是一款多版本的系列字体，本模板版本的字体是 `LXGW WenKai GB`，对应英文名称为 `LXGW WenKai GB`。使用本模板前，需要先下载这些字体并安装到系统中。霞鹜文楷 GB 下载地址 [GitHub 项目网](#)，[猫啃网](#)

**霞鹜文楷 GB** 字重分 `light`(细), `regular` (常规), `medium` (中粗)，总体风格较为纤细。且没有自带斜体字形，在 `typst` 生成的 pdf 中斜体风格无法显示。

若以上字体满足不了需求的，可根据需要修改。`pandoc` 模板包的，修改方法如下：

```
// // for windows
// // fonts for emoji
// #let ememoji_fonts=("Segoe UI Emoji", "Noto Color Emoji")
// #let mainfont = ("Segoe UI",)
// #let monofont = ("Consolas",)

// for macos
// fonts for emoji
#let ememoji_fonts=("Apple Color Emoji", "Noto Color Emoji")
#let mainfont = ("SF Pro Text",)
#let monofont = ("SF Mono",)

// // for linux
```

```

// // fonts for emoji
// #let emmoji_fonts=("Noto Color Emoji")
// #let mainfont = ("Menlo",)
// #let monofont = ("DejaVu Sans Mono")

#let lang = "zh"
#let region = "CN"
#if lang == "zh" {
  mainfont=("PingFang SC", )
  // mainfont=("Microsoft YaHei UI", )
}

// fonts defined
#let fonts=(
  normal: (
    // typst 0.12: typst
    // normal font for whole doc
    font: (name: ("LXGW WenKai GB", ..mainfont, "Libertinus
Serif", ..emmoji_fonts))
  ),
  codeblock: (
    // mono font for code snippets
    font: (name: ("LXGW WenKai Mono GB", ..monofont, "DejaVu Sans
Mono", ..emmoji_fonts)),
  ),
  math: (
    // typst built-in math font, New Computer Modern Math
    font: (name: ("New Computer Modern Math", "LXGW WenKai Mono
GB", ..emmoji_fonts)),
  )
)
)

```

修改 `mainfont`、`monofont`、`LXGW WenKai GB`、`LXGW WenKai Mono GB` 为你操作系统或者其它默认的字体。

`themekit` 模板中的字体设置，位于包内的 `fonts.typst` 文件中，内容同上。

可以在 front matter 中设置 `mainfont`、`monofont` 参数，具体参数见到 [front matter 参数](#)。



### i Tip

1. 若运行提示字体不存在, `unknown font family`, `typst` 会回退默认的系统字体, 不会中断。但不优雅。
2. `typst` 的字体机制, 从设置的字体组的第一个字体族中查找, 若找不到, 即在下个字体族中查找。可能会在显示表情符号时会出现不一致的情况。如 `△` 这个符号, 使用到的 `LXGW WenKai GB` 中的线框符号风格, 而不是 `Apple Color Emoji` 中的 `emoji` 风格, 这并不是 `typst` 的程序错误。你可以调整字体族顺序, 或者使用其它字体来解决。

### 1.4.3. 默认字体大小

- 正文: 12 磅 (小四)
- 标题
  - 文章标题: 28pt
  - 1 号标题: 20pt
  - 2 号标题: 16pt
  - 3 号标题: 14pt
  - 4 号标题: 12pt
  - 5 号标题: 12pt
  - 6 号标题: 12pt
- 代码块: 11pt
- 引用文本块: 12pt

### 1.4.4. 缩进

1. 文字段落首行缩进**两个汉字宽度**。
2. 引用 (文本块) 整体右移缩进**两个汉字宽度**。
3. 有序列表、无序列表, 在正文中整体右移缩进**两个汉字宽度**。
4. alerts、引用、定义列表 中, 多个文本段落, 除**第一段首行无法缩进**<sup>1</sup>外, 其他后续段落首行缩进 **两个汉字宽度**。当仅有序列表、无序列表单独出现时, 为保持视觉紧凑, 有序列表、无序列表不缩进。

---

<sup>1</sup>这些情况在 alerts、引用、定义列表 这类使用 `block` 包裹内容的元素出现。它与 `typst` 本身的实现机制有关, 多数元素与文本布局、排版控制, 使用了 `block`, `block` 后的正文段可以通过特殊手段处理来实现, 但 `block` 内的开头, 暂没有办法。

### ⚠ Warning

typst 0.12 仍未解决连续多段文本段落第一段标识的问题，故文本段第一段首行无法控制缩进。

#### 1.4.5. 代码高亮颜色主题

本模板包中，目前包含的现代化、美观的代码高亮主题有：

- [ayu](#):
  - ayu-mirage
  - ayu-dark
  - ayu-light
- [Catppuccin](#):
  - Catppuccin Frappe
  - Catppuccin Latte
  - Catppuccin Macchiato
  - Catppuccin Mocha

在 pandoc typst 模板文件中，`#set raw(theme: )` 指定。

#### 1.4.6. 样式修改

在每个独立的 pandoc typst 模板文件的前面，定义了全文档可使用的全局变量，这里集中了模板中大部分可变的值，比如 [字体大小](#)、[字体名称](#)、[颜色](#) 等。

1. 字体：见上文 [字体](#) 一节
2. 代码高亮主题：本模板包中，已经列举了所有颜色主题，并一一对应一个模板，根据偏好，选择对应的模板文件即可。如有自定义的 `*.tmTheme` 样式文件，可按 [代码高亮主题](#) 中的方法修改。

### 1.5. pandoc 模板程序使用方法

本模板程序目录没有界面操作的方法，模板包在使用 pandoc 命令时指定。在 [typst 的限制](#) 中已描述了图片引用、代码高亮主题引用的问题，若想同时兼容两者问题，需要模板中的 `themes` 目录与图片目录、markdown 文件存在同一主目录，或者往上要有相同的祖先目录。

要求目录结构如下[情况 1](#)：

```
my_notes
├── images
│   ├── a.png
│   └── ...
```

```
├─ themes
|   └─ 一些 tmTheme 文件
|       └─ ...
├─ notes.md
```

## 或者情况 2

```
my_notes
├─ content
|   └─ images
|       └─ a.png
|           └─ ...
|   └─ notes.md
|   └─ ...
├─ themes
|   └─ 一些 tmTheme 文件
|       └─ ...
├─ typst
|   └─ 一些 pandoc typst 模板文件
|       └─ ...
```

在 `my_notes` 这个目录位置启动 `pandoc`，`typst` 才可能同时访问到 `images` 和 `themes` 目录。

但通常我们的目录是这样的：

```
top
├─ my_notes
|   └─ images
|       └─ a.png
|           └─ ...
|   └─ notes.md
├─ pandoc_typst
|   └─ themes
|       └─ 一些 tmTheme 文件
|           └─ ...
|   └─ typst
|       └─ 一些 pandoc typst 模板文件
|           └─ ...
```

当然，我们可以在 top 目录下运行 pandoc，但必须满足以下情况：

1. notes.md 中的图片引用路径是 my\_notes/images/xxx.png
2. typst 模板文件中 #set raw(theme: ) 必须是  
`#set raw(theme: "pandoc_typst/themes/xxx.tmTheme")`

怎么看，都觉得别扭。

可以在 my\_notes 目录下，创建一个符号链接，指向 pandoc\_typst/themes 目录，这是最简单直接的方法。方法如下。

### 1.5.1. 运行

命令：

```
# linux / macos
cd 内容目录
template_dir=模板文件包目录绝对路径
template=$template_dir/pandoc/pandoc.dark.catppuccin-frappe.typst
md_args="+emoji+strikeout+definition_lists+implicit_figures+fancy_lists+yaml_
metadata_block"
# 如何没有 themes 目录，创建软链接
[ ! -d themes ] && ln -s $template_dir/themes
cat markdown 文件.md | pandoc -f gfm"$md_args" -t pdf --pdf-engine typst \
    --template=$template -o markdown 文件.pdf
# pdf 生成后，删除链接
[ -L themes ] && rm themes;
```

1. 将 themes 目录链接到内容目录下，这样对 typst 来说，需要引用的图片、tmTheme 文件等资源，均在同一个目录下
2. 使用 `cat markdown.md | pandoc -f gfm` 或者 `pandoc markdown.md -f gfm`，因为 markdown.md 文件可能需要预处理，如需要使用 multimarkdown 预处理 markdown 文件，即可以这样 `multimarkdown -tmmd markdown.md | pandoc`

## 1.6. themekit 使用方法

themekit 所生成的 pdf 文件效果与 pandoc 模板包相同。采用 typst package 的形式来引用。使用 themekit 包没有 themes tmTheme 高亮主题文件位置位置，因为它可以做为 typst 包一部组件而存在，只要先将 themekit 安装为 typst local package 即可。

typst local package 包的目录规范是：{data-dir}/typst/packages/{namespace}/{name}/{version}，不同操作系统，{data-dir} 不同，具体如下：

- linux: \$XDG\_DATA\_HOME or ~/.local/share

- maxos: ~/Library/Application Support
- whidowns: %APPDATA%

根据习惯, {namespace} 为 local, 本模板包的名称为 themekit, {version} 对应 themekit 发布的最新版本号。

将 themekit 目录复制或链接到 {data-dir}/typst/packages/local 目录下, 如:

```
mkdir -p "{data-dir}/typst/packages/local"
cp -R themekit "{data-dir}/typst/packages/local"
# 或者
ln -s themekit "{data-dir}/typst/packages/local/themekit"
```

将 {data-dir} 修改为你操作系统对应的目录。

### 1.6.1. 运行

命令:

```
cd markdown/文档所在目录
template="{data-dir}/typst/packages/local/themekit/版本号/pandoc/目录下的任意一个 pandoc 模板.typst"
pandoc_args="markdown_mmd+emoji+strikeout+fancy_lists+task_lists+implicit_figures+link_attributes+inline_notes+tex_math_single_backslash+simple_tables+grid_tables+multiline_tables+table_captions+yaml_metadata_block-blank_before_blockquote"
cat markdown 文档.md |
  pandoc --from $pandoc_args \
  -t pdf --pdf-engine typst --template="$template" \
  -o "输出文件名称.pdf"
```

将 {data-dir} 修改为你操作系统对应的目录。pandoc 目录下的任意一个 typst 模板文件, 按 [页面尺寸大小](#)、[dark/light 背景](#)、[代码高亮主题](#) 区分。

目录下有示例 shell 脚本, 见 {data-dir}/typst/packages/themekit/版本号/examples.sh

### 1.6.2. 与 sublime text 集成

将上述的命令行, 集成到 sublime text 的 build 系统, 可以快速的执行转换, 而无须切换到命令行终端。

1. **第一步**: 新建一个 sublime text 构建系统, 方法如下: A. 菜单 setting/Browse Packages..., 在 Packages/User 下创建一个空白文本文件

markdown.sublime-build B. `cmd + shift + p`, 打开命令面板, 输入 `new build system`, 保存文件为 `markdown.sublime-build`

2. **第二步**: 在 `markdown.sublime-build` 文件中, 保存如下内容:

```
{
  "selector": "text.html.markdown",
  "variants": [
    {
      "name": "md to pdf (by typst)",
      "working_dir": "$file_path",
      "env": {
        // 修改为 themekit 所在目录`
        "THEMEKIT_DIR": "path/to/themekit",
        "PANDOC_MD_ARGS": "--from=markdown_mmd+emoji+strikeout+fancy_
lists+task_lists+implicit_figures+link_attributes+inline_notes+tex_math_
single_backslash+simple_tables+grid_tables+multiline_tables+table_captions+
yaml_metadata_block-blank_before_blockquote",
        "PANDOC_TARGET": " -t pdf --pdf-engine typst ",
        // 修改为对应高亮主题模板
        "PANDOC_TEMPLATE": "themekit/版本号/pandoc/dark.catppuccin-
frappe.typst"
      },
      "shell_cmd": "cat \"$file\" | pandoc \\$PANDOC_MD_ARGS \\$PANDOC_
TARGET --template=\\$THEMEKIT_DIR/\\$PANDOC_TEMPLATE -o $file_path/$file_base_
name.pdf",
    },
  ]
}
```

将 `THEMEKIT_DIR` 参数设置为模板包在本地的目录路径, 然后在 `sublime text` 中随意编辑一个 `markdown` 文件, `cmd + shift + p` 打开命令面板, 输入 `pdf typst`, 并回车, `markown` 文件同目录下, 生成了一个 `pdf` 文件。`PANDOC_TEMPLATE` 参数指定了要使用哪一个模板, 每个模板都有特有的预定颜色。

还可以使用 `pandoc` 与 `typst`, 生成为 `pdf` 样式相同的 `png` 图片。

`themekit` 中包含 `sublime text markdown build` 的示例文件 `example.markdown.sublime-build`。

## 2. front matter

pandoc 解释 markdown 文档时，支持解释 front-matter 中的参数，我们可以使用这些参数，来修改文档模板样式。

本文档模板支持的参数有：

### 2.1. 文章参数

#### title

文章标题，类型为 `文本`。

#### subtitle

文章副标题，类型为 `文本`。

#### author

文章作者。类型为 `文本`，`对象`，或 `文本数组`、`对象数组`。当有多个署名时，使用 `,` 分隔，并使用 `[]` 将多个名称包裹起来。当使用对象来描述作者时，支持 `name`、`affiliation`、`email` 三个属性。如，多个作者名称：

```
# 单个姓名
author: 作者姓名
# 多个姓名
author: [作者1, 作者2]
```

使用对象来描述作者：

```
# 单个姓名
author:
  name: 作者名称
  affiliation: 可空
  email: 电子邮件, 可空
# 多个姓名
author:
- name: 作者名称
  affiliation: 可空
  email: 电子邮件, 可空
- name: 作者名称
  affiliation: 隶属, 可空
  email: 电子邮件, 可空
```

#### date

文章日期，类型为 `文本`

#### lang

语言与区域参数，格式为 `语言-区域代码`，`lang` 值为 [ISO 639-1/2/3 language code](#)。`lang` 设置 `typst` 中的语言属性，影响 `Typst` 的各个部分，

- 文本处理管道可以做出更明智的选择
- 连字符将使用该语言的正确模式
- [智能绰号](#)会变成该语言的正确引号
- 以及所有其他具有语言感知能力的内容

如 `lang` 为 `zh` 时，目录标题为 [目录](#)，当 `lang` 为 `en` 时，目录标题为 [Contents](#)。`region` 可以让 `typist` 在处理文本时做出更明智的选择。值为 [ISO 3166-1 alpha-2 region code](#)

### abstract

摘要，类型为 `文本`，因为摘要可能有多行的情况，遵守 `yaml` 的多行语法，如

```
abstract: |
  This is the abstract.

  It consists of two paragraphs.
```

### citations

引用，见 [Citation rendering](#)

### bibliographystyle/csl

`bibliography` 的格式，类型为 `文本`，值见 [bibliography style](#)

### bibliography

书目或 参数文献，类型 `文本`，指向 `Hayagriva.yml` 或 `BibLaTeX .bib` 文件的路径。路径相对规则遵守 `typst` 的路径安全规定。

## 2.2. 页面布局样式参数

### mainfont

默认字体，类型为 `文本` 或 `文本数组`，列举文中使用的字体。需要注意的，所列字段在当前操作系统中已经存在，否则会出现无法字体的警告。如下面例子，优先使用 `Fira Code` 中的字体，中文使用 `PingFang SC` 字体：

```
mainfont: [Fira Code, PingFang SC]
```

### monofont

等宽字体，类型为 `文本` 或 `文本数组`，指定生成的 `pdf` 文档中的 `code block` 代码字体，一般选用适合开发者友好的编程字体。格式同 `mainfont`。

### mathfont



数学公式字体，类型为 `文本` 或 `文本数组`，格式同 `mainfont`。用于设置非英文的字体，如公式中的中文。

### fontsize

字体大小，类型为 `typst length`。如 `12pt`。

### papersize

页面尺寸，`typst` 中预定页面大小的名称，常见的有 `"a4"`，`"us-letter"` 等，详见 `typst page paper`。`papersize` 会覆盖文档模板原有页面大小，当设置 `pagesize` 时，可覆盖本 `papersize` 页面大小。

### pagesize

页面大小，见 `typst page width` 与 `page height` 参数，可覆盖 `papersize`。如下实际的页面尺寸为宽 `400pt`、宽 `800pt`，不管 `papersize` 设置了何值。当 `pagesize: {height:auto}` 时，不分页，可用于生成长页。

```
papersize: presentation-16-9
pagesize:
  width: 400pt
  height: 800pt
```

### margin

页面边距，见 `typst page margin` 参数，默认值见 `pdf 页面` 中说明。

### page-numbering:

页码格式，见 `typst page numbering` 参数。设置为 `disable` 或 `none` 时不显示页码。

### section-numbering:

标题编号，见 `typst heading numbering` 参数。当 `section-numbering` 设置为 `disable` 或 `none` 时，可隐藏标题编号。

### columns

页面分栏数，默认为 `1`，见 `typst page columns` 参数。

### toc

目录，类型为 `bool`，默认为 `false`。当 `toc: true` 时，显示目录。当 `pagesize.height` 为 `auto` 时，不显示目录。

### toc-depth:

目录层次级别，控制可显示的标题层次，类型为数字，见 `typst outline depth` 参数

### toc-title:

目录标题，当 `toc-title: auto` 时，根据 `lang` 自动定义，如 `lang: zh-CN` 时，标题为 目录；当 `toc-title: none` 时，目录标题不显示。见 [typst outline title 参数](#)。

### table-caption-position:

设置表格标题相对于表格的位置，默认位于表格上面，见 [typst figure caption position 参数](#)。

### figure-caption-position:

设置图片标题相对于图片的位置，默认位于图片下方，见 [typst figure caption position 参数](#)。

## 2.3. 内容插入参数

### header-includes

pandoc 参数，文本内容插入于目录之后，正文之前，通常有于 markdown 转 html 时的头部。javascript、css 引用。见 [pandoc Variables set automatically](#)，也可使用 `pandoc` 命令行参数 `-H/--include-in-header` 指定内容文件。

### include-before

pandoc 参数，文本内容插入到目录之前，见 [pandoc Variables set automatically](#)，也可使用 `pandoc` 命令行参数 `-B/--include-before-body` 来指定内容。

### include-after

pandoc 参数，文本内容插入到正文之后，位置文档末尾，见 [pandoc Variables set automatically](#)，也可使用 `pandoc` 命令行参数 `-B/--include-after-body` 来指定内容。

#### i Tip

使用 `pandoc` 命令时，需要使用 `--from gfm` 或者 `--from markdown_mmd` 或 `--from markdown+yaml_metadata_block` 参数。

## 3. 限制

本手册不是 markdown 规范说明，也不是 typst 语法说明书。因此本手册不会包括 markdown、typst 语法说明。

## 3.1. !! pandoc 的限制

### 3.1.1. 重复脚注引用的问题

pandoc 存在一个 10 年仍未解决的 bug [issue 1603](#), 无法正确解析重复脚注。

## 3.2. typst 的限制

### 3.2.1. 首行缩进

typst 0.12 暂不支持识别第一段, 故无法对第一段进行缩进, 与它的内部实现有关。typst 中的元素, 非常多使用 block 做为最外层容器来实现整体布局。即使 `set par(first-line-indent:)` 是指连续段落间的首行, 即不包含第一段首行。如果是 block 或 box 后的段落, 可以通过特殊手段变通应对, 但 block 或 box 内的文本首段, 目前还找不到方法。在 [alerts](#) 和 [引用](#) 中, 当文本内容比较长或者有多段落时, 会看到其中的第一段首行不缩进。

期待 typst 在未来的版本中能简单解决, 这样可以去掉一些为了兼容的垃圾代码。

### 3.2.2. 图片文件

为了保证 typst 脚本执行安全无害, typst 自带沙箱机制。`.typst` 文件编译成 pdf 时, 它引用的图片文件, 不能使用 url 路径, 并且图片只能位于 `.typst` 文件所在目录或子目录, 使用相对路径指定图片文件位置。这与 markdown 文件中无限制的路径写法不同。需要特别注意。

图片文件必须位置 `*.typst` 内容文件同一个顶级目录下, 层次无要求, 只要 `*.typst` 中的相对路径层级对应正确即可。如:

```
my_notes
├── images
│   ├── a.png
│   └── ...
├── readme.typst
└── readme.md
```

假设图片文件位于 `images` 目录下, 那么图片文件相对于 `readme.typst` 内容文件的路径为 `./themes`, 即在 `readme.typst` 中指定图片路径时, 填写为:

```
#image("./userguide/images/a.png")
```

因使用 pandoc 转换 markdown, 即在 markdown 中指定图片引用的写法为:

```
![图片示例](./userguide/images/a.png)
```

按照这个相对路径，pandoc 通过 typst 将 markdown 转换为 pdf 时，可正确转换。

### 3.2.3. 样式主题文件

与图片文件引用相同，代码高亮颜色主题文件，也是通过相对路径引用。目前没有办法可以将主题文件直接嵌入到 \*.typst 文件中。

样式主题文件必须与 \*.typst 位于同一个顶级目录下，层次无要求。只要 \*.typst 中的相对路径层级正确即可。如：

```
pandoc_typst
├── themes
│   ├── 一些 tmTheme 文件
│   └── ...
└── readme.typst
```

若主题文件位于 `themes` 目录下，那么 `.tmTheme` 相对于 `readme.typst` 的路径为 `./themes`，即在 `readme.typst` 中指定 `theme` 路径时，填写为：

```
#set raw(theme: "themes/文件名称.tmTheme")
```

### 3.2.4. 斜体

对于没有自带斜体风格字形的字体，正文中无法显示斜体风格。

### 3.2.5. 表格行数

typst 的表格非常强大，可以画制各种各样的形式的表格，但目前缺少类似于 `y==table.footer.row-number` 的函数，pandoc 转换 markdown 的 grid table 时，无法编写 typst 的规则来为表格 footer 修改样式。

## 4. 技巧

### 4.1. 缩小 pdf 文件体积

本模板程序使用 typst 生成 pdf 文档，并没有考虑 pdf 文档生成后的体积占用问题。pdf 文件的大小，内嵌的图片、字体、冗余信息的大小决定。在生成 pdf 文档前，应减少图片大小，或者先将图片裁剪或者压缩到诺丁汉适应的大小，在不能无限制的使用高清图片，pdf 文件生成后，可使用 pdf 压缩工具，对 pdf 进行压缩。如 macos os 自带 pdf 文件体积压缩过滤器，使用 ghost script 压缩 pdf 文档等。

本模板程序发布包 `examples` 目录中的示例 pdf，均使用 ghost script 压缩，原 4.8 M 大小的 pdf 压缩到如你所见的 1.2 M 左右，压缩效果显著，命令如下：

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -dCompatibilityLevel=1.7 \  
    -dNOPAUSE -dQUIET -dPDFSETTINGS=/prepress -sOutputFile="输了文件名称"  
"原 pdf 文件"
```

## 4.2. typst 导出 png 图片时不够清晰

使用 `typst compile 原.typst 结果-{0p}-of-{t}.png` 将多页布局的 typst 文档转换成多张 png 图片。导出时 png 使用 2 倍 pdf 文件尺寸大小，若 pdf 页面尺寸设置过小时，图片效果不够清晰，如本模板程序中，以 `phone.` 开头的 pandoc typst 模板，页面大小为  $300\text{pt} \times 650\text{pt}$ ，导出的图片为  $600\text{pt} \times 1300\text{pt}$ ，当字体较小时，图片效果不佳，此时设置，让 typst 以更高分辨率生成图片，示例命令为：

```
pandoc --from markdown... --typst --template=pandoc 或 themekit 下的 pandoc 模板 |  
typst compile --ppi 200 - 输出-{0p}-of-{t}.png
```

关键参数为 `--ppi 200`，可以根据要求，设置为符合打印质量的 `--ppi 300`。因为 `--ppi` 参数选项的存在，我们不需要使用 pdf 转 png 图片工具来额外生成分辨率更高的图片。

## 4.3. 隐藏章节标题

设置 markdown front matter 参数 `section-numbering` 为 `disable` 或 `none`，可隐藏标题编号。

```
section-numbering: disable
```

## 4.4. 隐藏页码

设置 markdown front matter 参数 `page-numbering` 为 `disable` 或 `none`，页码不显示。

```
page-numbering: disable
```

## 4.5. 长页(不分页)

长页，即不分页，设置 `pagesize.height` 为 `auto`，此时不显示目录与页码，即使 `toc` 为 `true`。

```
# toc 会被忽略  
toc: true  
pagesize:
```

```
height: auto
# 或
pagesize: { height: auto }
```

## 5. 效果示例

### 5.1. 段落与换行

**备注：**本节对段落、换行进行了约定。

段落，也是“行链接”，在 markdown 中有软换行与硬换行问题，换行不分段，换行自动拼接、尾盘追加两个空格等，非常个性化，一个 markdown 文件，只支持一种约定。

但是为了简化书写，保持原有的书写习惯，减少规则的记忆与混乱，本模板建议默认你使用了以下规则来编写 markdown。要求你按以下的形式写：

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
```

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
```

中文测试中文测试中文测试中文测试中

这是另一行，但与上一行仍然属于同一行。

要想另起一个新的段落，则先输入一行空白行。

这一句仍然紧接上一句。

这又是一个新段落。中文测试中文测试中文测试中文测试中文测试中文测试中文测试中文测试中文测试

渲染为：

---

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

中文测试中文测试中文测试中文测试中 这是另一行，但与上一行仍然属于同一行。

要想另起一个新的段落，则先输入一行空白行。这一句仍然紧接上一句。

这又是一个新段落。中文测试中文测试中文测试中文测试中文测试中文测试中文测试中文测试中文测试中文测试

### Caution

在使用本模板时，根据本模板中的换行约定，不建议 `pandoc -f gfm` 或 `pandoc -f markdown` 添加 `+hard_line_breaks` 扩展。在编写长文字段落时，会生成非常多的折行断句，影响完整文字段落的排版。

确实需要新段落的，通过空一行的方式实现。便于阅读，也便于编写。普通的换行，都被视为方便编写，或者不经意间按了回车键，不影响实际效果。

## 5.2. 强调

强调，又称斜体，使用 `*星号*` 或 `_下划线_`。

强调，又称粗体，使用 `**星号**` 或 `__下划线__`。

结合使用 `**星号`和 `_下划线_**` 来强调。

删除线使用两个波浪号。 `~~划掉这个~~`

`**这是粗体文本**`

`__这是粗体文本__`

`*这是斜体文本*`

`_这是斜体文本_`

`~~删除线~~`

渲染成：

强调，又称斜体，使用 星号 或 下划线。

强调，又称粗体，使用 **星号** 或 **下划线**。

结合使用 **星号**和 下划线 来强调。

删除线使用两个波浪号。 ~~划掉这个~~。

**这是粗体文本**

**这是粗体文本**

这是斜体文本

## 这是斜体文本

删除线

---

本例中的斜体并没有显示为斜体风格，是本模使用的霞鹜文楷 GB 字体族中不包含斜体风格字形的原因，并不是模板问题或者 typst 的程序错误。具体见 [字体](#) 中的说明。

### 5.3. 有序列表

1. 第一个有序列表项
2. 另一个项
  - \* 无序列表
1. 实际数字无关紧要，只要它是一个数字
  1. 序号会重新编排
4. 另一个项。

你还可以在列表项中正确缩进段落。
5. 多空一行

除了可以正确缩进段落外，在段落前多空一行，还可以形成一个新的缩进段落，首行缩进。

除了可以正确缩进段落外，在段落前多空一行，还可以形成一个新的缩进段落，首行缩进。

7. 再来一个项
  1. 进行更改
    1. 修复错误
    2. 改进格式
      - 使标题更大
  2. 将我的提交推送到 GitHub
  3. 打开拉取请求
    - \* 描述我的更改
    - \* 提及我团队的所有成员
      - \* 征求反馈
      - \*

上面的有序列表渲染成：

---

1. 第一个有序列表项
2. 另一个项
  - 无序列表



1. 实际数字无关紧要，只要它是一个数字
2. 序号会重新编排
3. 另一个项。你还可以在列表项中正确缩进段落。
4. 多空一行

除了可以正确缩进段落外，在段落前多空一行，还可以形成一个新的缩进段落，首行缩进。

除了可以正确缩进段落外，在段落前多空一行，还可以形成一个新的缩进段落，首行缩进。

5. 再来一个项
6. 进行更改
7. 修复错误
8. 改进格式 - 使标题更大
9. 将我的提交推送到 GitHub
10. 打开拉取请求
  - 描述我的更改
  - 提及我团队的所有成员
    - 征求反馈
    -

---

pandoc 还提支 `fancy_lists`，使用 `#.` 来替代数字，pandoc 会自动计算出序号，如：

```
1. one
#. two
#. three
```

渲染为：

---

1. one
2. two
3. three

---

## 5.4. 无序列表

```
* 无序列表可以使用星号 (`*`)
- 或减号 (`-`)
+ 或加号 (`+`)

+ 通过在行开头使用 `+`、`-` 或 `*` 创建列表
+ 通过缩进 2 个空格来创建子列表:
  - 标记字符更改强制开始新列表:
    * Ac tristique libero volutpat at
    + Facilisis in pretium nisl aliquet
    - Nulla volutpat aliquam velit
+ 非常简单!

+ 无序列表也可以
+ 与序列列表混合
  1. 缩进并创建一个序号列表项
  2. 再来一个序号项
+ 另一个无充列表项
```

渲染成:

- 
- 无序列表可以使用星号 (**\***)
  - 或减号 (**-**)
  - 或加号 (**+**)
  - 通过在行开头使用 **+**、**-** 或 **\*** 创建列表
  - 通过缩进 2 个空格来创建子列表:
    - 标记字符更改强制开始新列表:
      - Ac tristique libero volutpat at
      - Facilisis in pretium nisl aliquet
      - Nulla volutpat aliquam velit
  - 非常简单!
  - 无序列表也可以
  - 与序列列表混合
    1. 缩进并创建一个序号列表项
    2. 再来一个序号项
  - 另一个无充列表项

---

## 5.5. 任务列表

使用 GitHub-Flavored Markdown 语法，使用 `pandoc` 参数

`pandoc --from markdown+task_lists` 或 `pandoc -f gfm` 解析任务列表。

- [x] 完成我的更改
- [ ] 将我的提交推送到 GitHub
- [ ] 打开拉取请求
- [x] 支持 `\@mentions`、`#refs`、`[links]()`、`**formatting**` 和 `<del>tags</del>`
- [x] 需要列表语法（支持任何无序或有序列表）
- [ ] 这是一个完整的项目
- [ ] 这是一个不完整的项目
- 这一行不是 task list 项，但也属于列表项

渲染成：

- 
- 完成我的更改
  - 将我的提交推送到 GitHub
  - 打开拉取请求
  - 支持 `@mentions`、`#refs`、`links`、`formatting` 和 `tags`
  - 需要列表语法（支持任何无序或有序列表）
  - 这是一个完整的项目
  - 这是一个不完整的项目
  - 这一行不是 task list 项，但也属于列表项
- 

## 5.6. 忽略 Markdown 格式

您可以告诉 GitHub 忽略（或通过在 Markdown 字符前使用 `<code>` 来启用 Markdown 格式的转义。

让我们将 `\*our-new-project\*` 重命名为 `\*our-old-project\*`。

---

让我们将 `*our-new-project*` 重命名为 `*our-old-project*`。

---

## 5.7. emoji

可以使用 emoji 字体，或者使用 emoji 标记



一些文字表明参考链接可以稍后跟进。

[任意不区分大小写的引用文本]: <https://www.mozilla.org>

[1]: <http://slashdot.org>

[链接文本本身]: <http://www.reddit.com>

渲染成:

---

[我是内联样式链接](#)

[我是带有标题的内联样式链接](#)

[我是引用样式链接](#)

[我是存储库文件的相对引用](#)

[您可以使用数字进行引用样式链接定义](#)

或者将其留空并使用 [链接文本本身](#)。

URL 和尖括号中的 URL 将自动转换为链接。<http://www.example.com> 或 <http://www.example.com>, 有时是 [example.com](http://www.example.com) (但不是 Github 上的)。

一些文字表明参考链接可以稍后跟进。

---

## 5.9. 图片

这是徽标:

内联样式:

```
![替代文本](./userguide/images/icon48.png "徽标标题文本 1")
```

引用样式:

```
![替代文本][徽标]
```

```
[徽标]: ./userguide/images/icon48.png "徽标标题文本 2"
```

图片下方没有题注, 并且图片紧靠着文字:

```
![Minion](./userguide/images/minion.png){width=20%}  
![Stormtroopocat](./userguide/images/stormtroopocat.png "The Stormtroopocat")  
{width=20%}
```

还可以在图片下方添加题注, 需要单独成为段落, 即前后各空一行:

```
![图片标题, 来点说明][id]
```

在文档后面的引用中定义图片位置:

```
[id]: ./userguide/images/dojocat.jpg "The Dojocat"
```


还可以指定图片大小:


```
![Minion](./userguide/images/minion.png){width=5%} ![Minion](./userguide/  
images/minion.png){width=10%}  
![Minion](./userguide/images/minion.png){width=20%}![Minion](./userguide/  
images/minion.png){width=30%}
```

渲染成:

---

这是徽标:

内联样式: 

引用样式: 



图片下方没有题注, 并且图片紧靠着文字:

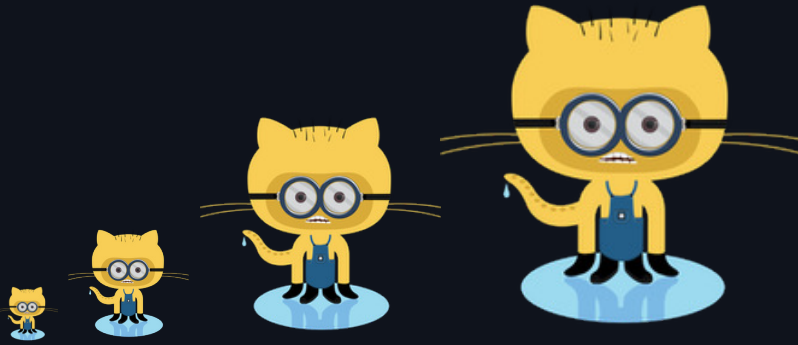
还可以在图片下方添加题注, 需要单独成为段落, 即前后各空一行:



图 1 图片标题, 来点说明

在文档后面的引用中定义图片位置:

还可以指定图片大小:



1. 要在图片下方添加注明，使用 pandoc 参数：

```
pandoc --from gfm+implicit_figures 或 pandoc --from markdown
```

2. 要控制图片大小，使用 pandoc 参数：pandoc --from gfm+attributes 或 pandoc --from markdown+link\_attributes

当图片单独成段落时，图片居中，并且会根据页面大小设置图片大小，当页面宽度小于 480 磅时，图片会平铺不拉伸。当为 A4 页面时，图片最大宽度不超过 11 厘米。内联图片不控制宽度。

## 5.10. 脚注

前文 [缩进](#) 一节使用过脚注，占用了脚注编号，因此下面 markdown 示例，编号将从 2 继续编号，即使示例中使用了 [脚注 1 链接<sup>\[^1\]</sup>](#) 对应的脚注编号自动从 2 继续编号，请注意这个细节。

脚注 1 [链接<sup>\[^1\]</sup>](#)。

脚注 2 [链接<sup>\[^2\]</sup>](#)。

~内联脚注<sup>[内联脚注的文本]</sup> 定义。

重复的脚注引用<sup>[^2]</sup>。

<sup>[^1]</sup>：脚注 **可以有标记** 和多个段落。

<sup>[^2]</sup>：脚注文本。

渲染成：

脚注 1 链接<sup>2</sup>。

脚注 2 链接<sup>3</sup>。

内联脚注<sup>4</sup> 定义。

重复的脚注引用<sup>5</sup>。

### ⚠ Warning

pandoc 存在一个 10 年还未解决的 bug [issue 1603](#), 无法正确解析重复脚注。

使用 pandoc 要正确解析带有 [内联引用](#) 的脚注语法时,, 需要使用以下的参数。

```
pandoc --from markdown
# 或者
pandoc --from markdown_mmd+inline_notes
```

即内联引用形式的脚注, 只有 pandoc 的 `--from markdown` 才能支持, 或者开启 `inline_notes` 选项的 `markdown_mmd`。

## 5.11. 定义列表

术语

: 术语名称独立一行, 第二行以冒号定格 ( `:` `) 开头, 紧接是术语定义的描述

又一个术语

: 还是以冒号定格 ( `:` `) 开头, 紧接还是术语定义的描述。**\*定义列表\***经常用在文档, 对专业术语、业务术语、名词进行统一的定义。

术语条目换行

: 与 markdown 的换行解释有关, 如果是 ``hard_linkbreak``, 即条目间需要空出一行。

这不是一个定义

: 定义名称没有与上个条目空行分隔, 无法成为新的条目。

术语定义段落

: 术语内容作为整体段落, 向右缩进。

---

<sup>2</sup>脚注 可以有标记 和多个段落。

<sup>3</sup>脚注文本。

<sup>4</sup>内联脚注的文本

<sup>5</sup>脚注文本。



渲染成:

---

### 术语

术语名称独立一行，第二行以冒号定格（:）开头，紧接是术语定义的描述

### 又一个术语

还是以冒号定格（:）开头，紧接还是术语定义的描述。[定义列表](#)经常用在文档，对专业术语、业务术语、名词进行统一的定义。

### 术语条目换行

与 markdown 的换行解释有关，如果是 `hard_linkbreak`，即条目间需要空出一行。这不是一个定义

定义名称没有与上个条目空行分隔，无法成为新的条目。

### 术语定义段落

术语内容作为整体段落，向右缩进。

---

需要使用 pandoc 参数 `--from markdown` 或 `--from gfm+definition_lists` 解析定义列表。

## 5.12. 数学公式

pandoc 中 markdown 编写数学公式时，使用 `$$`、`\( \)`、或者 `\[ \]` 这几类符号，把公式包裹起来。也有 GitHub flavored Markdown 特有格式。

单独行显示

```
: $$
```

```
e=mc^2
```

```
$$
```

内联

: 这是一个小兔子，紧接一个公式，`$e=mc^2$`。紧接着后面还可以有文字。

使用 pandoc 参数 `pandoc --from gfm` 或 `pandoc --from markdown+tex_math_dollars`，渲染成:

---

### 单独行显示

$$e = mc^2$$

### 内联

这是一个小兔子，紧接一个公式， $e = mc^2$ 。紧接着后面还可以有文字。

若要使用 `pandoc --from gfm`，还可支持，

gfm 特有的：

```
$`e=mc^2`$  
``` math  
e=mc^2  
```
```

还可以使用 `pandoc --from markdown+tex_math_single_backslash`，解析以下格式：

gfm 不能解释的：

```
\( e=mc^2 \  
\[ e=mc^2 \]
```

### 5.13. 代码和语法突出显示

代码高亮显示的颜色主题 具体见 [代码高亮颜色主题](#)，每个主题在对应的 `pandoc tystpt template` 模板文件中设定。

内联 `代码` 周围有 `反引号`。

内联 代码 周围有 反引号。

代码块使用一对 三反引号 ```

css 示例代码：

```
```css  
@font-face {  
  font-family: Chunkfive; src: url('Chunkfive.otf');  
}  
  
body, .usertext {  
  color: #F0F0F0; background: #600;  
  font-family: Chunkfive, sans;  
}  
  
@import url(print.css);  
@media print {  
  a[href^=http]::after {
```

```
    content: attr(href)
  }
}
...
```

css 代码高亮效果:

```
@font-face {
  font-family: Chunkfive; src: url('Chunkfive.otf');
}

body, .usertext {
  color: #F0F0F0; background: #600;
  font-family: Chunkfive, sans;
}

@import url(print.css);
@media print {
  a[href^=http]::after {
    content: attr(href)
  }
}
```

---

javascript 示例代码:

```
```javascript
function $initHighlight(block, cls) {
  try {
    if (cls.search(/\bno\-highlight\b/) != -1)
      return process(block, true, 0x0F) +
        ` class="${cls}"`;
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]) === undefined)
      console.log('undefined');
  }
}
```
```

```
export $initHighlight;
```
```

javascript 代码高亮效果:

```
function $initHighlight(block, cls) {
  try {
    if (cls.search(/\bno\-\highlight\b/) != -1)
      return process(block, true, 0x0F) +
        ` class="${cls}"`;
  } catch (e) {
    /* handle exception */
  }
  for (var i = 0 / 2; i < classes.length; i++) {
    if (checkCondition(classes[i]) === undefined)
      console.log('undefined');
  }
}

export $initHighlight;
```

#### ⚠ Warning

`typst #highlight()` 函数控制行高美观效果不如 `#box()`，因此模板中使用 `typst #box()` 实现内联代码。若使用内联代码格式，出现超过行尾需要折行时，无法分拆，可能会导致上一行过多的留白，或者超过一行，此时可调整语句的描述方式，或者将 `` 的语句变短，目前暂未找到更好的实现方法。

## 5.14. 表格

冒号可用于对齐列。

表格	很	酷	
-----	:-----:	-----:	
列 3 是	右对齐	\$1600	
列 2 是	居中	\$12	
斑马条纹	整齐	\$1	

每个标题单元格之间必须至少有 3 个破折号 (``)。

外部管道 (`|`) 是可选的，您不需要让原始 Markdown 排列得很漂亮。您还可以使用内联 Markdown。

表格示例，外部管道符 (|) 可省略：

```
Markdown | 少也 | 漂亮
--- | --- | ---
*仍然* | `呈现` | **很好**
1 | 2 | 3

| 第一个标题 | 第二个标题 |
| ----- | ----- |
| 内容单元格 | 内容单元格 |
| 内容单元格 | 内容单元格 |
```

表格示例，单元格内容可以标记样式：

```
| 命令 | 描述 |
| --- | --- |
| git status | 列出所有新的或修改过的文件 |
| git diff | 显示尚未暂存的文件差异 |

| 命令 | 说明 |
| --- | --- |
| `git status` | 列出所有 *新的或修改过的* 文件 |
| `git diff` | 显示**尚未**暂存的文件差异 |
```

表格示例，表格各列可自由对齐：

```
| 左对齐 | 居中对齐 | 右对齐 |
| :--- | :---: | ---: |
| git status | git status | git status |
| git diff | git diff | git diff |
```

表格示例，单元格中的特殊符号：

```
| 名称 | 字符 |
| --- | --- |
| 反引号 | ` |
| 管道符 | \| |
```

渲染成：

---

冒号可用于对齐列。

表格	很	酷
列 3 是	右对齐	\$1600
列 2 是	居中	\$12
斑马条纹	整齐	\$1

每个标题单元格之间必须至少有 3 个破折号 ( - )。外部管道 ( | ) 是可选的, 您不需要让 原始 Markdown 排列得很漂亮。您还可以使用内联 Markdown。

表格示例, 外部管道符 ( | ) 可省略:

Markdown	少也	漂亮
<u>仍然</u>	呈现	很好
1	2	3
第一个标题	第二个标题	
内容单元格	内容单元格	
内容单元格	内容单元格	

表格示例, 单元格内容可以标记样式:

命令	描述
git status	列出所有新的或修改过的文件
git diff	显示尚未暂存的文件差异
命令	说明
git status	列出所有 <u>新的或修改过的</u> 文件
git diff	显示 <u>尚未</u> 暂存的文件差异

表格示例, 表格各列可自由对齐:

左对齐	居中对齐	右对齐
git status	git status	git status
git diff	git diff	git diff

表格示例, 单元格中的特殊符号:

名称	字符
反引号	`
管道符	

以上的表格语法，称为 `pipe tables`。pandoc markdown 还支持 `simple_tables`、`grid_tables` 两种风格，同时还支持 [单元格多行](#)、[表格标题](#)。使用 `pandoc --from markdown+simple_tables+grid_tables+multiline_tables+table_captions` 开启扩展。

### 5.14.1. 表格标题 table caption

`table_captions` 可以用来为表格添加题注：

```
列 1 | 列 2
-----|-----
first | Second

: 在表格后面定义标题

table: 在表格前面定义表格标题

列 1 | 列 2
-----|-----
first | Second
```

以 `Table:`、`table:` 或 `:` 开头，在表格的前面或后面均可，记得要与表格之间间隔一行空白行。上面表格渲染为：

表 9 在表格后面定义标题

列 1	列 2
first	Second

表 10 在表格前面定义表格标题

列 1	列 2
first	Second

### 5.14.2. simple tables

`simple_tables`，看起来简单，但实际比 pipe tables 编辑起来要费事。它对格式有严格要求

```
      右      左      居中      Default
-----
      12      12      12      12
     123     123     123     123
        1        1        1        1
```

Table: 展示 simple table 语法.

通过 `-----` 来划分标题与单元格。标题与 `-----` 线相对位置，来设置单元格内容居左、居中、居右。为了接近实际布局，`-----`、标题、单元格内容最好按照需要对齐。上面的表格渲染为：

表 11 展示 simple table 语法.

右	左	居中	Default
12	12	12	12
123	123	123	123
1	1	1	1

### 5.14.3. 多行表格

`multiline_tables` 的语法与 `simple_tables` 接近。

```
-----
      居中      Default      靠右 靠左
      标题      Aligned      对齐 对齐
-----
      First      row      12.0 Example of a row that
                        spans multiple lines.
      Second      row      5.0 Here's another one. Note
                        the blank line between
                        rows.
-----
```

Table: Here's the caption. It, too, may span multiple lines.

`multiline_tables` 的格式要求：

1. 表格上下各一条 `-----` 线，并且表后要有一空白行



2. 标题与 `-----` 分割线的对齐与 `simple_tables` 相同
3. 每行内容之间，使用空白行分隔

上面的跨行表格渲染为：

表 12 Here's the caption. It, too, may span multiple lines.

居中 标题	Default Aligned	靠右 对齐	靠左 对齐
First	row	12.0	Example of a row that spans multiple lines.
Second	row	5.0	Here's another one. Note the blank line between rows.

#### 5.14.4. grid tables

`grid_tables` 则是结构最复杂，但功能也最多的表格：

: grid table 示例

```
+-----+-----+-----+
| Fruit      | Price      | Advantages  |
+=====+=====+=====+
| Bananas    | $1.34      | - built-in wrapper |
|            |            | - bright color  |
+-----+-----+-----+
| Oranges    | $2.10      | - cures scurvy  |
|            |            | - tasty         |
+-----+-----+-----+
```

单元格可跨多行或多列：

```
+-----+-----+
| Property          | Earth  |
+=====+=====+
|                   | min    | -89.2 °C |
| Temperature +-----+-----+
| 1961-1990  | mean  | 14 °C   |
|                   +-----+-----+
|                   | max   | 56.7 °C |
```

```
+-----+-----+-----+
```

多行标题的表格:

```
+-----+-----+-----+
| Location          | Temperature 1961-1990 |
|                   | in degree Celsius     |
|                   +-----+-----+-----+
|                   | min | mean | max |
+=====+=====+=====+=====+
| Antarctica        | -89.2 | N/A  | 19.8 |
+-----+-----+-----+-----+
| Earth             | -89.2 | 14   | 56.7 |
+-----+-----+-----+-----+
```

与 pipe tables 类似的对齐语法, 使用 `:` 设置标题分割线:

```
+-----+-----+-----+
| Right          | Left          | Centered          |
+=====+:=====+:=====+:+
| Bananas        | $1.34         | built-in wrapper  |
+-----+-----+-----+-----+
```

还可以使用 `=====` 设置表格脚部:

```
+-----+-----+
| Fruit          | Price         |
+=====+=====+
| Bananas        | $1.34         |
+-----+-----+
| Oranges        | $2.10         |
+=====+=====+
| Sum            | $3.44         |
+=====+=====+
```

渲染成:

表 13 grid table 示例

Fruit	Price	Advantages
Bananas	\$1.34	<ul style="list-style-type: none"><li>• built-in wrapper</li><li>• bright color</li></ul>
Oranges	\$2.10	<ul style="list-style-type: none"><li>• cures scurvy</li><li>• tasty</li></ul>

单元格可跨多行或多列:

Property		Earth
Temperature 1961-1990	min	-89.2 °C
	mean	14 °C
	max	56.7 °C

多行标题的表格:

Location	Temperature 1961-1990 in degree Celsius		
	min	mean	max
Antarctica	-89.2	N/A	19.8
Earth	-89.2	14	56.7

与 pipe tables 类似的对齐语法, 使用 `:` 设置标题分割线:

Right	Left	Centered
Bananas	\$1.34	built-in wrapper

还可以使用 `=====` 设置表格脚部:

Fruit	Price
Bananas	\$1.34
Oranges	\$2.10
Sum	\$3.44

---

`typst 0.12` 目前没有获得表格末行的方法, 则当表的 footer 时, 无法通过编写逻辑规则函数来为表的 footer 添加样式。

## 5.15. 表格宽度与列宽

`pandoc` 的 `markdown+grid_tables` 或者 `markdown+multiline_tables` 支持指定表格的宽度, 与列宽。当表格内容较少, 页面布面不自然时, 可改用 `grid tables` 或 `multiline tables` 来调整。

分别可干预的有 表格宽度与 列宽度。

表格宽度的影响因素有二：

1. `pandoc --columns` 参数值，默认为 `72`
2. 定义 `grid` 表格时，顶加框所左上角 `+` 号到右上角边 `+` 号之间的字符数，即顶边框的总字符数则为表格的相对宽度，相对于正文宽度的比例，其相对宽度  $\approx$  总长度  $\div$  72，当相对宽度  $> 1$  时，等于 1。实际宽度  $\approx$  相对长度  $\times$  正文宽度。

表格列宽也是相对的，列表标题分割线 `====` 或 `----`（包括后缀空格）的字符数，占全表相对宽度的比例，即为列相对宽度。

以 `grid_tables` 为例：

```
: 2 列 72 字符
```

```
+-----+-----+
| Fruit          Price          | Advantages          |
+=====+=====+:=====+
| Oranges        $2.10          | - cures scurvy     |
|                |                | - tasty             |
+-----+-----+
```

```
: 3 列 72 字符
```

```
+-----+-----+-----+
| Fruit        | Price          | Advantages          |
+=====+=====+:=====+
| Oranges      | $2.10          | - cures scurvy     |
|              |                | - tasty             |
+-----+-----+-----+
```

```
: 2 列 37 字符
```

```
+-----+-----+
| Fruit        | Price          |
+=====+=====+
| Oranges      | $2.10          |
|              |                |
+-----+-----+
```

渲染成：

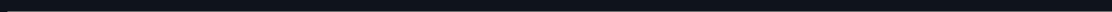


表 18 2 列 72 字符数

Fruit Price	Advantages
Oranges \$2.10	<ul style="list-style-type: none"> <li>• cures scurvy</li> <li>• tasty</li> </ul>

表 19 3 列 72 字符数

Fruit	Price	Advantages
Oranges	\$2.10	<ul style="list-style-type: none"> <li>• cures scurvy</li> <li>• tasty</li> </ul>

表 20 2 列 37 字符数

Fruit	Price
Oranges	\$2.10

以 `multiline_tables` 为例:

: 4 列, 72 字符数

```

-----
Centered Default      Right Left
Header  Aligned      Aligned Aligned
-----
First  row              12.0 Example of a row that
                             spans multiple lines.
-----

```

: 3 列, 34 字符数

```

-----
Centered Default      Right
Header  Aligned      Aligned
-----
First  row              12.0
-----

```

: 2 列 72 字符数

```

-----
Centered  Default      Right      Left
-----

```

```
Header    Aligned  Aligned      Aligned
-----
First    row      12.0        Example of a row that
-----
                    spans multiple lines.
```

渲染成:

表 21 4 列, 72 字符数

Centered Header	Default Aligned	Right Aligned	Left Aligned
First	row	12.0	Example of a row that spans multiple lines.

表 22 3 列, 34 字符数

Centered Header	Default Aligned	Right Aligned
First	row	12.0

表 23 2 列 72 字符数

Centered Default Right Header Aligned Aligned	Left Aligned
First row 12.0	Example of a row that spans multiple lines.

## 5.16. 引用

- > 文本区块引用在电子邮件中非常方便，可以模拟回复文本。
- > 此行是同一引文的一部分。

引文分隔符。

- > 这是一行非常长的行，换行时仍会正确引用。哦，让我们继续写，以确保它足够长，实际上可以为每个人换行。哦，你可以\*将\* **Markdown** 放入区块引用中。

- > 块引用也可以嵌套...

```
>> ...通过使用紧挨着彼此的附加大于号...
> > > ...或者在箭头之间留有空格。
```

渲染成:

---

文本区块引用在电子邮件中非常方便，可以模拟回复文本。此行是同一引文的一部分。

引文分隔符。

这是一行非常长的行，换行时仍会正确引用。哦，让我们继续写，以确保它足够长，实际上可以为每个人换行。哦，你可以将 `Markdown` 放入区块引用中。

块引用也可以嵌套...

```
    ...通过使用紧挨着彼此的附加大于号...
        ...或者在箭头之间留有空格。
```

---

当引用中的文本内容超过一行时或者多段落时，你无法看到第一段第一行缩进。目前 `tyspt` 暂不支持。具体见 [缩进](#) 中的说明。

## 5.17. alerts

`alerts` 是 GitHub-Flavored markdown 中专用的样式。本模板做了兼容性处理，支持除了 `gfm` 本身所规定的 `note`、`tip`、`important`、`caution`、`warning` 外，还支持 `idea`、`error`、`success`、`goal`、`notification`，大小写均可

```
> [!NOTE]
> 这是一个 note

> [!TIP]
> 这是一个提示

> [!IMPORTANT]
```

```
> 这很重要

> [!CAUTION]
> 要小心这里所描述的问题

> [!WARNING]
> 警告
>

>[!Idea]
> 这是一个想法
>
>

> [!Error]
> 这里错误了
>

> [!Success]
> 成功的时候

> [!Goal]
> 这是目标
>

> [!Notification]
> 这里有一个通知
>
```

渲染为:

---

 Note

这是一个 note

 Tip

这是一个提示

 Important

这很重要



 Caution

要小心这里所描述的问题

 Warning

警告

 Idea

这是一个想法

 Error

这里错误了

 Success

成功的时候

 Goal

这是目标

 Notification

这里有一个通知

---

若 alerts 样式过分，那么可以使用引用文本块来替代：

```
> Note:  
> Block quotes can be used to highlight important ideas.
```

渲染为：

---

**Note:** Block quotes can be used to highlight important ideas.

---

当 alerts 中的文本内容超过一行时或者多段落时，你无法看到第一段第一行缩进。目前 `typst` 暂不支持。具体见 [缩进](#) 中的说明。

## 5.18. 水平线

三个符号或更多

三个连字符 ``---``

---

三个星号 ``***``

\*\*\*

三个下划线 ``___``

\_\_\_

三个符号或更多

三个连字符 `---`

---

三个星号 `***`

---

三个下划线 `___`

---

## 5.19. 内联 HTML

```
<dl>
```

```
<dt>定义列表</dt>
```

```
<dd>人们有时会使用。</dd>
```

```
<dt>HTML 中的 Markdown</dt>
```

```
<dd>效果不是很好。使用 HTML <em> 标签</em>。</dd>
```

```
</dl>
```

### Caution

只有当使用 `pandoc -f markdown` 时，才可以将 markdown 文件转换为 `typst` 或通过 `--pdf-engine typst` 参数转换为 pdf 格式文档时，才能显示 **内联 html** 的文本内容，并且退为仅显示文本。

## 5.20. 标题

在标题演示会影响 pdf 目录结构，放在文档最后。

```
# (演示需要，请忽略) h1 标题
```

```
## h2 标题
```

```
### h3 标题
```

```
#### h4 标题
```

```
##### h5 标题
```

```
##### h6 标题
```

或者，对于 H1 和 H2，使用下划线样式：

```
(演示需要，请忽略) Alt-H1
```

```
=====
```

```
Alt-H2
```

```
-----
```

渲染成：

---

## 6. (演示需要，请忽略) h1 标题

### 6.1. h2 标题

#### 6.1.1. h3 标题

##### 6.1.1.1. h4 标题

###### 6.1.1.1.1. h5 标题

###### 6.1.1.1.1.1. h6 标题

或者，对于 H1 和 H2，使用下划线样式：

## 7. (演示需要，请忽略) Alt-H1

### 7.1. Alt-H2

---