

PanBook 使用手册

用 Markdown 和 Pandoc 写作

An He

PanBook

2022 年 7 月 23 日

Copyright © 2022 An He

PUBLISHED BY **PANBOOK**

[HTTPS://GITHUB.COM/ANNPROG/PANBOOK](https://github.com/ANNPROG/PANBOOK)

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

最后编译日期, 2022 年 7 月 23 日 20:46:00

前言

\LaTeX 可以用来排版格式精美的书籍，论文，幻灯片以及简历，但是学习成本较高。并且也不太适合直接用来写作，太多的标记使人不能专注于内容本身。另外，如果同一份文本有在不同场合使用的需求或者需要不同的格式，比如电子书同时需要 PDF 和 EPUB 格式，一份简历需要不同的风格，幻灯片需要在不同场合使用等，显然用 \LaTeX 来写作不是个好注意，最终还得维护多份源码，或者想办法转换格式，这是相当不方便的。如果能用简单易学的 Markdown 来写作，然后通过程序转换成各种格式或者各种风格，就比较理想了。幸运的是，已经有工具很好的实现了支持这一功能，那就是 Pandoc。

[Pandoc](#) 是由 John MacFarlane 教授开发的文档转换工具，实现了数十种文档格式之间的转换。Pandoc 还扩展了 Markdown 语法，比如标题表格等的 ID 属性，脚注，交叉引用，DIV 语法等，并且可以直接嵌入 \LaTeX 代码，这样在 Markdown 中就可以实现输入数学公式。Pandoc 还支持自定义模板和 Lua 脚本修改转换结果，有很好的自由度和扩展能力，能够适配各种复杂的 \LaTeX 模板。

PanBook 要做的事情，就是利用 Lua Filter 来适配各种书

籍, 论文, 幻灯片及简历模板, 目标是使用 Pandoc's Markdown 作为写作语言, 实现 **一次编写, 多次生成**。

目录

封面	ii
前言	iii
目录	ix
表格列表	xi
插图列表	xiii
第一章 PanBook 手册	1
1.1 快速开始	1
1.1.1 指定风格 (style)	2
1.1.2 调试模式	4
1.1.3 注意事项	4
1.2 书籍元数据	5
1.3 写作工具	6
第二章 Pandoc Markdown 语法	9
2.1 段落	9

2.2	标题	10
2.3	引用	12
2.4	代码	14
2.4.1	缩进式代码块	14
2.5	行区块	16
2.6	列表	18
2.6.1	无序列表	18
2.6.2	区块内容作为列表项	19
2.6.3	有序列表	21
2.6.4	定义列表	23
2.6.5	编号范例列表	25
2.6.6	紧凑与宽松列表	27
2.6.7	结束一个列表	28
2.7	分隔线	29
2.8	表格	29
2.8.1	简单表格	30
2.8.2	多行表格	32
2.8.3	格框表格	34
2.8.4	管线表格	35
2.9	元数据区块	37
2.9.1	Pandoc 标题区块	37
2.9.2	元数据区块	39
2.10	字符转义	42
2.11	行内格式	44
2.11.1	强调	44
2.11.2	删除线	45
2.11.3	上标与下标	45

2.11.4	字面文字	46
2.11.5	小型大写字母	47
2.12	数学公式	47
2.13	Raw HTML	49
2.13.1	通用 raw attribute	51
2.14	L ^A T _E X 宏指令	52
2.15	链接	52
2.15.1	自动链接	52
2.15.2	行内链接	53
2.15.3	参考链接	53
2.15.4	内部链接	55
2.16	图片	56
2.17	Divs 和 Spans 语法	59
2.18	脚注	61
2.19	参考文献	62
2.20	交叉引用	69
2.20.1	图片	69
2.20.2	公式	74
2.20.3	表格	75
2.20.4	标题	75
2.20.5	代码块	75
2.20.6	引用	76
2.20.7	自定义交叉引用配置	77

第三章 **PanBook** 扩展 **83**

3.1	abstract	84
3.2	column	84

3.3	copyright	87
3.4	coverletter	87
3.5	device	88
3.6	fontawesome5	89
3.7	grade	89
3.8	header	90
3.9	lineblock	91
3.10	listings	91
3.11	longtable	93
3.12	tabu	93
3.13	theorem	94
3.14	wrap	98
3.15	zh_en	101
3.16	plot	103
3.16.1	安装	103
3.16.2	用法	104
3.16.3	graphviz	104
3.16.4	dita	105
3.16.5	goseq 时序图	106
3.16.6	asciitotikz	107
3.16.7	gnuplot	108
3.16.8	Asymptote	110
3.16.9	乐谱 (abc notation)	112
3.16.10	tikz	113
3.16.11	子图	114
3.17	ipa	116

第四章 开发	117
4.1 Pandoc 开发基础	118
4.2 模块开发	118
4.3 风格开发	119
4.4 扩展	119
参考文献	121

表格列表

1.1	各模块默认风格	5
1.2	推荐插件	6
2.1	简单表格演示	30
2.2	这里是标题标题也可以是多行	32
2.3	无表头的多行表格	33
2.4	格框表格示例	34
2.5	管线表格示例	35
2.6	自动换行示例	37
2.7	Pandoc 支持的引用形式	63
4.1	PanBook 名词定义	118

插图列表

2.1	Caption of figure	70
2.2	Cool figure!	73
3.1	Dot 示例	105
3.2	Ditaa 示例	106
3.3	Goseq Demo	107
3.4	A2s Demo	108
3.5	gnuplot 示例	109
3.6	gnuplot 绘制数据图	110
3.7	Asymptote 示例	112
3.8	ABC notation	113
3.9	TIKZ 示例	114
3.10	子图示例	116
4.1	PanBook 流程图	117

第一章 PanBook 手册

PanBook 基于 Pandoc 的 lua filter 功能，适配各种书籍，论文，幻灯片及简历的 LaTeX 或 EPUB 模板。目标是使用 Pandoc's Markdown 作为写作语言，实现**一次编写，多次生成**。

1.1 快速开始

以 Windows 10 为例，演示如何使用。首先需要安装依赖软件。

- 安装 [msys2](#)（Linux 及 OS X 请忽略此步骤）
- 安装 [texlive](#) 2018 或以上版本
- 安装 [Pandoc](#) 2.7.3 或以上版本
- 下载 [pandoc-crossref](#) 对应版本安装到 path 目录下（建议和 Pandoc 放同一目录）

另外，本项目需要使用 4.x 版本以上的 bash, OS X 用户可能需要升级 bash 版本，参见 [stackoverflow](#)

然后下载 PanBook。打开终端 (msys2)，假设工作目录为 /d/dev，克隆代码并设置环境变量：

```
# 首先安装 git, msys2 包管理工具为 pacman
$ pacman -Ss git          # 查询软件包
$ pacman -S git           # 安装软件包
$ cd /d/dev
$ git clone https://github.com/annProg/PanBook
# 将 PanBook, TeXLive 及 Pandoc 加入环境变量
$ tail -n 1 ~/.bashrc
export PATH=$PATH
↪ :/d/texlive/2018/bin/win32:/d/dev/PanBook:/c/Use
```

完成环境变量设置之后，在任意空目录下执行 `panbook <command>`，会自动初始化写作环境，生成示例源码。然后在 `src` 目录下开始写作。`<command>` 可以是 `book`，`thesis`，`slide`，`cv` 和 `art` (article)。详细帮助信息执行 `panbook -h` 查看。目录规范见代码 1.1。

章节内容如果很多,也可以用目录来组织章节,见代码 1.2。也需要能按顺序列出。举例来说，如果 `chapter` 或者小节数量超出 10，那么编号要从 01 开始，不能从 1 开始，否则 10 将错误的排在 1 后面。

1.1.1 指定风格 (style)

风格 (style)，也可以理解为模板（为了和 `pandoc` 模板区分），是预定义的文档样式。通过 `--style` 参数指定，缺失时，会使用默认风格，列表如表 1.1。

代码 1.1: 目录规范

```
.
|-- extensions          # 自定义扩展
|-- fonts              # 自定义字体
|-- src                # Markdown 源码目录
|   |-- images         # 插图目录
|   |-- metadata.yaml  # 书籍元数据文件
|   |-- frontmatter.md # 前言文件
|   |-- backmatter.md  # 后记文件
|   |-- 100-chapter1.md # 正文，命名须保证能按正确
↪   章节顺序列出
|   `-- 200-chapter2.md # 正文也可以用子目录组织
|-- styles              # 自定义风格
|-- templates           # 自定义模板
`-- build               # 电子书构建目录
```

通过命令 `panbook <module> -l` 查看模块支持的风格列表，以 `book` 为例：

```
$ panbook book -l
ctexbook
elegantbook
epub
html5
# 指定风格 elegantbook 来编译
$ panbook book --style=elegantbook
```

```
.  
|-- backmatter.md  
|-- bibliography.bib  
|-- chapter-01  
|   |-- 01-title.md  
|-- chapter-02  
|   |-- 01-title.md  
|-- chapter-03  
|   |-- 01-title.md  
|-- frontmatter.md  
|-- images  
|-- metadata.yaml
```

1.1.2 调试模式

加 `-d` 选项, 会输出详细的 `latexmk` 编译过程, 加 `--trace` 选项, 可以输出更多的调试信息。加 `--part=` 源文件名只编译某个 Markdown 源文件, 方便调试。通过配置 VS Code Tasks, 使用 `--part` 选项, 可实现使用快捷键 (`ctrl + shift + b`) 编译。

1.1.3 注意事项

- Markdown 源码文件需要使用 UTF-8 编码
- Pandoc 扩展的 Markdown 语法要求在标题前留出一个空行, 因此按章节拆分的多个 Markdown 文件, 开头需要空一行, 否则 `pandoc` 不能正确识别标题

表 1.1: 各模块默认风格

book	thesis	article	slide	cv
ctexbook	thesis	ctexart	metropolis	moderncv
				classic

- 请勿将正文文件命名为 *frontmatter.md* 或者 *backmatter.md*，这 2 个文件有特殊用途

1.2 书籍元数据

在 `src/metadata.yaml` 中使用 [Yaml 语言](#) 定义书籍的数据及可用的模板变量，示例见代码 1.3。

代码 1.3: Metadata

```
---
title: PanBook 使用手册
subtitle: 用 Pandoc 和 Markdown 写作
author: # 作者 (数组)
  - An He
homepage: https://github.com/annProg/PanBook
header-includes:
  - \usepackage{cleveref}
  - \usepackage{float}
...
```

② 模板变量

查看模板文件，可以获取模板支持的所有变量（形如 `var`）。也可以通过修改模板来添加自定义的变量。

更多信息请参考章节 2.9.2。

1.3 写作工具

推荐使用 [Visual Studio Code](#)。一些有用的插件见表 1.2。

表 1.2: 推荐插件

插件	功能
Markdown Preview	Markdown 实时预览
L ^A T _E X language support	L ^A T _E X 语言高亮
All Autocomplete	自动补全（支持单词补全）
Pangu-Markdown	中英文自动加空格

使用代码 1.4 配置将终端改为 `msys2` 的 `bash`。

使用用户代码片段，可以定义一些快捷输入，例如以下代码定义了快速输入 `columns` 环境，代码块以及 `Div` 环境。`$` 符号表示需要用户输入的内容，可使用 `TAB` 键跳转：

```
"Columns": {  
  "prefix": "col",  
  "body": [  
    "
```

代码 1.4: VS Code 使用 msys2

```
{  
  "terminal.integrated.shell.windows":  
    ↪ "D:\\msys64\\usr\\bin\\bash.exe",  
  "terminal.integrated.shellArgs.windows":  
    ↪ ["-l"],  
  "terminal.integrated.env.windows": {  
    "CHERE_INVOKING": "1",  
    "MSYSTEM": "MINGW64",  
  },  
  "git.postCommitCommand": "push",  
  "git.path": "D:\\msys64\\git-vscode.bat",  
  "editor.insertSpaces": false,  
  "editor.detectIndentation": false,  
}
```

```
    "::::: {.columns}",  
    ":::: {.column}",  
    "$1",  
    "::::",  
    ":::: {.column}",  
    "$2",  
    "::::",  
    ":::::"  
  ],  
  "description": "Columns"  
},  
"Code": {
```

```

    "prefix": "cod",
    "body": [
        "```${#lst:$1 . $2 caption=\"${3}\"}",
        "$4",
        "```"
    ],
    "description": "Code"
},
"Environment": {
    "prefix": "env",
    "body": [
        " ::: {.$1}",
        "$2",
        " :::"
    ],
    "description": "fenced Div environment"
}

```

VS Code 使用技巧:

- 通过文件 -> 首选项 -> 用户代码片段, 选择 Markdown , 然后编辑 markdown.json 来定义代码片段
- snippetSuggestions 设置为 top, 可以优先显示代码片段
- tabCompletion 设置为 on 使用 TAB 自动补全
- pangu.auto_format_on_save 设置为 on 在保存时自动给中英文加空格

第二章 Pandoc Markdown 语法

Pandoc 的目标与原始 Markdown 的最初目标有着方向性的不同。在 Markdown 原本的设计中，HTML 是其主要输出对象；然而 Pandoc 则是针对多种输出格式而设计。因此，虽然 Pandoc 同样也允许直接嵌入 HTML 标签，但并不鼓励这样的作法，取而代之的是 Pandoc 提供了许多非 HTML 的方式，来让使用者输入像是定义列表、表格、数学公式以及脚注等诸如此类的重要文件元素。

Pandoc Markdown 语法介绍可以在 [Pandoc 主页](#) 找到。以下翻译部分摘自 tzengyuxiao 的翻译^[1]，在此向译者表示感谢。

2.1 段落

一个段落指的是一行以上的文字，跟在一行以上的空白行之后。换行字元会被当作是空白处理，因此你可以依自己喜好排列段落文字。如果你需要强制换行，在行尾放上两个

以上的空白字元即可。

i Extension escaped_line_breaks

一个反斜线后跟着一个换行字元，同样也有强制换行的效果。这也是在表格单元格中添加换行的唯一形式。

2.2 标题

有两种不同形式的标题语法，Setext 以及 Atx。Setext 风格的标题是由一行文字底下接着一行 = 符号（用于一阶标题）或-符号（用于二阶标题）所构成；Atx 风格的标题是由一到六个 # 符号以及一行文字所组成，你可以在文字后面加上任意数量的 # 符号。由行首起算的 # 符号数量决定了标题的阶层，如代码代码 2.1 所示。

i Extension blank_before_header

原始 markdown 语法在标题之前并不需要预留空白行。Pandoc 则需要（除非标题位于文件最开始的地方）。这是因为以 # 符号开头的情况在一般文字段落中相当常见，这会导致非预期的标题。例如代码 2.2。

这也是章节 1.1.3 所述注意事项的原因。

代码 2.1: Markdown 标题

```
Setext A level-one header
```

```
=====
```

```
Setext A level-two header
```

```
-----
```

```
# Atx level-one
```

```
## Atx level-two
```

```
### Atx level-three
```

代码 2.2: 标题前置空行

```
I like several of their flavors of ice cream:
```

```
#22, for example, and #5.
```

Extension header_attributes

在标题文字所在行的行尾，可以使用 `{#identifier .class .class key=value key=value}` 语法为标题加上属性：

虽然这个语法也包含加入类别 (class) 以及键／值形式的属性 (attribute)，但目前只有标识符 (identifier/ID) 在输出时有实际作用（且只在部分格式的输出，包括：HTML, \LaTeX ,

ConTeXt, Textile, AsciiDoc)。举例来说，以下是将标题加上 foo 标识符的几种方法：

```
# My header {#foo}

## My header ##      {#foo}

My other header      {#foo}
-----
```

此语法与 PHP Markdown Extra 相容。

具有 .unnumbered 类别的标题将不会被编号,即使 -numbered 选项是开启的。单一连字符号 (-) 等同于 .unnumbered，且更适用于非英文文件中。因此以下两行是等价的：

```
# My header {-}
# My header {.unnumbered}
```

2.3 引用

This is a block quote. This paragraph has two lines.

1. This is a list inside a block quote.
2. Second item.

Markdown 使用 email 的习惯来建立引用区块。一个引用区块可以由一或多个段落或其他区块元素（如列表或标题）组成，并且其行首均是由一个 > 符号加上一个空白作为

开头。(> 符号不一定要位在该行最左边，但也不能缩进超过三个空白)。

```
> This is a block quote. This
> paragraph has two lines.
>
> 1. This is a list inside a block quote.
> 2. Second item.
```

有一个「偷懒」的形式：你只需要在引用区块的第一行行首输入> 即可，后面的行首可以省略符号。

```
> This is a block quote. This
paragraph has two lines.
> 1. This is a list inside a block quote.
2. Second item.
```

由于区块引用可包含其他区块元素，而区块引用本身也是区块元素，所以，引用是可以嵌套入其他引用的。

```
> This is a block quote.
>
>> A block quote within a block quote.
```

Extension blank_before_blockquote

原始 markdown 语法在区块引用之前并不需要预留空白行。Pandoc 则需要（除非区块引用位于文件最开始

的地方)。这是因为以 `>` 符号开头的情况在一般文字段落中相当常见（也许由于断行所致），这会导致非预期的格式。因此，除非是指定为 `markdown_strict` 格式，否则需要预留空白。

2.4 代码

2.4.1 缩进式代码块

由四个空格或一个 `tab` 缩进的文本取做代码块，区块中的特殊字符、空格和换行都会被保留，而缩进的空格和 `tab` 会在输出中移除，但在代码块中的空行不必缩进。

```
#!/bin/bash
```

```
echo "Hello Markdown"
```

```
echo "Hello LaTeX"
```

Extension fenced_code_blocks

除了标准的缩进式代码块之外，Pandoc 还支持围栏式代码块，代码块以三个或三个以上的 `~` 符号行开始，以等于或多于开始行 `~` 个数符号行结束，若是代码块中含有 `~`，只需使开始行和结束行中的 `~` 符号个数多于代码块中的即可，见代码 2.3。

代码 2.3: 围栏式代码块

```
~~~~~  
~~~  
code here  
~~~  
~~~~~
```

i Extension `backtick_code_blocks`

与 `fenced_code_blocks` 相同，只不过使用反引号 ``` 替换波浪线 `~` 而已

i Extension `fenced_code_attributes`

可用代码 2.4 为围栏式代码块添加属性。

代码 2.4: 代码块属性

```
~~~ {#code:mycode .bash .numberLines  
↪   startFrom="100" caption=" 围栏式代码块"}  
#!/bin/bash  
echo "fenced_code_attributes"  
~~~
```

代码 2.4 中 `lst:mycode` 为 ID, `.bash` 与 `.numberLines` 是类别, 而 `startFrom` 则是值为 100 的属性。`numberLines` 和 `startFrom` 配合使用可以显示代码行号, 如果没有指定 `start-`

From, 则默认从 1 开始。caption 指定代码块标题, 如果没有设置 caption, 则默认使用 ID 作为 caption。有些输出格式可以利用这些信息来作语法高亮。目前使用到这些信息的输出格式仅有 HTML 与 \LaTeX 。如果指定的输出格式及语言类别有语法高亮支持, 那么上面那段代码区块将会以高亮并带有行号的方式呈现。

仅指定高亮语言时, 可以简写为以下形式:

```
~~~bash
#!/bin/bash
echo "short code highlight"
~~~
```

2.5 行区块

行区块的效果:

The limerick packs laughs anatomical
In space that is quite economical.

But the good ones I've seen

So seldom are clean

And the clean ones so seldom are comical

200 Main St.

Berkeley, CA 94718

i Extension: line_blocks

行区块是从 reStructuredText 借来的语法，以一连串以竖线 | 加上一个空格所构成的连续行。行与行间的区隔在输出时将会以原样保留，行首的空白字元数目也一样会被保留；反之，这些行将会以 markdown 的格式处理。这个语法在输入诗句或地址时很有帮助（代码 2.5）。

代码 2.5: 行区块

```
| The limerick packs laughs anatomical  
| In space that is quite economical.  
|     But the good ones I've seen  
|     So seldom are clean  
| And the clean ones so seldom are comical  
  
| 200 Main St.  
| Berkeley, CA 94718
```

如果有需要的话，书写时也可以将完整一行拆成多行，但后续行必须以空白作为开始。比如下面的示例前两行在输出时会被视为一整行：

```
| The Right Honorable Most Venerable and Righteous  
↪ Samuel L.  
  Constable, Jr.  
| 200 Main St.  
| Berkeley, CA 94718
```

2.6 列表

2.6.1 无序列表

无序列表是以项目符号作列举的列表。每条项目都以项目符号 (*, + 或-) 作开头。以下是个简单的例子：

- ★ one
- ★ two
- ★ three

这会产生一个「紧凑」列表。如果你想要一个「宽松」列表，也就是说以段落格式处理每个项目内的文字内容，那么只要在每个项目间加上空白行即可：

- ★ one
- ★ two
- ★ three

项目符号不必与左边缘齐平；它们可以缩进一个、两个或三个空格。项目符号后面必须有空格。

列表项目中的接续行，若与该项目的第一行文字对齐（在项目符号之后），看上去会较为美观。

- ★ here is my first
list item.
- ★ and my second.

但 markdown 也允许下面「偷懒」的格式：

```
* here is my first  
list item.  
* and my second.
```

2.6.2 区块内容作为列表项

列表项可以包含多个段落和其他区块级内容。但是，后续段落必须在前面加上空行并缩进，以便与列表标记符后面的第一个非空格内容对齐。

```
* First paragraph.
```

```
Continued.
```

```
* Second paragraph. With a code block, which must  
↪ be indented  
eight spaces:
```

```
{ code }
```

⚠ 例外情况

如果列表标记符后面紧跟着一个缩进的代码块，它必须在列表标记符之后开始 5 个空格，那么后续段落必须在列表标记符的最后一个字符之后开始两列：

```
* code
```

```
continuation paragraph
```

列表项可以包括其他列表。在这种情况下，前面的空白行是可选的。嵌套列表必须缩进，以便与包含列表项的列表标记符之后的第一个非空格字符对齐：

```
* fruits
  + apples
    - macintosh
    - red delicious
  + pears
  + peaches
* vegetables
  + broccoli
  + chard
```

上一节提到，Markdown 允许你以「偷懒」的方式书写，项目的接续行可以不和第一行对齐。不过，如果一个列表项目中包含了多个段落或是其他区块元素，那么每个元素的第一行都必须缩进对齐：

```
+ A lazy, lazy, list
item.

+ Another one; this looks
bad but is legal.
```

Second paragraph of second
list item.

2.6.3 有序列表

有序列表与无序列表相类似，唯一的差别在于列表项目是以列举编号作开头，而不是项目符号。

在原始 markdown 中，列举编号是阿拉伯数字后面接着一个句点与空白。数字本身代表的数值会被忽略，因此代码 2.6，代码 2.7 两个列表并无差别。

代码 2.6: 有序列表

1. one
2. two
3. three

代码 2.7: 有序列表数字值会被忽略

5. one
7. two
1. three

i Extension: fancy_lists

与标准的 Markdown 不同，pandoc 除了阿拉伯数字外，还允许用大写字母、小写字母和罗马数字来标记有序列表项。列表标记可以用圆括号括起来，也可以用单右括号或句号括起来。它们必须与后面的文本至少分隔一个空格，如果列表标记是带句号的大写字母，则至少分隔两个空格。

fancy_lists 扩展还允许用 # 代替数字作为有序列表标记：

```
#. one  
#. two
```

i Extension: startnum

除了列表标记外，Pandoc 也能判读列表的起始编号，这两项信息都会保留于输出格式中。举例来说，以下代码可以产生一个从编号 9 开始，以单括号为编号标记的列表，底下还跟着一个小写罗马数字的子列表。

```
9) Ninth  
10) Tenth  
11) Eleventh  
    i. subone  
    ii. subtwo  
    iii. subthree
```

当遇到不同形式的列表标记时，Pandoc 会重新开始一个新的列表。所以，以下代码会产生三份列表：

```
(2) Two
(5) Three
1.  Four
*   Five
```

如果需要预设的有序列表标记符号，可以使用 `#.`：

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

Extension: task_lists

Pandoc 支持任务列表，使用 GitHub-Flavored Markdown 语法。

```
- [ ] an unchecked task list item
- [x] checked item
```

2.6.4 定义列表

定义列表的效果：

Term 1 Definition 1

Term 2 with *inline markup* Definition 2

```
{ some code, part of Definition 2 }
```

Third paragraph of definition 2.

i Extension: definition_lists

Pandoc 支持定义列表，使用带扩展的 PHP Markdown Extra 语法。

Term 1

: Definition 1

Term 2 with *inline markup*

: Definition 2

{ some code, part of Definition 2 }

Third paragraph of definition 2.

每个术语必须单独在一行中，这一行后面可以有选择地后跟空白行，并且必须后跟一个或多个定义。定义以冒号或波浪号开头，可以缩进一个或两个空格。

一个术语可以有多个定义，每个定义可以由一个或多个块元素（段落、代码块、列表等）组成，每个定义缩进四个空格或一个制表符。定义的主体（包括除冒号或波浪号外的第一行）应该缩进四个空格。但是，与其他标记列表一样，您可以“惰性地”省略缩进，除非在段落或其他块元素的开头。

Term 1

: Definition
with lazy continuation.

Second paragraph of the definition.

如果在定义前留出空格（如以上示例所示），定义的文本将被视为段落。在某些输出格式中，这意味着术语/定义对之间的间距更大。若要获得更紧凑的定义列表，请省略定义前的空格：

Term 1

~ Definition 1

Term 2

~ Definition 2a

~ Definition 2b

注意，定义列表中的项之间需要空格。（可以使用 `compact_definition_lists` 激活一个变体，它放宽了这一要求，但不允许“惰性”硬包装）。

2.6.5 编号范例列表

编号范例列表效果如下：

- (1) My first example will be numbered (1).
- (2) My second example will be numbered (2).

Explanation of examples.

(3) My third example will be numbered (3).

i Extension: example_lists

这个特别的列表标记 @ 可以用来产生连续编号的范例列表。列表中第一个以 @ 标记的项目会被编号为‘1’，接着编号为‘2’，依此类推，直到文件结束。范例项目的编号不会局限于单一列表中，而是文件中所有以 @ 为标记的项目均会次序递增其编号，直到最后一个。

```
(@) My first example will be numbered (1).
```

```
(@) My second example will be numbered (2).
```

Explanation of examples.

```
(@) My third example will be numbered (3).
```

编号范例可以加上标签，并且在文件的其他地方引用：

```
(@good) This is a good example.
```

```
As (@good) illustrates, ...
```

标签可以是由任何英文字母、底线或是连字符号所组成的字串。

📌 提示

无论列表标记的长度如何，范例列表中的延续段落都必须缩进 4 个空格。也就是说，范例列表的行为总是像设置了 `four_space_rule` 扩展一样。这是因为示例标签往往很长，将内容缩进到标签之后的第一个非空格字符会很麻烦

2.6.6 紧凑与宽松列表

在与列表相关的「边界处理」上，Pandoc 与 Markdown.pl 有着不同的处理结果。以下代码：

```
+ First
+ Second:
  - Fee
  - Fie
  - Foe
+ Third
```

Pandoc 会将以上列表转换为「紧凑列表」（在“First”，“Second”或“Third”之中没有、`<p>` 标签），而 markdown 则会在“Second”与“Third”（但不包含“First”）里面置入、`<p>` 标签，这是因为“Third”之前的空白行而造成的结果。Pandoc 依循着一个简单规则：如果文字后面跟着空白行，那么就会被视为段落。既然“Second”后面是跟着一个列表，而非空白

行，那么就不会被视为段落了。至于子列表的后面是不是跟着空白行，那就无关紧要了。

📌 提示

即使是设定为 `markdown_strict` 格式，Pandoc 仍是依以上方式处理列表项目是否为段落的判定。这个处理方式与 `markdown` 官方语法规则里的描述一致，然而却与 `Markdown.pl` 的处理不同。

2.6.7 结束一个列表

如果你在列表之后放入一个缩排的代码区块，会有什么结果？

```
- item one
- item two

  { my code block }
```

问题大了！这种情况 Pandoc（其他的 Markdown 实现也是如此）会将 `{ my code block }` 视为 `item two` 这个列表项目的第二个段落来处理，而不会将其视为一个代码区块。

要在 `item two` 之后「切断」列表，你可以插入一些没有缩排、输出时也不可见的内容，例如 HTML 的注释格式：

```
- item one
- item two
```

```
<!-- end of list -->
```

```
{ my code block }
```

当你想要两个各自独立的列表，而非一个大且连续的列表时，也可以运用同样的技巧。

2.7 分隔线

一行中若包含三个以上的 * , - 或 _ 符号（中间可以以空白字元分隔），则会产生一条分隔线：

```

_____

*   *   *   *
-----
```

2.8 表格

有四种表格的形式可以使用。前三种适用于等宽字型的编辑环境，例如 Courier。第四种则不需要直行的对齐，因此可以在比例字型的环境下使用。

2.8.1 简单表格

❗ Extension: table_captions

4 种表格都支持添加可选的标题。标题是以 `Table:` 字符串（或简写为 `:`）开头的段落，可以出现在表格开头或者结尾。

❗ Extension: simple_tables

简单表格如表 2.1 所示，源码见代码 2.8。

表 2.1: 简单表格演示

Right		Left	Center		Default
12	12		12		12
123	123		123		123
1	1		1		1

表头和行必须是一行，不能换行。列对齐方式根据表头文字和其底下虚线的相对位置来决定：

- 如果虚线与表头文字的右侧对齐，而左侧比表头文字长，则该列右对齐。
- 如果虚线与表头文字的左侧对齐，而右侧比表头文字长，则该列左对齐。
- 如果虚线的两侧都比表头文字长，则该列居中对齐。

代码 2.8: 简单表格源码

Right	Left	Center	Default
-----	-----	-----	-----
12	12	12	12
123	123	123	123
1	1	1	1

Table: 简单表格演示 {#tbl:simple_table}

- 如果虚线与表头文字的两侧都对齐，则会套用预设的对齐方式（在大多数情况下，这将会是左对齐）。
- 表格底下必须接着一个空白行，或是一行虚线后再一个空白行。

表头也可以省略，在省略表头的情况下，表格下方必须加上一行虚线以清楚标明表格的范围。例如代码 2.9 显示为：

12	12	12	12
123	123	123	123
1	1	1	1

代码 2.9: 省略表头的表格

-----	-----	-----	-----
12	12	12	12
123	123	123	123
1	1	1	1
-----	-----	-----	-----

当省略表头时，对齐根据表格内容的第一行决定。所以，以上面的表格为例，各列的对齐依序会是右对齐、左对齐、居中以及右对齐。

2.8.2 多行表格

i Extension: multiline_tables

多行表格允许表头与行的文字以多行呈现（但不支持横跨多栏或纵跨多列的单元格）。范例如表 2.2，源码见代码 2.10。

表 2.2: 这里是标题
标题也可以是多行

Centered Header	Default Aligned	Right Aligned	Left Aligned
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.

看起来很像简单表格，但两者间有以下差别：

- 在表头文字之前，必须以一系列虚线作为开头（除非有省略表头）。

- 必须以一系列虚线作为表格结尾，之后接一个空白行。
- 行之间以空白行隔开。

⚠ 多行标题

注意，多行标题第一行末尾需用 2 个空格强制换行。

在多行表格中，表格分析器会计算各列的宽度，并在输出时尽可能维持各列在原始文件中的相对比例。因此，要是你觉得某列在输出时不够宽，你可以在 markdown 源码种加宽一点。

和简单表格一样，表头在多行表格中也是可以省略的：

表 2.3: 无表头的多行表格

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.

多行表格可以只包含一行，但该行之后必须接着一个空白行（然后才是表示表格结尾的一行虚线）。如果没有此空白行，此表格将会被解读成简单表格。

2.8.3 格框表格

 **Extension: grid_tables**

格框表格看起来像表 2.4，源码见代码 2.11。

表 2.4: 格框表格示例

Fruit	Price	Advantages
Bananas	\$1.34	built-in wrapperbright color
Oranges	\$2.10	cures scurvytasty

以 = 串成的一行区分了表头与表格本体，这在没有表头的表格中也是可以省略的。在格框表格中的单元格可以包含任意的区块元素（多个段落、代码块、列表等等）。单元格不支持横跨多栏或纵跨多列。格框表格可以在 Emacs table mode 下轻松建立。

通过在表头与表格分界行种添加冒号来指定对齐方式(源码代码 2.12)：

Right	Left	Centered
Bananas	\$1.34	built-in wrapper

对于没有表头的表格，冒号放在第一行：


```
┌
↩  +-----:+:-----+-----:-----
| Right          | Left          | Centered
↩  |
└
↩  +-----+-----+-----+-----
```

⚠ 格框表格的局限性

Pandoc 不支持合并行或列的表格，必须有相同数量的行和列。例如，Docutils 的 [示例表格](#) 在 Pandoc 种不会像预期那样渲染。

2.8.4 管线表格

📌 Extension: pipe_tables

管线表格看起来像表 2.5，源码见代码 2.13。

表 2.5: 管线表格示例

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

这个语法与 [PHP Markdown Extra tables](#) 中的表格语法相同。每行开头与结尾的管线字符是可选的，但各列间的管线不可省略。上面范例中的冒号表明了对齐方式。表头可以省略，但表头下的水平虚线必须保留，因为虚线上定义了列对齐方式。

因为管线界定了各栏之间的边界，表格的原始码并不需要像上面例子中各栏之间保持直行对齐。所以，以下例子是个完全合法（虽然丑陋）的管线表格：

```
fruit| price
-----|-----:
apple|2.05
pear|1.37
orange|3.09
```

管线表格的单元格不能包含如段落、列表之类的区块元素，也不能包含多行文字。如果管线表格包含了比 `column` 定义的宽度（参见 [-columns](#)）还要宽的行，表格宽度将会被设置为文本宽度，并且单元格会自动换行，并且各列的相对宽度由表头的连接线数量决定。（比如，`---|-` 将会生成第一列占文本宽度 3/4，第二列占文本宽度 1/4 的表格）另一方面，如果没有比列宽更宽的行，那么单元格内容将不会被换行，并且单元格将根据其内容大小进行调整。自动换行效果见表 [2.6](#)。

orgtbl-mod

Pandoc 也可以看得懂以下形式的管线表格，这是由 Emacs 的 `orgtbl-mod` 所绘制：

表 2.6: 自动换行示例

fruit	price
apple apple apple apple apple apple	2.05
apple apple apple apple apple apple	
apple apple apple apple apple apple	
apple	
pear	1.37
orange	3.09

```
| One | Two |
|-----+-----|
| my | table |
| is | nice |
```

主要的差别在于以 + 取代了 |。其他的 `orgtbl` 功能并未支援。如果要指定非预设的列对齐形式，你仍然需要在上面的表格中自行加入冒号。

2.9 元数据区块

2.9.1 Pandoc 标题区块

(译注：本节中提到的「标题」均指 Title，而非 Headers)

i Extension: pandoc_title_block

标题区块源码见代码 2.14。

如果一个文件以文件标题 (Title) 区块开头, 这部份将不会作为一般文字处理, 而会以书目信息的方式解析。(这可用在像是单一 LaTeX 或是 HTML 输出文件的书名上。) 这个区块仅能包含标题, 或是标题与作者, 或是标题、作者与日期。如果你只想包含作者却不想包含标题, 或是只有标题与日期而没有作者, 你需要利用空白行:

```
%  
% Author  
  
% My title  
%  
% June 15, 2006
```

标题可以包含多行文字, 但接续行必须以空白字元开头, 例如:

```
% My title  
  on multiple lines
```

如果文件有多个作者, 作者也可以分列在不同行并以空白字元作开头, 或是以分号间隔, 或是两者并行。所以, 下列各种写法得到的结果都是相同的:

```
% Author One
  Author Two
```

```
% Author One; Author Two
```

```
% Author One;
  Author Two
```

日期就只能写在一行之内。

所有这三个 Metadata 字段都可以包含标准的行内格式(斜体、链接、脚注等等)。

2.9.2 元数据区块

文件标题区块功能有限，实际使用中建议用 YAML 元数据区块方式。

Extension: `yaml_metadata_block`

YAML 元数据区块是一个有效的 YAML 对象，开头和结尾使用包含 3 个连字符 (---) 的单独一行来界定(结尾也可以使用 3 个点 ...)。YAML 元数据区块可以出现的文档的任何地方，但是如果不在文档开头，它的前面必须空一行。YAML 元数据区块也可以单独写到一个文件里，使用 `--metadata-file` 参数传递给 Pandoc。(本项目除了简历功能，都使用了独立的 metadata file，路径为 `src/metadata.yaml`)

元数据将从 YAML 对象的字段中提取，并添加到任何现

有的文档元数据中。元数据可以包含列表和对象（任意嵌套），但是所有字符串标量都将被解释为 Markdown。名称以下划线结尾的字段将被 Pandoc 忽略。（它们可能被外部程序处理。）字段名不能包含能被解释为 YAML 数字或布尔值的字符串（比如，yes, True, and 15 不能作为字段名）。

一个文档可以包含多个元数据块。元数据字段的取值将通过 left-biased union 确定：如果两个元数据块试图设置相同的字段，则第一个元数据块的值会生效。

注意，元数据区块必须遵循 YAML 转义规则。因此，如果一个值包含冒号，它必须被引号包裹。可以使用管道字符 (|) 开始一个缩进的块，该块将按字面解释，不需要转义。当字段包含空行或块级格式化时，需要使用这种方式：

```
---
title: 'This is the title: it contains a colon'
author:
- Author One
- Author Two
keywords: [nothing, nothingness]
abstract: |
  This is the abstract.

  It consists of two paragraphs.
...
```

模板变量将从元数据中自动设置。例如，在编写 HTML 时，变量 abstract 将被设置为与 abstract 字段中的 Markdown 等价的 HTML：

```
<p>This is the abstract.</p>
<p>It consists of two paragraphs.</p>
```

变量可以包含任意的 YAML 结构，但是模板必须匹配这个结构。默认模板中的 `author` 变量需要一个简单的列表或字符串，但是可以更改为支持更复杂的结构。例如，如果给定一个从属关系，下面的组合将向作者添加一个从属关系：

```
---
title: The document title
author:
- name: Author One
  affiliation: University of Somewhere
- name: Author Two
  affiliation: University of Nowhere
...
```

要使用上面例子中的 `author` 字段，您需要一个自定义模板：

```
$for(author)$
$if(author.name)$
$author.name$$if(author.affiliation)$
↪ ($author.affiliation)$endif$
$else$
$author$
$endif$
$endfor$
```

要包含在文档 `header` 部分中的原始内容可以使用 `head`

指定；但是，使用 `raw_attribute` 扩展将该内容标记为特定输出格式的原始代码非常重要，否则它将被解释为 Markdown。例如：

```
header-includes:
```

```
- |  
  ```{=latex}  
 \let\oldsection\section

↪ \renewcommand{\section}[1]{\clearpage\oldsection
  ````
```

2.10 字符转义

i Extension: `all_symbols_escapable`

除了在代码区块或行内代码之外，任何标点符号或空白字元前面只要加上一个反斜线，都能使其保留字面原义，而不会进行格式的转义解读。

例如，下面的写法

```
*\*hello\**
```

输出后会得到

`*hello*`

而不是

`hello`

这条规则比原始的 Markdown 规则来得好记许多，原始规则中，只有以下字元才支持反斜线转义：

`\`*_{}[]()>#+-.,!`

（然而，如果使用了 `markdown_strict` 格式，那么就会采用原始的 `markdown` 规则）

一个反斜线之后的空白字元会被解释为不断行的空白（nonbreaking space）。这在 `TEX` 的输出中会显示为 `~`，而在 HTML 与 XML 则是显示为 `\ `；或 `\ sp`；。

一个反斜线之后的换行符（例如反斜线符号出现在一行的最尾端）则会被解释为强制换行。这在 `TeX` 的输出中会显示为 `\\`，而在 HTML 里则是 `
`。相对于原始 Markdown 是以在行尾加上两个空白字元这种「看不见」的方式进行强制换行，反斜线接换行字元会是比较好的替代方案。

转义字符在代码区块中不起任何作用。

2.11 行内格式

2.11.1 强调

要强调某些文字，只要以 * 或 _ 符号前后包住即可，像这样：

```
This text is _emphasized with underscores_, and  
↪ this  
is *emphasized with asterisks*.
```

重复两个 * 或 _ 符号以产生更**强烈**的 **强调**：

```
This is **strong emphasis** and __with  
↪ underscores__.
```

一个前后以空格包住，或是前面加上反斜线的 * 或 _ 符号，都不会转换为强调格式：

```
This is * not emphasized *, and \*neither is  
↪ this\*.
```

i Extension: intraword_underscores

因为 _ 字元有时会使用在单词或是 ID 之中，所以 Pandoc 不会把被字母包住的 _ 解读为强调标记。如果有需要特别强调单词中的一部分，需要用 *：

```
feas*ible*, not feas*able*.
```

2.11.2 删除线

i Extension: **strikeout**

要将一段文字加上水平线作为删除效果，将该段文字前后以 ~~ 包住即可。

This ~~is deleted text.~~

2.11.3 上标与下标

i Extension: **superscript,subscript**

要输入上标可以用 ^ 字符将要上标的文字包起来；要输入下标可以用 ~ 字符将要下标的文字包起来。比如 2^{10} 是 1024，源码为：

```
H~2~0 is a liquid. 2^10^ is 1024.
```

如果要上标或下标的文字中包含了空格，那么这个空格之前必须加上反斜线。（这是为了避免一般使用下的 ~ 和 ^ 在非预期的情况下产生出意外的上标或下标。）所以，如果你想要让字母 P 后面跟着下标文字'a cat'，那么就要输入 $P_{a\ cat}$ ，而不是 $P_a\ cat$ 。

2.11.4 字面文字

要让一小段文字直接以其字面形式呈现，可以用反引号将其包住：

What is the difference between `>>=` and `>>`?

如果字面文字中也包含了反引号，那就使用双重反引号包住：

Here is a literal backtick ``` ` ```.

（在起始反引号后的空格以及结束反引号前的空格都会被忽略。）

一般性的规则如下，字面文字区段是以连续的反引号字元作为开始（反引号后的空格为可选），一直到同样数目的反引号字元出现才结束（反引号前的空格也为可选）。

要注意的是，转义字符（以及其他 Markdown 结构）在字面文字的上下文中是没有效果的：

This is a backslash followed by an asterisk: ``.*

Extension: inline_code_attributes

与围栏代码区块一样，字面文字也可以附加属性：

```
<$>`{.haskell}
```

2.11.5 小型大写字母

小型大写字母使用章节 2.17 语法, 指定 `.smallcaps` 样式:

```
[Small caps]{.smallcaps}
```

或者, 不用 `bracketed_spans` 扩展, 直接写 HTML:

```
<span class="smallcaps">Small caps</span>
```

为了兼容其他 Markdown 风格, 也支持 CSS:

```
<span style="font-variant:small-caps;">Small  
↪ caps</span>
```

此功能在所有支持小型大写字母的输出格式中都有效。

2.12 数学公式

❗ Extension: `tex_math_dollars`

所有介于两个 `$` 字符之间的内容将会被视为 $\text{T}_{\text{E}}\text{X}$ 数学公式处理。开头的 `$` 右侧必须立刻接上任意文字, 而结尾 `$` 的左侧同样也必须紧挨着文字。这样一来, `$20,000` and `$30,000` 就不会被当作数学公式处理了。如果基于某些原因, 有必须使用 `$` 符号将其他文字括住的需求时, 那么可以在 `$` 前使用反斜线转义字符, 这样 `$` 就

不会被当作数学公式的分隔符。

$\text{T}_{\text{E}}\text{X}$ 数学公式会在所有输出格式中打印。至于如何渲染则取决于输出的格式：

- $\text{E}_{\text{T}}\text{X}$ ：转换为 $\backslash(\dots\backslash)$ （行内公式）或者 $\backslash[\dots\backslash]$ （跨行公式）
- Docx：公式会以 OMML 数学标记的方式渲染。
- HTML 和 EPUB：公式会依照命令行 [选项](#) 的设置，以不同的方法演算编排为 HTML 代码。

以下是一个例子，显示效果见公式 2.1：

```
$$\begin{cases} a_1x+b_1y+c_1z=d_1\\ a_2x+b_2y+c_2z=d_2\\ a_3x+b_3y+c_3z=d_3 \end{cases} \\ \end{cases} \quad \{\#eq:math\_demo\}
```

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases} \quad (2.1)$$

2.13 Raw HTML

i Extension: raw_html

Markdown 允许你在文件中的任何地方插入原始 HTML（或 DocBook）指令（除了在代码块中，此时的 `<`, `>` 和 `&` 都会按其字面意义显示）。（技术上而言这不算扩充功能，因为原始 Markdown 本身就有提供此功能，但做成扩充形式便可以在有特殊需要的时候关闭此功能。）

输出 HTML, S5, Slidy, Slideous, DZSlides, EPUB, Markdown, CommonMark, Emacs Org mode 以及 Textile 等格式时，原始 HTML 代码会不作修改地保留至输出档案中；而其他格式的输出内容则会将原始 HTML 代码去除掉。

i Extension: markdown_in_html_blocks

原始 Markdown 允许你插入 HTML「区块」：所谓的 HTML 区块是指，上下各由一个空白行所隔开，开始与结尾均由所在行最左侧开始的一连串对称均衡的 HTML 标签。在这个区块中，任何内容都会当作是 HTML 来分析，而不再视为 Markdown；所以（举例来说），`*` 符号就不再代表强调。

当指定格式为 `markdown_strict` 时，Pandoc 会以上述方式处理；但预设情况下，Pandoc 能够以 Markdown 语法解读 HTML 区块标签中的内容。举例说明，Pandoc 能够将代码 2.15 转换为代码 2.16。

而 `Markdown.pl` 则是保留该段原样。

这个规则只有一个例外:那就是介于 `<script>` 与 `<style>` 之间的文字都不会被拿来当作 Markdown 解读。

这里与原始 Markdown 的分歧,主要是为了让 Markdown 能够更便利地混入 HTML 区块元素。比方说,一段 Markdown 文字可以用 `<div>` 标签将其前后包住来进行样式指定,而不用担心里面的 Markdown 不会被解释到。

i Extension: native_divs

直接使用 HTML 的 `<div>` 标签,大部分情况下将产生和 `markdown_in_html_blocks` 相同的输出。使用 `native_divs` 的好处是让 `pandoc filter` 更容易的处理。

i Extension: native_spans

直接使用 HTML 的 `` 标签,输出和 `raw_html` 一样。使用 `native_spans` 的好处也是让 `pandoc filter` 更容易的处理。

i Extension: raw_tex

除了原始 HTML 之外, Pandoc 还允许在文档中包含原始 \LaTeX 、 \TeX 和 `ConTeXt`。行内 \TeX 命令将被保留,并不加更改地传递给 \LaTeX 和 `ConTeXt`。例如,你可以使用 \LaTeX 来引用 BibTeX:

This result was proved in `\cite{jones.1967}`.

注意，在 \LaTeX 环境中，比如：

```
\begin{tabular}{|l|l|}\hline
Age & Frequency \\ \hline
18--25 & 15 \\
26--35 & 33 \\
36--45 & 22 \\ \hline
\end{tabular}
```

`begin` 和 `end` 之间的内容会被解释成原始的 \LaTeX ，而不是 Markdown。

有关在标记文档中包含原始 \TeX 的更显式和更灵活的方法，请参见 `raw_attribute` 扩展章节 [2.13.1](#)。

除了 Markdown、 \LaTeX 、Emacs Org mode 和 ConTeXt 之外，行内 \LaTeX 在输出格式中被忽略。

2.13.1 通用 `raw_attribute`

i Extension: `raw_attribute`

此扩展和本项目关系不大，请阅读 [原始文档](#) 了解更多内容。

2.14 L^AT_EX 宏指令

Extension: latex_macros

当输出格式不是 L^AT_EX 时，Pandoc 会分析 L^AT_EX 的 `\newcommand` 和 `\renewcommand` 定义，并套用其产生的巨集到所有 L^AT_EX 数学公式中。所以，举例来说，下列指令对于所有的输出格式均有作用，而非仅仅作用于 L^AT_EX 格式：

```
\newcommand{\tuple}[1]{\langle #1 \rangle}
```

```
 $\tuple{a, b, c}$ 
```

在 L^AT_EX 的输出中，`\newcommand` 定义会单纯不作修改地保留至输出结果。

2.15 链接

Markdown 支持以下数种指定链接的方式。

2.15.1 自动链接

如果你用角括号将一段 URL 或是 email 位址包起来，它会自动转换成链接：

```
<http://google.com>  
<sam@green.eggs.ham>
```

2.15.2 行内链接

一个行内链接包含了位在方括号中的链接文字，以及方括号后以圆括号包起来的 URL。（你可以选择性地在 URL 后面加入链接标题，标题文字要放在引号之中。）

```
This is an [inline link](/url), and here's [one  
↪ with  
a title](http://fsf.org "click here for a good  
↪ time!").
```

方括号与圆括号之间不能有空白。链接文字可以包含格式（例如强调），但链接标题不可以。

行内链接中的电子邮件地址不会自动检测到，因此它们必须以 `mailto` 开头：

```
[Write me!](mailto:sam@green.eggs.ham)
```

2.15.3 参考链接

一个明确的参考链接包含两个部分，链接本身以及链接定义，其中链接定义可以放在文件的任何地方（不论是放在链接所在处之前或之后）。

链接本身是由两组方括号所组成，第一组方括号中为链接文字，第二组为链接标签。（在两个方括号间不能有空格。）

链接定义则是以方括号框住的链接标签作开头，后面跟着一个冒号一个空白，再接着一个 URL，最后可以选择性地（在一个空白之后）加入由引号或是圆括号包住的链接标题。链接标签不能被解析为 citation（假设启用了 citations 扩展，参见章节 2.19）：citation 的优先级高于链接标签。

以下是一些范例：

```
[my label 1]: /foo/bar.html "My title, optional"
[my label 2]: /foo
[my label 3]: http://fsf.org (The free software
↪ foundation)
[my label 4]: /bar#special 'A title in single
↪ quotes'
```

链接的 URL 也可以选择性地以角括号包住：

```
[my label 5]: <http://foo.bar.baz>
```

链接标题可以放在第二行，效果见 my label 3：

```
[my label 3]: http://fsf.org
"The free software foundation"
```

需注意链接标签并不区分大小写。所以下面的例子会建立合法的链接：

```
Here is [my link][FOO]
```

```
[Foo]: /bar/baz
```

在一个隐性参考链接中，第二组方括号的内容是空的：

```
See [my website][].
```

```
[my website]: http://foo.bar.baz
```

注意：在 Markdown.pl 以及大多数其他 Markdown 实现中，参考链接的定义不能存在于嵌套结构中，例如清单项目或是引用区块。Pandoc 解除了这个看似武断的限制。所以虽然下面的语法在几乎所有其他实现中都是错误的，但在 Pandoc 中可以正确处理：

```
> My block [quote].  
>  
> [quote]: /foo
```

Extension: shortcut_reference_links

在 shortcut_reference_links 中，第二对括号可完全省略：

```
See [my website].
```

```
[my website]: http://foo.bar.baz
```

2.15.4 内部链接

要链接到同一份文件的其他章节，可使用自动产生的 ID (参见章节 2.2)。

See the `[Introduction](#introduction)`.

或是

See the `[Introduction]`.

`[Introduction]: #introduction`

内部链接目前支持的格式有 HTML（包括 HTML slide shows 与 EPUB）、 \LaTeX 以及 ConTeXt。

pandoc-crossref

也可以使用 `pandoc-crossref` 的方式来处理内部链接，参见章节 2.20。

2.16 图片

在链接语法的前面加上一个 `!` 就是图片的语法了。链接文字将会作为图片的替代文字 (`alt text`):

```
![la lune](Pictures/background.pdf "Voyage to the  
↪ moon")
```

```
![movie reel]
```

```
[movie reel]: movie.gif
```

i Extension: implicit_figures

一个图片若自身单独存在一个段落中，那么将会以附上图片说明（caption）的图表（figure）形式呈现。（在 LaTeX 中，会使用图表环境；在 HTML 中，图片会被放在具有 figure 类别的 div 元素中，并会附上一个具有 caption 类别的 p 元素。）图片的 alt 文字同时也会用来作为图片说明。

```
![This is the caption](/url/of/image.png)
```

如何呈现取决于输出格式。一些输出格式（例如 RTF）还不支持插图标题。在这些格式中，您只会在一个段落中单独获得一个没有标题的图片。

如果您只是想要一个常规的行内图像，请确保它不是该段中的唯一内容。一种方法是在图像后面插入一个不间断的空格：

```
![This image won't be a figure](/url/of/image.png)\
```

注意，在 reveal.js 幻灯片显示中，一个段落中的图像本身具有 stretch 样式时将填充屏幕，而标题和图标记将被忽略。

i Extension: link_attributes

链接和图像可以设置属性，如代码 2.17。

(当只使用 `#id` 和 `.class` 时, 此语法与 [PHP Markdown Extra](#) 兼容。)

对于 HTML 和 EPUB, 除了 `width` 和 `height` (但包括 `srcset` 和 `sizes`) 以外的所有属性都按原样传递。其他编写器忽略输出格式不支持的属性。

对图像的宽度和高度属性进行了特殊处理。如果不使用单位, 则假定单位为像素。但是, 可以使用以下任何单位标识符: `px`、`cm`、`mm`、`in`、`inch` 和 `%`。数字和单位之间不能有空格。例如:

```
{ width=50% }
```

- 尺寸转换为英寸, 以便以基于页面的格式 (如 \LaTeX) 输出。尺寸被转换为像素, 以便以类似 HTML 的格式输出。使用 `--dpi` 选项指定每英寸的像素数。默认值是 `96dpi`。
- `%` 单位通常是基于一些可用空间的相对值。例如, 上面的例子将呈现给以下内容。
 - HTML: ``
 - \LaTeX : `\includegraphics[width=0.5\textwidth]{file.jpg}` (如果使用自定义模板, 则需要像默认模板中那样设置 `graphicx`。)
- 当没有指定宽度或高度属性时, 应变的方法是查看图像分辨率和嵌入到图像文件中的 `dpi` 元数据。

i 图片交叉引用

使用 `pandoc-crossref` 处理图片交叉引用，请参考章节 2.20。

2.17 Divs 和 Spans 语法

使用 `native_divs` 和 `native_span` 扩展（见章节 2.13），HTML 语法可以作为 Markdown 的一部分，在 `pandoc AST` 中创建本地 `Div` 和 `Span` 元素（与原始 HTML 相反）。不过，还有更好的语法。

i Extension: fenced_divs

为原生 `Div` 块提供了特殊的围栏式语法。`Div` 以一个包含至少三个连续冒号和一些属性的围栏开始。属性后面可以选择跟随另一串连续冒号。属性语法与围栏式代码块完全相同（参见 [Extension: fenced_code_attributes](#)）。与围栏式代码块一样，可以使用大括号中的属性，也可以使用单个无括号单词（无括号单词将被视为样式名）。`Div` 以另一个包含至少三个连续冒号组成的字符串的行结束。围栏式 `Div` 应该用空行与前面和后面的区块分隔。

示例：

```
::::: {#special .sidebar}
Here is a paragraph.

And another.
:::::
```

围栏式 Div 支持嵌套。开头围栏很容易区分因为它们带有属性：

```
::: Warning ::::::
This is a warning.

::: Danger
This is a warning within a warning.
:::
::::::::::::::::::::
```

没有属性的栏总是围栏结尾。与围栏式代码块不同，围栏结尾中的冒号数不必与围栏开头中的冒号数匹配。然而，使用不同长度的栅栏来区分嵌套 div 和它们的父 div 可能有助于提高视觉清晰度。

 **Extension: bracketed_spans**

用括号括起来的行内序列，语法和链接开头一样，如果紧接其后的是属性，则将被视为带有属性的 Span：

```
[This is some text]{.class key="val"}
```

2.18 脚注

i 脚注

Pandoc's markdown 支持脚注功能，使用如代码 2.18 所示的语法。

Here is a footnote reference,¹ and another.²

This paragraph won't be part of the note, because it isn't indented.

脚注参考用的 ID 不得包含空格、制表符或换行符。这些 ID 只会用来建立脚注位置与脚注文字的对应关连；在输出时，脚注将会依序递增编号。

脚注本身不需要放在文件的最后面。它们可以放在文件里的任何地方，但不能被放入区块元素（列表、引用、表格等）之中。每个脚注应与周围的内容（包括其他脚注）用空行隔开。

i Extension: inline_notes

Pandoc 也支持行内脚注（尽管，与一般脚注不同，行内脚注不能包含多个段落）。其语法如下：

¹Here is the footnote.

²Here's one with multiple blocks.

Subsequent paragraphs are indented to show that they belong to the previous footnote.

```
{ some.code }
```

The whole paragraph can be indented, or just the first line. In this way, multi-paragraph footnotes work like multi-paragraph list items.

```
Here is an inline note.^[Inlines notes are  
↪ easier to write, since  
you don't have to pick an identifier and move  
↪ down to type the  
note.]
```

Here is an inline note.³

行内与一般脚注可以自由交错使用。

2.19 参考文献

⚠ 注意

参考文献请作为正文处理, 勿放在 `backmatter.md` 中, 因为 `backmatter.md` 是单独编译成 \LaTeX 的, `pandoc-citeproc` 不能读取到正文中的引用。

📖 Extension: citations

使用外部 filter (`pandoc-citeproc`), Pandoc 可以自动生成多种风格的引用和文献目录。基本用法是

```
$ pandoc --filter pandoc-citeproc myinput.txt
```

³Inlines notes are easier to write, since you don't have to pick an identifier and move down to type the note.

为了使用这个功能，您需要通过 YAML 元数据区块中的 `bibliography` 字段或 `--bibliography` 命令行参数指定参考文献文件。如果希望使用多个参考文献文件，可以在 YAML 元数据区块中设置多个 `bibliography` 字段，或者多次设置 `--bibliography` 参数。参考文献可采用表 2.7 中任何一种格式。

表 2.7: Pandoc 支持的引用形式

| Format | File extension |
|-------------|----------------|
| BibLaTeX | .bib |
| BibTeX | .bibtex |
| Copac | .copac |
| CSL JSON | .json |
| CSL YAML | .yaml |
| EndNote | .enl |
| EndNote XML | .xml |
| ISI | .wos |
| MEDLINE | .medline |
| MODS | .mods |
| RIS | .ris |

需注意 `.bib` 扩展名同时适用于 BibTeX 与 BibLaTeX，使用 `.bibtex` 来强制指定 BibTeX。

注意，`pandoc-citeproc --bib2json` 和 `pandoc-citeproc --bib2yaml` 可以从任何受支持的格式生成 `.json` 和 `.yaml`

文件。

i 标记方式说明

在 BibTeX 和 BibLaTeX 数据库中, pandoc-citeproc 解析 \LaTeX 标记的子集; 在 CSL YAML 数据库中, pandoc 解析 Markdown; 在 CSL JSON 数据库中, 解析类似 HTML 的标记:

```
<i>...</i>
```

italics

```
<b>...</b>
```

bold

```
<span style="font-variant:small-caps;">...</span>
```

↪ `` or

↪ `<sc>...</sc>`

small capitals

```
<sub>...</sub>
```

subscript

```
<sup>...</sup>
```

superscript

```
<span class="nocase">...</span>
```

prevent a phrase from being capitalized as

↪ title case

pandoc-citeproc 的 `-j` 和 `-y` 选项尽可能相互转换 CSL JSON 和 CSL YAML 格式。

作为使用 `--bibliography` 或 YAML 元数据中 `bibliography` 字段指定参考文献文件的替代方法, 您可以将引文数据直接

包含在文档的 YAML 元数据的 references 字段中。字段应该包含一个 YAML 编码的引用数组，例如：

```
---
references:
- type: article-journal
  id: WatsonCrick1953
  author:
  - family: Watson
    given: J. D.
  - family: Crick
    given: F. H. C.
  issued:
    date-parts:
      - - 1953
        - 4
        - 25
  title: 'Molecular structure of nucleic acids: a
↪  structure for deoxyribose
    nucleic acid'
  title-short: Molecular structure of nucleic acids
  container-title: Nature
  volume: 171
  issue: 4356
  page: 737-738
  DOI: 10.1038/171737a0
  URL:
↪  http://www.nature.com/nature/journal/v171/n4356/
  language: en-GB
...
```

(`pandoc-citeproc --bib2yaml` 可以从支持的格式之一的参考文献文件中生成 `references` 字段格式。)

可以使用 [Zotero 样式库](#) 中列出的 [引文样式语言](#) 支持的任何样式对引文和引用进行格式化。这些文件是使用 `--csl` 选项或 `csl` 元数据字段指定的。默认情况下, `pandoc-citeproc` 将使用样式为 `author-date` 格式的 [Chicago Manual of Style](#)。CSL 项目提供了关于 [查找和编辑样式](#) 的进一步信息。

要使引用超链接到相应的参考文献条目, 请在 YAML 元数据区块中设置 `link-citations: true`。

文献引用放在方括号中, 以分号隔开。每一条引用都需要有一个 key, 由 `@` 加上文献目录数据库中的文献 ID 组成, 并且可以选择性地包含前缀、定位以及后缀。引用键必须以字母、数字或 `_` 开头, 并且可以包含字母数字、`_` 和内部标点符号 (`:.#%&-+?<>~/`)。以下是一些范例:

```
Blah blah [see @doe99, pp. 33-35; also @smith04,  
↪ ch. 1].
```

```
Blah blah [@doe99, pp. 33-35, 38-39 and *passim*].
```

```
Blah blah [@smith04; @doe99].
```

`pandoc-citeproc` 检测 [CSL 语言环境文件](#) 中的定位项。缩写形式和非缩写形式都可以。在 `en-US` 语言环境中, 定位器术语可以用单数或复数形式编写, 比如 `book`, `bk./bks.`; `chapter`, `chap./chaps.`; `column`, `col./cols.`; `figure`, `fig./figs.`; `folio`, `fol./fols.`; `number`, `no./nos.`; `line`, `l./ll.`; `note`, `n./nn.`; `opus`, `op./opp.`; `p`

p./pp.; paragraph, para./paras.; part, pt./pt.
section, sec./secs.; sub verbo, s.v./s.vv.; v
v./vv.; volume, vol./vols.; ¶/¶¶; §/§§. 如果没有
使用定位器术语, 则默认为 page。

pandoc-citeproc 将使用启发式来区分定位符和后缀。在
复杂的情况下, 定位器可以用大括号括起来 (需要 pandoc-
citeproc 0.15 或更高版本):

```
[@smith{ii, A, D-Z}, with a suffix]  
[@smith, {pp. iv, vi-xi, (xv)-(xvii)} with suffix  
↪ here]
```

在 @ 前面的减号 (-) 将会避免作者名字在引用中出现。
这可以用在已经提及作者的文章场合中:

```
Smith says blah [-@smith04].
```

你也可以在文字中直接插入文献引用, 方式如下:

```
@smith04 says blah.
```

```
@smith04 [p. 33] says blah.
```

如果样式需要引用的文献列表, 文献列表将被放置在一个带有 ID 为 refs 的 div 中, 如果存在此 div:

```
::: {#refs}  
:::
```

否则，文献列表将被放置在文档结尾。可以通过在 YAML 元数据中设置 `suppress-bibliography: true` 来抑制文献列表的生成。

如果你希望文献列表有一个标题，你可以在元数据中设置 `reference-section-title`，或者在文档的末尾加上：

```
last paragraph...
```

```
# References
```

另一种方式，可以使用 `fenced_divs` 语法，参考文献将显示在带 `{#refs}` 属性的 `div` 中第一个大标题下：

```
::: {#refs}
# 我的参考文献
:::
```

文献列表将插入本标题之后。注意，`.unnumbered` 样式将被添加到这个标题中，这样该部分就不会被编号。

有时正文中没有引用也需要在参考文献列表显示，可以通过 YAML 元数据区块定义 `nocite`：

```
nocite: |
  @item1,@item2
```

通过使用通配符，可以创建包含所有引用的文献列表，无论它们是否出现在文档中（这种情况在简历制作时会遇到）：

```
nocite: |  
  @*  
  ...
```

对于 LaTeX 输出，还可以使用 `natbib` 或 `biblatex` 来呈现参考书目。要做到这一点，请按照上面所述指定书目文件，并将 `--natbib` 或 `--biblatex` 参数添加到 `pandoc` 调用中。记住，书目文件必须是各自的格式（BibTeX 或 BibLaTeX）。

更多信息请参考 [pandoc-citeproc man page](#)。

2.20 交叉引用

PanBook 使用 [pandoc-crossref](#) 处理交叉引用。

2.20.1 图片

`![Caption](file.ext){#fig:label}`

要标记(隐式)图形,请在图像定义之后立即附加 `{#fig:label}` (其中 `label` 需要是一个独一无二的插图标签)。

这只适用于隐式图形，即在段落中单独出现的图像（将由 `pandoc` 呈现为带有标题的图形）

图像块和标签不能用空格分开。

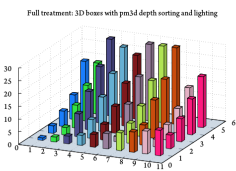
子图

结合 `fenced_divs` 语法,可以将图片分组为子图片。基本语法如下（效果见图 2.1）：

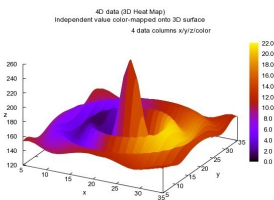
```
::: {#fig:figureRef}
![subfigure 1
↪ caption](images/image1.png){#fig:figureRefA}

![subfigure 2
↪ caption](images/image2.png){#fig:figureRefB}

Caption of figure
:::
```



(a) subfigure 1 caption



(b) subfigure 2 caption

图 2.1: Caption of figure

综上所述，子图由一个具有图片 id 的 div 构成。该 div 的内容由几个段落组成。除最后一段外，所有段落都包含一个子图，并带有标题、图像和（可选的）引用属性。最后一段包

含图片标题。

如果在段落中放置多个图片，那么仍然会呈现这些图形，但是 Pandoc 会在大多数输出中省略子图形标题(但是在 \LaTeX 中它会像预期的那样工作)。您可以使用特定于输出的技巧来解决这个问题，或者使用 `subfigGrid` (见下文)。

输出是可定制的，使用元数据字段。有关更多信息，请参见 [Customization](#)。

默认设置将从上面的例子中产生以下等价的标记：

```
<div id="fig:figureRef" class="subfigures">
```

```
![a](image1.png){#fig:figureRefA}
```

```
![b](image2.png){#fig:figureRefB}
```

```
Figure 1: Caption of figure. a — subfigure 1  
↪ caption, b — subfigure 2  
caption
```

```
</div>
```

对子图的引用将呈现为 `figureNumber (subfigureNumber)`，例如，在这个特定的例子中，`[@fig:figureRefA]` 将生成图 [2.1a](#)。

可以将 `notitle` 样式添加到图片中，以完全抑制子图标题。注意，抑制的子图仍然会被计数。

子图网格

如果需要对网格中的子图进行对齐，并且不能使用输出格式样式，则可以使用 `subfigGrid` 选项。此选项将用表格中组织子图。

不同的段落作为表格的行，段落内的子图作为表格的列。列宽度将从相应图像的宽度属性中提取，例如（效果见图 2.2）：

```
::: {#fig:coolFig}
![caption a](images/image1.png){#fig:cfa width=40%}
![caption b](images/image2.png){#fig:cfb width=40%}

![caption c](images/image3.png){#fig:cfc width=40%}
![caption d](images/image4.png){#fig:cfd width=40%}

Cool figure!
:::
```

将产生列宽度为 40% 和 40% 的子图网格。

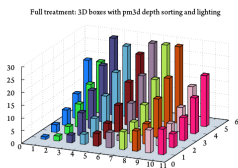
表格宽度计算只考虑图像的第一行，其他行完全忽略。

除了图像之外的任何元素都会被静默地忽略。因此，任何文本、空格、软换行符等都将静默地从输出中消失。当然，这不适用于图片标题段落。

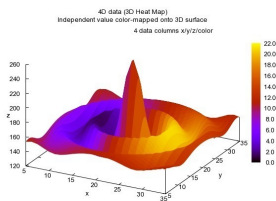
所有图像将自动设置宽度属性为 100%，以便填充整个列。在除 % 之外的任何地方指定宽度都会引发错误。

如果未指定第一行中某些图像的宽度，则这些图像将在剩余空间中均匀地展开。

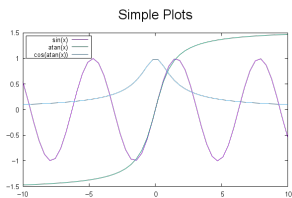
如果没有为第一行中的任何图像指定宽度，那么这些图



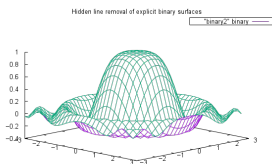
(a) caption a



(b) caption b



(c) caption c



(d) caption d

图 2.2: Cool figure!

像将平均占用页面宽度的 99%（因为 Pandoc 忽略了表的宽度属性）。

在 LaTeX 输出中忽略这个选项，但是段落断行应该产生类似的效果，所以图像应该正确地排版。您不需要启用 `subfigGrid` 就可以使用 LaTeX，但是您仍然可以启用它。

2.20.2 公式

要标记一个跨行公式,请在 `math` 区块之后立即添加 `{#eq` (其中 `label` 需要是一个独一无二的公式标签)，源码如下（效果见公式 2.2）：

```
$$ x=\frac{-b\pm\sqrt{b^2-4ac}}{2a} $$ {#eq:label}
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{2.2}$$

数学块和标签可以用一个或多个空格分隔。

还可以在元数据中设置 `autoEqnLabels: true` 来实现为所有跨行公式编号。但是，请注意，如果没有显式的标签，就无法引用公式。

公式编号将在数学内部排版，前面有 `\qquad`。如果您想使用表格，请在元数据中设置 `tableEqns: true` 选项。根据输出格式的不同，表可能比 `\qquad` 工作得更好，也可能更差。

2.20.3 表格

标记表格请在表格标题的末尾附加 `{#tbl:label}` (`label` 是引用这个表格的惟一标签)。标题和标签必须至少用一个空格隔开。

a	b	c
1	2	3
4	5	6

: Caption `{#tbl:label}`

2.20.4 标题

您还可以标记任何级别的章节。章节标签使用原生 Pandoc 语法，但必须以 `sec:` 开头，例如。

Section `{#sec:section}`

2.20.5 代码块

代码块 ID 应以 `lst:` 开头，例如 `{#lst:label}` `caption` 属性将转换为代码块标题。如果代码块同时有 ID 和 `caption`，将会被编号。

```
```{#lst:code .haskell caption="Listing caption"}
main :: IO ()
```

```
main = putStrLn "Hello World!"
...
```

## 2.20.6 引用

以下都是合法的引用方式：

```
[@fig:label1;@fig:label2;...]
[@eq:label1;@eq:label2;...]
[@tbl:label1;@tbl:label2;...]
@fig:label
@eq:label
@tbl:label
```

引用语法重度参考了参考文献引用（citation）语法。基本的引用是通过 @ 创建的，然后是带有前缀的标签。也可以引用一组对象，方法是将它们放在括号中并且用 ; 作为分隔符。相似的对象将按照它们出现在引文括号中的顺序进行分组，并将序号缩短，如 1, 2, 3 将缩短为 1-3。

您可以将第一个引用字符大写以获得大写前缀，例如 [Fig:label1] 默认将生成 Fig. ... 。除非使用元数据设置覆盖，否则将每个单词的首字母大写的非大写前缀自动派生出大写前缀。有关更多信息参见 [Customization](#)。

### 链接到引用元素

若要使引用带有指向引用元素的超链接，请在元数据中设置 linkReferences: true。这对 L<sup>A</sup>T<sub>E</sub>X 输出没有影响，

因为在  $\text{\LaTeX}$  中，这种引用是 `hyperref` 包处理的。

## 单个引用自定义前缀

可以为每个引用提供自己的前缀。`[Prefix @reference]` 将以前缀逐字替换默认前缀(`fig./sec./` 等),例如 `[Prefix @fig:1]` 将被渲染为 `Prefix 1` 而不是默认的 `fig .1`。

## 不生成前缀

在 `@` 前添加 `-`，就像这样 `[-@citation]`，将禁用默认前缀，例如 `[-@fig:1]` 将只生成 `1`，而没有 `fig.` 前缀。

在分组的引用中，有或没有前缀的引文将在不同的组中。所以 `[-@fig: 1;@fig: 2;-@fig:3]` 将被渲染为 `1, fig. 2, 3`，所以要注意这个特性。同样，非连续序列不会分组在一起。

## 2.20.7 自定义交叉引用配置

使用选项 `--crs` 指定自定义的交叉引用配置，默认为 `crs/zh-CN.yaml`。配置方法请参考 `pandoc-crossref` 文档：<https://lierdakil.github.io/pandoc-crossref>

## 代码 2.10: 多行表格示例

```

┌
↪ -----
Centered Default Right Left
Header Aligned Aligned Aligned

↪ -----
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: 这里是标题

标题也可以是多行 {#tbl:multiline\_tables}

## 代码 2.11: 格框表格示例

: 格框表格示例 {#tbl:grid\_tables}

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

代码 2.12: 格框表格对齐语法

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

代码 2.13: 管线表格示例

```
| Right | Left | Default | Center |
|-----:|:-----|-----|:-----:|
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |
```

: 管线表格示例 {#tbl:pipe\_tables}

代码 2.14: 文件标题区块

```
% title
% author(s) (separated by semicolons)
% date
```

## 代码 2.15: markdown\_in\_html\_blocks 示例

```
<table>
 <tr>
 <td>*one*</td>
 <td>[a link](http://google.com)</td>
 </tr>
</table>
```

## 代码 2.16: markdown\_in\_html\_blocks 转换示例

```
<table>
 <tr>
 <td>one</td>
 <td>a
 ↪ link</td>
 </tr>
</table>
```

## 代码 2.17: 图片属性设置

An inline `![image](foo.jpg){#id .class width=30`  
↪ `height=20px}`  
and a reference `![image][ref]` with attributes.

`[ref]: foo.jpg "optional title" {#id .class key=val`  
↪ `key2="val 2"}`

## 代码 2.18: 脚注语法

Here is a footnote reference, `[^1]` and

↪ `another.[^longnote]`

`[^1]`: Here is the footnote.

`[^longnote]`: Here's one with multiple blocks.

*Subsequent paragraphs are indented to show that*

↪ *they*

belong to the previous footnote.

*{ some.code }*

*The whole paragraph can be indented, or just*

↪ *the first*

*line. In this way, multi-paragraph footnotes*

↪ *work like*

*multi-paragraph list items.*

This paragraph won't be part of the note, because

↪ *it*

isn't indented.



## 第三章 PanBook 扩展

PanBook 扩展主要基于 pandoc 的 [lua filter](#) 功能定制了一些功能或修改了 pandoc 默认行为。本章将依次介绍每个扩展的用途和用法。

### ③ 扩展通用的命令

- 查看扩展列表 `panbook ext -l`
- 查看扩展帮助 `panbook ext -h`
- 查看指定扩展的帮助 `panbook ext -h abstract`

大部分扩展支持用户开启或关闭，但是一般不建议这样做，除非你知道为何要那么做并且知道如何处理编译冲突问题。

## 3.1 abstract

### **i** 扩展信息

**使用场景** 提供摘要环境

**启用状态** 默认在 art, thesis 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** fenced\_divs 参见章节 2.17

示例

todo

## 3.2 column

### **i** 扩展信息

**使用场景** 提供分栏环境

**启用状态** 默认在 book, thesis, art, cv 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** fenced\_divs 参见章节 2.17

# 示例

```
::::::::::::::::: {.columns}
```

```
::: {.column width="56%"}

听人家背地里谈论，孔乙己原来也读过书，但终于没有进学，又不会营
→ 生；于是愈过愈穷，弄到将要讨饭了。幸而写得一笔好字，便替
→ 人家钞钞书，换一碗饭吃。可惜他又有一样坏脾气，便是好吃懒
→ 做。坐不到几天，便连人和书籍纸张笔砚，一齐失踪。如是几次，
→ 叫他钞书的人也没有了。孔乙己没有法，便免不了偶然做些偷窃
→ 的事。但他在我们店里，品行却比别人都好，就是从不拖欠；虽
→ 然间或没有现钱，暂时记在粉板上，但不出一月，定然还清，从
→ 粉板上拭去了孔乙己的名字。

:::

::: {.column width="40%"}

孔乙己喝过半碗酒，涨红的脸色渐渐复了原，旁人便又问道，“孔乙己，
→ 你当真认识字么？”孔乙己看着问他的人，显出不屑置辩的神气。
→ 他们便接着说道，“你怎的连半个秀才也捞不到呢？”孔乙己立刻
→ 显出颓唐不安模样，脸上笼上了一层灰色，嘴里说些话；这回可
→ 是全是之乎者也之类，一些不懂了。在这时候，众人也都哄笑起
→ 来：店内外充满了快活的空气。

:::

:::::::::::::::::
```

显示效果:

听人家背地里谈论，孔乙己原来也读过书，但终于没有进学，又不会营生；于是愈过愈穷，弄到将要讨饭了。幸而写得一笔好字，便替人家钞钞书，换一碗饭吃。可惜他又有一样坏脾气，便是好吃懒做。坐不到几天，便连人和书籍纸张笔砚，一齐失踪。如是几次，叫他钞书的人也没有了。孔乙己没有法，便免不了偶然做些偷窃的事。但他在我们店里，品行却比别人都好，就是从不拖欠；虽然间或没有现钱，暂时记在粉板上，但不出一月，定然还清，从粉板上拭去了孔乙己的名字。

孔乙己喝过半碗酒，涨红的脸色渐渐复了原，旁人便又问道，“孔乙己，你当真认识字么？”孔乙己看着问他的人，显出不屑置辩的神气。他们便接着说道，“你怎的连半个秀才也捞不到呢？”孔乙己立刻显出颓唐不安模样，脸上笼上了一层灰色，嘴里说些话；这回可是全是之乎者也之

## 3.3 copyright

### ❗ 扩展信息

**使用场景** 提供版权声明页

**启用状态** 默认在 book 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 无。需要模板支持。

### 示例

需要模板支持，具体请看模板帮助。

```
$ panbook book -V copyright:ccnd
```

## 3.4 coverletter

### ❗ 扩展信息

**使用场景** 提供求职信功能

**启用状态** 默认在 cv 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** fenced\_divs 参见章节 [2.17](#)

## 示例

```
韩荆州 {.letter company=" 公司" addr=" 地址" city="
↪ 城市"} // 接收方，全部必须
开元 22 年 {.letter .date}
↪ // 日期，可选
Dear Sir or Madam, {.letter .opening}
↪ // 称呼，必须
Yours faithfully, {.letter .closing}
↪ // 祝颂语，必须
curriculum vit\ae {.letter .enclosure}
↪ enclosure=" 附件" // 附件，可选
```

正文

## 3.5 device

### 扩展信息

**使用场景** 适配不同设备 (pc, mobile, kindle)

**启用状态** 默认在 book 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 无。需要模板支持，通过命令行参数传递。

```
$ panbook book -V device:mobile
```

## 3.6 fontawesome5

### **i** 扩展信息

**使用场景** 修复 moderncv 中 fontawesome5 和 fontawesome 冲突

**启用状态** 只对 cv 的 moderncv 模板有效

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 无

## 3.7 grade

### **i** 扩展信息

**使用场景** 打分命令，适用于简历

**启用状态** 默认在 cv 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

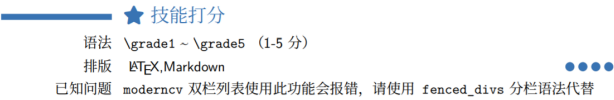
**语法系列** raw latex

# 示例


```
\faStar~ 技能打分

- [语法]{.cat} `grade1` ~ `grade5` (1-5 分)
- [排版]{.cat} \LaTeX,Markdown
↪ [\grade4]{.comment}
- [已知问题]{.cat} moderncv 双栏列表使用此功能会报错, 请
↪ 使用 fenced_divs 分栏语法代替
```

## 效果



## 3.8 header

 扩展信息

**使用场景** 面向开发者, 被大部分其他扩展依赖。提供将 tex 文件加入 metadata 区块的 header-includes 字段功能

**启用状态** 默认在所有模块中启用

**格式支持**  $\LaTeX$

**语法系列** 无



## 3.9 lineblock

### **i** 扩展信息

**使用场景** 提供 lineblock 环境，用于修复 lineblock 缩进问题。lineblock 请看章节 2.5

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 原生 pandoc's Markdown

## 3.10 listings

### **i** 扩展信息

**使用场景** 提供 listings 及 pandoc 原生代码环境样式定义。为 pandoc 原生代码块环境实现阴影和自动换行。

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 无

### 示例

此扩展提供的 listings 样式如下：

```

\lstset{
 basicstyle=\small\linespread{1}\ttfamily,
 keywordstyle=\color[rgb]{0.13,0.29,0.53}
↪ \textbf,
 commentstyle=\color{gray},
 identifierstyle=\color[rgb]{0.00,0.00,0.00},
 stringstyle=\color[rgb]{0.31,0.60,0.02},
 frame=shadowbox,
 rulesepcolor=\color{red!20!green!20!blue!20},
 backgroundcolor=\color[rgb]{0.97,0.97,0.97},
 tabsize=4,
 breaklines=tr,
 showstringspaces=false,
}

```

用户可通过命令行参数替换为自己的样式：

```

$ panbook ext -h listings
 -G ext-listings:<true|false>
↪ enable listings(default true)
 -G ext-listings-lstset:<custom listings set
↪ file> change listings set file

```

## 3.11 longtable

### **i** 扩展信息

**使用场景** 用于修复 longtable [bug](#)（不完全修复）

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 无

## 3.12 tabu

### **i** 扩展信息

**使用场景** 由于 longtable 有难以修复的 [bug](#), 使用 tabu 替代 longtable

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** 自定义

### 示例

默认使用 tabu, 除非以下两种情况。

表格标题开头使用 `{.longtable}` 表示使用 pandoc 原生表格处理方式:

```
a | b
--|--
c | d
```

: `{.longtable}` Use pandoc longtable

表格标题开头使用 `{.longtabu}` 表示使用 longtabu 处理换页表格：

```
a | b
--|--
c | d
```

: `{.longtabu}` Use longtabu

## 3.13 theorem

### 扩展信息

**使用场景** 提供定理环境

**启用状态** 默认在 book, thesis 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** fenced\_divs 参见章节 [2.17](#)

### 示例

definition, theorem, lemma, corollary, proposition  
支持引用, ID 前缀必须分别是 def:, thm:, lem:, cor:, pro:。

效果见 `[@def:test]`, `[@thm:test]`, `[@lem:test]`,  
↪ `[@cor:test]`, `[@pro:test]`。

`::: {#def:test .definition caption=" 标题"}`

Definition

`:::`

`::: {#thm:test .theorem caption=" 标题"}`

Theorem

`:::`

`::: {#lem:test .lemma caption=" 标题"}`

Lemma

`:::`

`::: {#cor:test .corollary caption=" 标题"}`

Corollary

`:::`

`::: {#pro:test .proposition caption=" 标题"}`

Proposition

`:::`

`::: {.example}`

Example

`:::`

`::: {.exercise}`

Exercise

`:::`

::: {.problem}

Problem

:::

::: {.proof}

Proof

:::

::: {.note}

Note

:::

::: {.conclusion}

Conclusion

:::

::: {.assumption}

Assumption

:::

::: {.property}

Property

:::

::: {.remark}

Remark

:::

::: {.solution}

效果见 定义 3.1, 定理 3.1, 引理 3.1, 推论 3.1, 命题 3.1。

定义 3.1. 标题

Definition



定理 3.1. 标题

Theorem



引理 3.1. 标题

Lemma



推论 3.1. 标题

Corollary



命题 3.1. 标题

Proposition



例 3.1 Example

练习 3.1 Exercise

例题 3.1 Problem

证明 Proof

**注意** *Note*

**结论** *Conclusion*

**假设** *Assumption*

**性质** *Property*

**注** *Remark*

**解** *Solution*

## 3.14 wrap

### 扩展信息

**使用场景** 提供文本框环境

**启用状态** 默认在 book, art, thesis, cv 中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** fenced\_divs 参见章节 [2.17](#)

### 示例

```
::: {.help}
```

```
Help
```

```
:::
```

```
::: {.info caption=" 自定义标题"}
```

```
Info
```

```
:::
```



```
::: {.warn}
```

Warn

```
:::
```

```
::: {.alert}
```

Alert

```
:::
```

```
::: {.tip}
```

Tip

```
:::
```

introduction (章节内容提要) 和 problemset (习题) 环境必须  
↔ 使用列表

```
::: {.introduction}
```

– Introduction

```
:::
```

```
::: {.problemset}
```

– Problemset

```
:::
```

```
::: {.solu}
```

解题环境，默认整体缩进 2em，可通过 caption 自定义缩进

```
:::
```

效果:

## ② 帮助

Help

## ❶ 自定义标题

Info

## ⚠ 注意

Warn

## ⊗ 警告

Alert

## 👉 提示

Tip

introduction（章节内容提要）和 problemset（习题）环境必须使用列表

### 内容提要

❏ **Introduction**

## 📖 第 3 章 习题 📖

### 1. Problemset

## **i** 更改 Wrap 风格

可以通过命令行参数指定 Wrap 风格 (`-G ext-wrap-style:<bclogo|tcolorbox>`), 默认为 `tcolorbox`。也可以通过 `-G ext-wrap-tex:<custom wrap set file>` 自定义风格

## 3.15 zh\_en

### **i** 扩展信息

**使用场景** 主要用于中英文翻译, 可选择隐藏中文或英文, 或对照显示。

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X

**语法系列** `fenced_divs` 参见章节 [2.17](#)

### 示例

```
::: {.zh}
```

有几回, 邻舍孩子听得笑声, 也赶热闹, 围住了孔乙己。他便给他们茴

↪ 香豆吃, 一人一颗。孩子吃完豆, 仍然不散, 眼睛都望着碟子。

↪ 孔乙己着了慌, 伸开五指将碟子罩住, 弯腰下去说道, “不多了,

↪ 我已经不多了。”直起身又看一看豆, 自己摇头说, “不多不多!

↪ 多乎哉? 不多也。”于是这一群孩子都在笑声里走散了。

```
:::
```

```
::: {.en}
```

```
A few times, the children in the neighbourhood
↪ heard laughter and also rushed around, and
↪ surrounded Kong Yiji. He gave them fennel
↪ beans, one for each. The child finished the
↪ beans, still not scattered, eyes are looking at
↪ the dishes. Kong Yi himself panicked, stretched
↪ out the five fingers to cover the plate, and
↪ bent down and said, "Not much, I am not much."
↪ Straight up and look at the beans, shaking his
↪ head and saying, "Not much! 哉 哉? Not much." So
↪ this group of children are gone in laughter.
:::
```

默认只显示中文，以上代码显示效果如下：

有几回，邻舍孩子听得笑声，也赶热闹，围住了孔乙己。他便给他们茴香豆吃，一人一颗。孩子吃完豆，仍然不散，眼睛都望着碟子。孔乙己着了慌，伸开五指将碟子罩住，弯腰下去说道，“不多了，我已经不多了。”直起身又看一看豆，自己摇头说，“不多不多！多乎哉？不多也。”于是这一群孩子都在笑声里走散了。

通过命令行改变显示语言：

```
$ panbook ext -h zh_en
-G ext-zh_en:<true|false>
↪ enable zh_en(default true)
-G ext-zh_en-tex:<custom zh_en set file>
↪ change zh_en set file
```

```
-G ext-zh_en-texlang:<custom language
↔ file> change language file
-G ext-zh_en-lang:<zh|en|both>
↔ select language(default zh)
```

## 3.16 plot

### **i** 扩展信息

**使用场景** 使用 graphviz, ditaa 等图形描述语言画图

**启用状态** 默认在所有模块中启用

**格式支持** L<sup>A</sup>T<sub>E</sub>X, EPUB

**语法系列** fenced\_divs 参见章节 2.17

### 3.16.1 安装

需要安装 librsvg 用于 SVG 转 PDF。msys2 上使用 pacman  
-S mingw64/mingw-w64-x86\_64-librsvg 安装。

目前支持以下画图引擎，如需使用，请安装对应软件并  
加入 PATH 环境变量

- dot,neato,fdp,sfdp,twopi,circo (from graphviz)
- ditaa (建议使用 go 语言版本 [https://github.com/akavel/d](https://github.com/akavel/ditaa))
- goseq (<https://github.com/pandoc-ebook/goseq/releases>)
- a2s (<https://github.com/pandoc-ebook/asciitosvg/releases>)
- gnuplot

- abc (安装 abcm2ps, 可以在 <http://abcplus.sourceforge.net> 下载 windows 和 linux 的二进制文件)

## 3.16.2 用法

使用代码块语法

- 为正常使用交叉引用, id 格式须符合 `fig:<label>`
- 样式格式须为 `plot:<plot engine>`, 比如 `plot:dot`
- 其他样式需在 `plot:<plot engine>` 之后
- 可像图片语法 (见章节 2.16) 一样设置 `key=val` 格式属性, 比如 `width=50%`
- 可以使用 `pandoc-crossref` 的子图语法 (见章节 2.20), 区别是图片代码块必须有 `subfig` 属性来给予图分组, 格式 `subfig=group number`, 分组数字相同的将在同一行显示。

```
```${#fig:<label> .plot:<engine> .class
↪  caption="<caption>" key=val}
digraph G {
    Markdown -> LaTeX;
}
...`
```

3.16.3 graphviz

dot 代码如下, 图片见图 3.1。

```

```{#fig:plot_dot .plot:dot caption="Dot 示例"}
digraph {
 "A" [shape="circle"];
 "B" [shape="rectangle"];
 "C" [shape="diamond"];

 "A" -> "B" [label="A to B"];
 "B" -> "C" [label="B to C"];
 "A" -> "C" [label="A to C"];
}
```

```

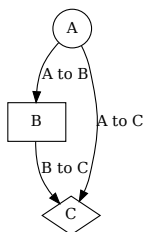


图 3.1: Dot 示例

3.16.4 ditaa

ditaa 代码如下，图片见图 3.2。

```

```{#fig:plot_ditaa .plot:ditaa caption="Ditaa 示
↪ 例"}

+-----+ +-----+ +-----+
| +=--+ ditaa +--> | | | | |
| Text | | | |diagram|
|Document| |!magic!| | |
| {d}| | | | |
+---+---+ +-----+ +-----+

 : ^
 | Lots of work |
 +-----+

...

```

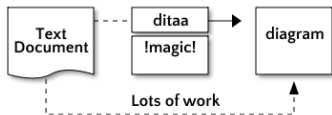


图 3.2: Ditaa 示例

### 3.16.5 goseq 时序图

goseq 代码如下,图片见图 3.3。更多信息请参考:<https://github.com/robert-i/knitr/blob/master/inst/figs/goseq.R>

```

```{#fig:plot_goseq .plot:goseq caption="Goseq
↪ Demo" width=50%}

Client->Server: Make request
Server->Database: Make database\nrequest

```


Database->Server: The result
Server->Client: The response
...

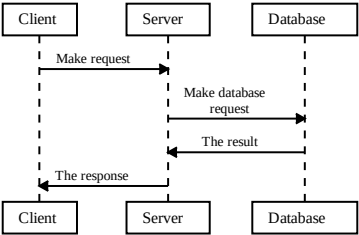


图 3.3: Goseq Demo

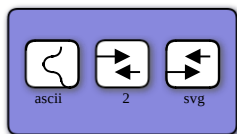
3.16.6 asciitosvg

asciitosvg 代码如下，图片见图 3.4。更多信息请参考 <https://github.com/asciitosvg/asciitosvg>。

```
```{#fig:plot_a2s .plot:a2s caption="A2s Demo"}
.------.
|
| .---. .---. .---. |
| | .- | +--> | | <-- | |
| | '- ' | | <-- | +--> | |
| '---'-' '-----' '-----' |
| ascii 2 svg |
|
|'-----'
```

<https://github.com/asciitosvg>

```
[1,0]: {"fill":"#88d","a2s:delref":1}
...
```



<https://github.com/asciitosvg>

图 3.4: A2s Demo

### 3.16.7 gnuplot

gnuplot 代码如下，效果见图 3.5。

```
```{#fig:plot_gnuplot .plot:gnuplot  
↪ caption="gnuplot 示例"}  
set terminal pngcairo background "#ffffff"  
↪ enhanced font "arial,8" fontscale 1.0 size 540,  
↪ 384  
set output 'hidden2.1.png'  
set isosamples 25,25  
set xyplane at 0  
unset key
```

```

set palette rgbformulae 31,-11,32
set style fill solid 0.5
set cbrange [-1:1]

set title " 中文 Mixing pm3d surfaces with
↪ hidden-line plots"

f(x,y) = sin(-sqrt((x+5)**2+(y-7)**2)*0.5)

set hidden3d front
splot f(x,y) with pm3d, x*x-y*y with lines lc rgb
↪ "black"
...

```

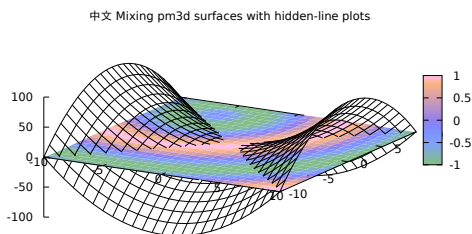


图 3.5: gnuplot 示例

使用 gnuplot 绘制数据图时,建议将数据文件放在 `src/data` 目录下, 示例如下, 效果见图 3.6。

```

```{#fig:plot_gnuplot2 .plot:gnuplot
↪ caption="gnuplot 绘制数据图"}
set datafile separator comma
set title 'Browser popularity'
set xlabel 'Year'
set ylabel '% usage'
set style data histogram
set style histogram clustered gap 1
set style fill solid 1 noborder
set xtics scale 0
plot for [i=2:5] 'data/usage.csv' using i:xtic(1)
↪ title columnheader
...

```

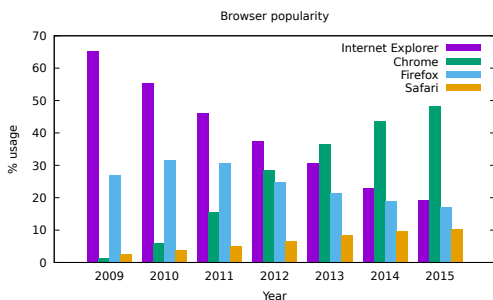


图 3.6: gnuplot 绘制数据图

### 3.16.8 Asymptote

示例如下，效果见图 3.7。

```

```{#fig:plot_asy .plot:asy caption="Asymptote 示例"
↪ width=40%}

import math;
import graph;
size(0,400);

real f(real t) {return 2*cos(t);}
pair F(real x) {return (x,f(x));}

draw(polargraph(f,0,pi,operator ..));

defaultpen(fontsize(20pt));

xaxis("$x$");
yaxis("$y$");

real theta=radians(50);
real r=f(theta);

↪ draw("$\theta$",arc((0,0),0.5,0,degrees(theta)),

pair z=polar(r,theta);
draw(z--(z.x,0),dotted+red);
draw((0,0)--(z.x,0),dotted+red);
label("$r\cos\theta$", (0.5*z.x,0),0.5*S,red);
label("$r\sin\theta$", (z.x,0.5*z.y),0.5*E,red);
dot("$ (x,y) $",z,N);

↪ draw("r", (0,0)--z,0.5*unit(z)*I,blue,Arrow,DotMa

```

```
dot("$ (a, \theta) $", (1, \theta), NE);
dot("$ (2a, \theta) $", (2, \theta), NE);
...
```

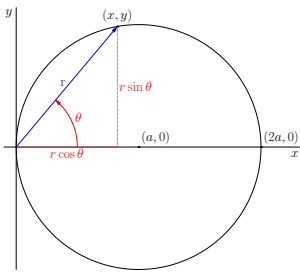


图 3.7: Asymptote 示例

3.16.9 乐谱 (abc notation)

示例如下，效果见图 3.8。

```
```{#fig:plot_abc .plot:abc caption="ABC notation"}
X:1
T:Speed the Plough
M:4/4
C:Trad.
K:G
|:GABc dedB|dedB dedB|c2ec B2dB|c2A2 A2BA|
 GABc dedB|dedB dedB|c2ec B2dB|A2F2 G4:|
|:g2gf gdBd|g2f2 e2d2|c2ec B2dB|c2A2 A2df|
 g2gf g2Bd|g2f2 e2d2|c2ec B2dB|A2F2 G4:|
...
```
```



图 3.8: ABC notation

3.16.10 tikz

示例如下，效果见图 3.9。如需引入自定义包，请使用 `usepackage` 属性。

```

```\#fig:plot_tikz .plot:tikz caption="TIKZ 示例"
↪ usepackage="\usepackage{ctex}"
\begin{tikzpicture}
 \draw (0,0)--(0,4);
 \draw (0,0)--(4,0);
 \draw (4,0) arc[start angle=0,end
↪ angle=90,radius=4cm];
\end{tikzpicture}
...
```

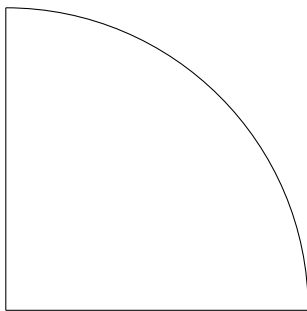


图 3.9: TIKZ 示例

### 3.16.11 子图

代码如下，效果见图 3.10。

```

::: {#fig:plot_subfig}
```${#fig:sub_gnuplot .plot:gnuplot caption="gnuplot
↪ 示例" width=48% subfig=1}
set terminal pngcairo background "#ffffff"
↪ enhanced font "arial,8" fontscale 1.0 size 540,
↪ 384
set output 'hidden2.1.png'
set isosamples 25,25
set xyplane at 0
unset key

set palette rgbformulae 31,-11,32
set style fill solid 0.5
set cbrange [-1:1]

set title " 中文 Mixing pm3d surfaces with
↪ hidden-line plots"

```



```

f(x,y) = sin(-sqrt((x+5)**2+(y-7)**2)*0.5)

set hidden3d front
plot f(x,y) with pm3d, x*x-y*y with lines lc rgb
↪ "black"
...

```{#fig:sub_gnuplot2 .plot:gnuplot
↪ caption="gnuplot 绘制数据图" width=48% subfig=1}
set datafile separator comma
set title 'Browser popularity'
set xlabel 'Year'
set ylabel '% usage'
set style data histogram
set style histogram clustered gap 1
set style fill solid 1 noborder
set xtics scale 0
plot for [i=2:5] 'data/usage.csv' using i:xtic(1)
↪ title columnheader
...

```

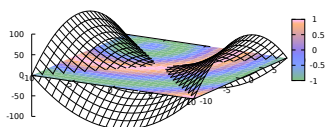
子图示例

```

:::

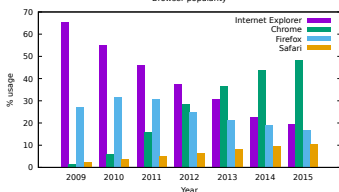
```

中文 Mixing pm3d surfaces with hidden-line plots



(a) gnuplot 示例

Browser popularity



(b) gnuplot 绘制数据图

图 3.10: 子图示例

## 3.17 ipa

### 扩展信息

**使用场景** IPA 是 International Phonetic Alphabet 的缩写。此扩展用于输入国际音标

**启用状态** 默认在所有模块中启用。个别 beamer 主题可能有冲突

**格式支持** L<sup>A</sup>T<sub>E</sub>X, EPUB

**语法系列** Spans

### 示例

Linux `[/'laɪnəks; 'lɪnəks/]{.ipa}`

效果

Linux /'laɪnəks; 'lɪnəks/

# 第四章 开发

PanBook 处理流程如图 4.1。

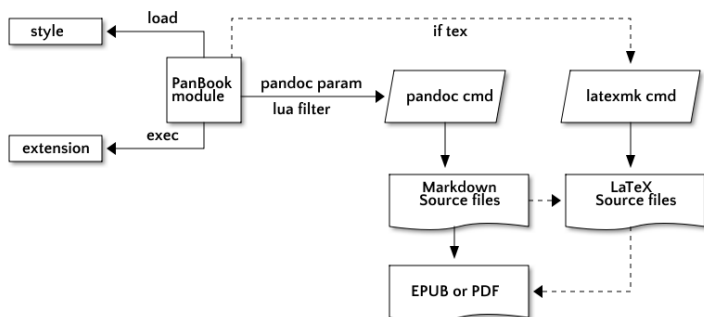


图 4.1: PanBook 流程图

PanBook 基于 Pandoc，首先要了解 Pandoc 的使用方法。PanBook 主体使用 Shell 脚本开发，Pandoc Lua filter 使用 Lua 开发，定制 style 需要了解  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  或者 CSS。表 4.1 给出了名词定义。

Panbook 的基本思路是，定义一些全局数组变量存储 Pandoc 参数信息，允许模块，风格和扩展根据输出目标情

表 4.1: PanBook 名词定义

名词	目录	功能
module	modules/	模块, book,cv,slide,thesis 都是模块
style	styles/	风格, -style 参数指定, 可以定制某一种文档类或者文档格式 (比如 Epub)
extension	extensions/	扩展, 使用 lua filter 或者基于 header-includes 提供某些可被复用的扩展功能
template	templates/	Pandoc 模板, 基于格式的模板, 对应 Pandoc 的 -template 参数

况及用户提供的参数修改这些变量, 最终拼接成 Pandoc 参数, 执行 Pandoc 转换。开发工作也主要是做风格和扩展, 去适配不同的文档类。全局变量定义见代码 4.1。

## 4.1 Pandoc 开发基础

## 4.2 模块开发

以 book 模块代码为例, 讲解模块开发流程。

## 代码 4.1: Panbook 全局变量

```
declare -A _G # panbook 全局选项
declare -A _P # pandoc 选项
declare -A _V # pandoc var (-V)
declare -A _M # pandoc meta (-M)
declare -A _F # lua filter (--lua-filter)
declare -A _FO # 需要在 citeproc 前执行的 lua
↪ filter
declare -A _MF # parse metadata.yml
declare -A _H # 帮助信息
```

## 4.3 风格开发

以简历 *moderncv* 风格为例，讲解风格开发流程。

## 4.4 扩展

以 *wrap* 扩展为例，讲解扩展开发流程。



# 参考文献

- [1] TZENGYUXIO. Pandoc's Markdown 語法中文翻譯 [M]. <http://pages.tzengyuxio.me/pandoc/>.
- [2] MACFARLANE J. Pandoc User Manual[J]. <https://pandoc.org/MANUAL.html>, 2019.
- [3] KING H. Test Dummy Book[M]. China, 1321.

