

Typesetting Tutorial for L^AT_EX

L^AT_EX 排版教程

——排版，只需要一个浏览器

BY 兔子草

v1.1

前言

本文为准备制作书籍类同人本的玩家撰写。

最低配置：一个浏览器。

推荐配置：一个电脑上的浏览器。

考虑了等宽文字与不等宽文字（通常就是中英文）掺杂的情况。有少量场景与操作系统有关，暂只包括 Windows 下的处理方案。

推荐新手使用在线工具，因此文中不包含 L^AT_EX 安装等内容。如果需要在本地运行的话，这部分教程网上已经很多，可以自行寻找。

我创立了一个模板，欢迎直接[下载使用](#)¹。如果本文或者本模板对你出本有帮助，你感谢我一句也挺好的。

本教程包含一些高投入低回报的内容，标注了“(进阶)”的字样。



by 兔子草

🐾 3440950898

¹下载使用：<https://github.com/zhuty18/fanfiction-sample/releases/latest>

目 录

前言	ii
----	----

目录	iii
----	-----

I L ^A T _E X 的优点	I
---------------------------------------	---

§ 1 简单易上手	I
§ 2 纯粹的排版工具	I
§ 3 创作与排版的分离	I
§ 4 编辑与渲染的分离	2
§ 5 默认细节极佳	2
§ 6 一句忠告	3

II 排印学知识	4
----------	---

§ 1 小说	4
§ 2 全网格化 (进阶)	4

III 从文本到文件	6
------------	---

§ 1 L ^A T _E X 正文语法入门	6
i 文件结构	6
ii 正文	6
iii 特殊字符	7
iv 标题	7
§ 2 格式转换	7

IV 建立文档	9
---------	---

§ 1 文档类型和主字号	9
i 中文处理: ctex 包	9
ii 字号命令使用	10
§ 2 纸张布局	10
i 出血	12
§ 3 各级标题	12
§ 4 标题格式	13

V 字体设定	14
--------	----

§ 1 字体入门	14
i 字体类别	14
ii 字体风格	14

iii 字符集和编码表	14
-------------	----

§ 2 ctex 中的字体	15
---------------	----

§ 3 常用字体获取	16
------------	----

§ 4 字体配置	17
----------	----

i 路径法	17
-------	----

ii 安装法	17
--------	----

§ 5 字体名称	17
----------	----

i 字体名称查询	17
----------	----

ii 配置的字体找不到?	19
--------------	----

iii fontspec 包字体配置	19
--------------------	----

iv ctex 包字体配置	20
---------------	----

v 外语连字 (进阶)	21
-------------	----

§ 6 高级配置	22
----------	----

i pdf 阅读器	22
-----------	----

ii 中文	23
-------	----

iii 外语	23
--------	----

VI 页面排版	25
---------	----

§ 1 行高和段距	25
-----------	----

i 行高	25
------	----

ii 段距	25
-------	----

§ 2 缩进	26
--------	----

§ 3 换页	26
--------	----

§ 4 新章节前的空白页	26
--------------	----

§ 5 手动空白页	27
-----------	----

§ 6 页码更改	27
----------	----

§ 7 孤字和孤行 (进阶)	27
----------------	----

i 手动	28
------	----

ii 自动	28
-------	----

iii 半自动	28
---------	----

VII 引用	29
--------	----

§ 1 段落格式	29
----------	----

§ 2 文本框	29
---------	----

VIII 注释	30
---------	----

§ 1 脚注	30
--------	----

§ 2 尾注	30
--------	----

IX	目录	32			
§ 1	目录标题	32		iii	功能欠缺
§ 2	目录配置	32		§ 3	为什么不用 INDESIGN?
§ 3	局部目录	33		§ 4	LaTeX 怎么念?
§ 4	文档内跳转	33		§ 5	补充阅读
§ 5	目录换页	34			
X	页眉页脚	35			
§ 1	页面风格定义	35			
§ 2	文档自动的标记	35			
§ 3	页面风格使用	36			
§ 4	问题处理	36			
XI	图片	38			
§ 1	插入图片	38			
§ 2	图片标题	38			
§ 3	旋转	39			
XII	其它	40			
§ 1	规定标点宽度	40			
§ 2	特殊字符换字体	40			
§ 3	行溢出	40			
§ 4	纸面不充盈	41			
§ 5	扉页	41			
§ 6	子文件	41			
§ 7	更多文字装饰	42			
§ 8	色彩	43			
§ 9	输入 EMOJI	43			
i	单色 emoji	43			
ii	多色 emoji	44			
XIII	严格网格排版 (进阶)	45			
§ 1	行高段距设定	45			
§ 2	版心规划	45			
§ 3	正文对齐	46			
XIV	篇末谈	47			
§ 1	为什么写这篇文章?	47			
§ 2	为什么不用 TYPST?	47			
i	中文处理能力不足	47			
ii	格式转换路径空缺	48			

I L^AT_EX 的优点

一般我们看到讲 L^AT_EX 的文章，总难免骂两句这东西磨磨唧唧的渲染、乱七八糟的规范和屎山一样的包。正因如此，无数人试图取代 L^AT_EX，但它目前依然是唯一可靠的学术类排版工具，也是世界上最流行的排版系统之一，这足以证明它的独特价值。

这份价值在于，外行人想要排出一个“能看的印刷品”，L^AT_EX 是最快也最简单的解决方案。

§1 简单易上手

*Overleaf*¹ 是 L^AT_EX 的在线 IDE，全网页端，只需注册一个账号就可免费使用。使用时，在工作区左侧先点击目录 (Menu)，将编译引擎换成 X_YL^AT_EX (XeLaTeX) 或 LuaL^AT_EX (LuaLaTeX)，才可正常处理中文。拼写检查系统不支持中文，建议关掉，不然中文全篇都是拼写警告。

截止到写这篇文档的 2026 年 8 月 29 日，*Overleaf* 为免费用户提供 10 秒的编译时间，不放图基本够用了。(编译一个 370 页，20 万字的本大概需要 10 秒)。

不过，如果想自定义字体的话，*Overleaf* 相对麻烦，需要将字体文件上传后调用。安装 *TeXLive*²，在本地进行工作，或许更便捷一些。

我的模板传到了 *Overleaf* 上，但是公开模板需要等待审核检查，等通过了我会在这里放出链接。

§2 纯粹的排版工具

目前的 DIY 出本教程里面，排版这一步都是用 *InDesign*。但 *InDesign* 本质上是版式设计软件，面向市场是杂志、画册、报纸等类型。目标情景是一页上要分若干区，有的区填字、有的区放图，设计者主要研究这些区域如何分配面积、如何摆放、如何提升信息传达率、如何保持美观。

那么问题来了，你真的需要这些功能吗？还是说，其实你需要的只是一个“能把我想印的那点东西漂亮又简单地打包成印刷用 pdf”的工具呢？

L^AT_EX 是一个简单粗暴的排版软件。格式越简单、越统一，用它来做就越容易。

排个书、插个图、把一页页画好的漫画封装成指定纸张大小的 pdf……放在设计业内属于绝对的轻度需求，而 T_EX 就是为这些轻度需求而生的。它的目的不是把二十页纸排出花来，而是让八百页的文档看源文件就知道格式不会出错。

§3 创作与排版的分离

L^AT_EX 里，创作和排版天然分离，十分契合同人创作的规律。

¹Overleaf: <https://cn.overleaf.com/>

²TeXLive: L^AT_EX 的编译器，仅能通过命令行使用，建议搭配 *VSCode* 或 *TeXstudio* 等工具。

出同人本，无一例外是拿着内容补格式。同人文的传播载体总是朴实无华的 txt，偶有文画双修的大佬，也基本都是图文分离式发放。这样的创作基础天然就适配 LaTeX。

LaTeX 的使用步骤分为两步，先用文本编辑器撰写工作文件，再用引擎编译出结果 pdf。在已有内容的情况下，只需要给内容文件增加简单的标记（基本也就是各级标题，有些作者会有脚注尾注，最多再加粗斜体），通过简单的代码结构设计实现文档内容与格式的解耦，就能令工作空间简化且专注，只需关注排版的各项参数。

它有一点尤为可贵：手持一份十万字的文稿，正常人的直觉就是把它一次排完，然后对着 pdf 挑问题，一个点一个点修。而排版工具里，只有 LaTeX 在一排几百页的时候不会挑你的机能。它只会慢，绝不会崩。

§4 编辑与渲染的分离

排版的本质是把文档变成一堆矢量图。这个步骤中最本质的是编辑（设定格式即转化规则），而最耗时的是渲染。

作为一个没有图形化操作界面的排版工具，LaTeX 渲染时可以把所有资源全部用在“得到 pdf”这一件事上，效率极高。如果你嫌 Windows 还是太慢，可以装一个 WSL，再提一波生产力。

§5 默认细节极佳

如果找一些专业的中文排版攻略读（比如中文排版需求¹，中文排版网格系统的五大迷思²），那么就会意识到，文字排版这件事其实非常细碎繁杂，需要关注的细节极多。

由于开源的特性，LaTeX 内有着大量前人造好的轮子，可以覆盖排版时应当处理的普适性专业细节，极大地节约工作成本。用户只需要进行自己的个性化即可。对于中文，最显著的案例就是标点压缩和字距。

• 标点压缩

在正式的排版原则中，使用全角标点时标点的宽度是会变的。较为常见的几个状态如下

1. 独立标点为 1 个字符宽
2. 连续的两个标点（如：“”）合并挤压为 1.5 个字符宽
3. 段首的起始标点（如“、《等）为 0.5 个字符宽

• 字距

字距指文字之间的水平间距。一份理想的中文文本，应该满足以下条件：

1. 行宽可以规定
2. 文字每行左右均顶格，恰好占满设定的行宽
3. 当汉字与外语字符/阿拉伯数字相邻时，间距一个空格宽
4. 在没有干扰的情况下，文字竖对齐。主要是最后一行，如果前文的字距不为默认值，那么最后一行的字距应与前文相同，而非简单左靠

¹中文排版需求：<https://www.w3.org/TR/clreq/>

²中文排版网格系统的五大迷思：<https://www.thetype.com/2020/01/16565/>

这些细碎的设定，在 L^AT_EX 的 `ctex` 包中默认就会满足。不需要自己进行任何选择或调整 `ctex` 包默认就提供最美观的排版。（破折号略有些问题，我在后文里附了解决方案）

字母类文字排版时会遇到的关键问题是，一行文字几乎不可能在换行处恰好断开。T_EX 对此的默认处理方案可谓是视觉上最美观的。*Knuth-Plass* 算法通过断字和拉伸，使得一段文本每行左右两侧都能对齐，且肉眼基本无法看出每行空格宽度的区别。

正因有如此基础，L^AT_EX 处理混合文本时的便捷程度天下无双。

§6 一句忠告

L^AT_EX 是很容易催生强迫症的，千万不要追求完美。谨记，你的时间其实也是成本。

II 排印学知识

§1 小说

版式设计是为了优化信息的传达。对于小说，最重要的信息永远是故事本身，即文字内容。因此，没有必要太深入地研究版面设计。但完全不管也不行，“易读性”和“可读性”是必须要考虑的。

易读性即 Legibility，对每个基础单元（中文的字、英文的单词）的识别程度，更多地指向字体。可读性即 Readability，即阅读体验的舒适性。在排版中，可读性代表一段文本的阅读容易度，主要指向基础单元的排布方式，例如字距、行高。

这都是排印学里非常根本的课题。有太多的人已经研究过了，我们可以直接使用现成的结论：

行高：中文的行高一般至少是字高的 1.5~1.8 倍，具体数值受字体和字号的双重影响。如果行高不足，汉字等高的特性会使得行与行之间空隙不清晰，降低阅读流畅性。

行数：翻页的频次主要是行数决定的，这个数极大地影响阅读体验。对于 A5 大小的纸张，可以以 25 为参照，每页行数小于等于 25 时，阅读体验比较休闲，反之行数越多，越接近于专业性强的书籍，阅读体验越严肃。实际使用中，23~27 行适用于小说类文章排版。

单行字数：A5 这个纸张大小还没有充分利用人眼的视野空间，因此单行字数只受限于字体大小和文字区域的宽度，不需要考虑人眼阅读能力而额外分栏。通常的书籍是每行 28~30，实际体验上，26~32 都是可接受的。

具体选择什么样的参数，建议按自己的需求来决定。决定一个本子气质的是单页信息量（可以简单用行数 * 单行字数来量化）。同字号下，单页信息量越大，阅读体验越严肃；越小体验越活泼。比方说，一行 18 个字，2 倍行高就可以轻松打造移动端的阅读体验。

但请注意，排版越疏松，本子越厚。本子越厚，成本越高。印厂算钱的时候只看页数，不看油墨密度，页数越多就越贵。并且，对于字数较多的本，不紧凑一点真的会印成砖头的。

一个更直观的算法：

每页行数 * 每行字数得到一页纸理论上能印的字数，但实际上不可能印满（出版物上标注的字数是版面字数，约等于行数 * 每行字数 * 页数，所以给人的感觉很注水）。实际每页平均能印下的字数约等于理论值乘上 0.6~0.65（排版增加的留白已经考虑在内）。

最通用的 80g 纸，厚度是 0.11 毫米。在双面印刷的情况下：总字数/每页行数/每行字数/11.364 就可以算出一个大约的厚度（单位为毫米）。

§2 全网格化（进阶）

在正式出版物中，还有其他强迫症发作的进阶规则。

1. 行宽是整数个字的宽

即一排汉字能将一行恰好排满，提升每页纸上的网格感。

II 排印学知识

2. 标题所占高度是整数个行的高

使得不同页面格式中，正文都能实现跨页行对齐。

这些规则本质上都是令阅读体验更接近于排字印刷时代，但是到了现代，出版社也不是一定遵守这些规矩，姑且看看，了解一下即可。除非有很强的强迫症，不然不需要应用。

了解了以上内容后，就让我们进入正题，开始使用 $\text{\LaTeX}$ 进行排版。

III 从文本到文件

注意， $\text{\LaTeX}$ 最好使用纯 ASCII 字符作为文件名和路径名，否则编译时可能出现乱码。（仅限编译记录，不影响正常生成文档）

§1 $\text{\LaTeX}$ 正文语法入门

$\text{\LaTeX}$ 内容写作语法复杂度只在公式上，普通文本只比 `txt` 复杂一点点。如果嫌我写得太抽象了，在 *Overleaf* 上找点模板玩玩即可理解。

i 文件结构

$\text{\LaTeX}$ 文件分导言区和正文区。导言区用于编写预设，正文区用于放置文档内容。

```
1 | % 导言区
2 | % 想立刻看编译效果的话，增加下面这行
3 | % \documentclass{ctexbook}
4 | \begin{document}
5 |     % 正文区
6 | \end{document}
```

代码 3.1: 文件结构

$\text{\TeX}$ 会先按顺序加载导言区的内容，前文的定义将影响后文，后文可以覆盖前文。导言区全部加载后开始逐行制作文档，正文区的命令在 $\text{\TeX}$ 处理到所在位置时起效。

ii 正文

正文分段需要使用两个回车（即空一行），连用额外的回车视作无效。可以通过加一个空白段落的方式空一段，如下。

```
1 | 上一节最后一行
2 |
3 | % 方案一，空白符
4 | ~\
5 | % 方案二，空白盒
6 | \mbox{}
7 | % 方案三，空
8 | \null
9 |
10 | 下一节第一行
```

代码 3.2: 空白段

单个回车在排版时会视作空格。

用\\（两个反斜杠）在不换段的情况下换行。

加粗方法：`\textbf{加粗内容}`

斜体方法：`\textit{斜体内容}`

强调方法：`\emph{强调内容}`，默认使用斜体进行强调，如果强调中再强调一次，则会把二次强调的内容换回正体。

iii 特殊字符

以下特殊字符在 $\text{\LaTeX}$ 中有特殊意义，在正文中使用不能直接写：

百分号（%）， $\text{\LaTeX}$ 将其视作注释标志，在正文中使用请用`\%`。

反斜杠（\，又叫转义符），用处多种多样，在正文中使用请用`\textbackslash`。

大括号（{和}），默认按分割标记理解。在正文中使用，请用`\textbraceleft`和`\textbraceright`。

波浪号（~），默认按硬空格（硬空格 = 不可在此处换行的空格）理解。在正文中使用，请用`\textasciitilde`。

iv 标题

各级标题的名称见各级标题。

有序标题的格式是`\level{标题内容}`（level 代指实际的级别）。其中，序号标记会自动处理，不用写进正文文档区。

无序标题的格式是`\level*{标题内容}`（level 代指实际的级别）。不含序号标记，不会参与索引建立。

§2 格式转换

对于已标记好格式的文本（包括但不限于.docx，.md，.rtf，.epub），可用各种免费网页工具或 **pandoc** 转化为.tex，然后手动检查。注意，转化会完全继承原文档的格式，导言区的内容大部分是冗余的，清掉即可。

中文和外语标点存在一定的 Unicode 码重叠，自动化工具通常会将标点按外语机制转化，并用外语字体进行渲染。

其中，单引号必须手动调整。转化机制不区分单引号的正反，一律将行首的单引号自动转化为`（一个英文反引号），其他单引号转化为'（一个英文单引号），因此会出现中文引号不配对的情况。在.tex 文件中按原文换回中文单引号即可解决问题。

其他标点的外语化仅会影响外语字体，看不顺眼的话，全文替换换回来即可。

标点	中文写法	外语写法
省略号	……	<code>\ldots\ldots</code> ，其后可能有（一个空格） 或 <code>{ }</code> （一组大括号）
左引号	“	<code>` `</code> （两个反引号）
右引号	”	<code>' '</code> （两个单引号）

----- (六个减号)		
破折号	——	L ^A T _E X 中一个减号代表连字符 (hyphen)，两个代表连接号/减号 (en dash)，三个代表破折号 (em dash)

表 3.1: 标点中文化

破折号有点复杂，见后文规定标点宽度。

另外, 自动化转换工具可能会将斜体转为`\emph{斜体内容}`, `\emph` 意为强调 (emphasis), 并不完全等同于斜体，见正文。

好了，你入门了。

读到这里，你已经掌握了 L^AT_EX 的全部语法知识了，可以排版了。

IV 建立文档

§1 文档类型和主字号

文档类型通常在文件的第一行定义。

```
1 | \documentclass[10pt]{book}
```

代码 4.1: 文档类型和主字号

TeX 原生有着 **article**、**book**、**report** 三种文档类别。三种类别的主要区别在默认层深和排版方式上，虽然排版之后肯定要自己改，但为了直观，本教程推荐使用 **book** 类。

book 类默认支持三种正文字号 10,11,12pt。三种字号可读性都属不错，不建议更大或者更小。如果一定想用其他的，可以使用 **extbook** 类文档，支持 8,9,10,11,12,14,17,20pt 的正文字号。

具体选择时，可以用“磅数 $\div 2.845 =$ 毫米数”来计算文字大小。也可参考 word，word 中的五号字是 10.5 磅，小四则是 12 磅。

i 中文处理：ctex 包

TeX 中使用中文时需要的内容集成在了 **ctex** 包里，需要 XeTeX 或 LuaTeX 引擎才能编译。使用方法为

```
1 | % 方案一
2 | % 在建立文档时指定使用ctex，Z代指文档类型
3 | \documentclass{ctexZ}
4 |
5 | % 方案二
6 | % 建立文档后引入包
7 | \documentclass{Z}
8 | \usepackage{ctex}
```

代码 4.2: 使用 **ctex**

方案一相当于建立中文文档，方案二相当于在英文文档里使用中文。方案一自动将所有预设词翻译为了中文，更加便捷；方案二在细节上更加通用。例如，按方案一生成的目录中，标题内“第 x 章”“第 x 部分”等字样需要用 **ctexset** 命令来调整；而方案二可以用更加基础（i.e. 与其他包兼容性更好）的方式对这些地方进行自定义。

ctex 的文档类别除了 10,11,12pt 外，还支持 word 款的两种正文字号，配置方法如下

```
1 | % 正文五号字
2 | \documentclass[zihao=5]{ctexbook}
3 | % 正文小四号字
4 | \documentclass[zihao=-4]{ctexbook}
```

代码 4.3: **ctex** 字号

ii 字号命令使用

表 4 标准字体命令与字号的对应

字体命令	zihao = 5		zihao = -4		10pt	11pt	12pt
	字号	bp	字号	bp	pt	pt	pt
<code>\tiny</code>	七号	5.5	小六	6.5	5	6	6
<code>\scriptsize</code>	小六	6.5	六号	7.5	7	8	8
<code>\footnotesize</code>	六号	7.5	小五	9	8	9	10
<code>\small</code>	小五	9	五号	10.5	9	10	11
<code>\normalsize</code>	五号	10.5	小四	12	10	11	12
<code>\large</code>	小四	12	小三	15	12	12	14
<code>\Large</code>	小三	15	小二	18	14	14	17
<code>\LARGE</code>	小二	18	二号	22	17	17	20
<code>\huge</code>	二号	22	小一	24	20	20	25
<code>\Huge</code>	一号	26	一号	26	25	25	25

图 4.1: 字号（出自 `ctex` 文档）

在文档字号不同时，各个字号命令对应的字号如上图。字号命令一次使用对后文全部生效，局部使用时，应用 `{ }` 将对应区域括起，其中包含字号命令和改变字号的内容。

此外，可以用`\fontsize`命令手动定义文字大小，如下。

```

1 | \fontsize{字号}{所占高度}

```

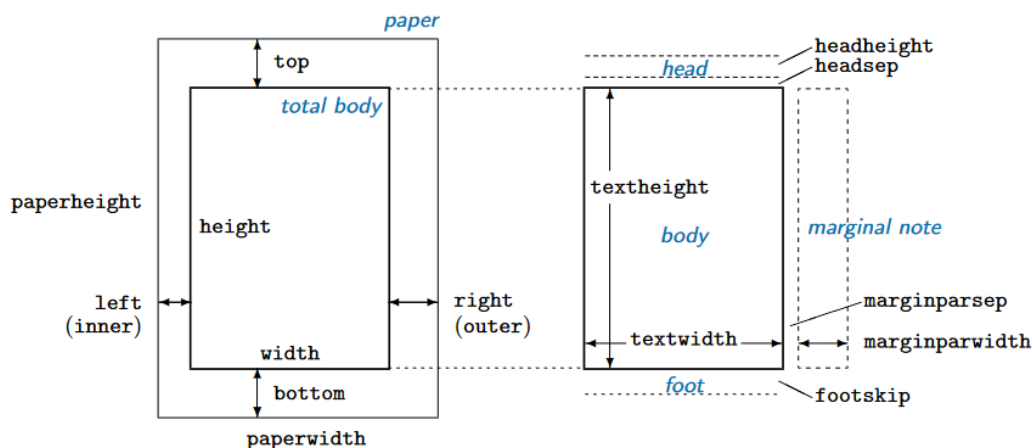
代码 4.4: 自定义字号

对于中文，可以用`\ctexset{ziju= 额外倍数}`来额外改变字距，但是不设定的话就是最好看的了。

§2 纸张布局

一般来说，同人文本的尺寸是 A5 左右。标准 A5 是 148*210mm，实际中印厂提供的尺寸通常会更窄一些。但我个人还是建议按标准尺寸来做设计，外边距适当留一些余量（5~8mm 即可）。

`geometry` 是处理布局的包，参数极多，这里只介绍少量常用内容，有其他需求建议阅读文档。

图 4.2: 纸面布局 (出自 `geometry` 文档)

版式中，水平竖直方向各有四个参数，`paperheight`，`paperwidth` 这组对应纸张大小，`textheight`，`textwidth` 这组对应版心（纸上印内容的区域）大小。竖直向的 `top`，`bottom` 和水平的 `left`，`right` 或 `inner`，`outer` 则用于在纸张上确认版心位置。

规定版式时，水平竖直方向各明确三个参数即可，最后一个会自动计算。注意，水平方向 `left`，`right` 和 `inner`，`outer` 这两组是二选一的关系。双页印刷应当使用 `inner`，`outer` 这组参数，能根据页面单双自动调整文字区域位置。

以下是一个布局实例。

```

1 % 引入包
2 \usepackage{geometry}
3 \geometry{
4     % A5纸张标准宽高
5     paperwidth=148mm,
6     paperheight=210mm,
7     % 天头，文字区到纸张上沿的距离
8     top=17mm,
9     % 地脚，文字区到纸张下沿的距离
10    bottom=23mm,
11    % 切口，文字区到纸张被翻开一侧的距离
12    outer=13mm,
13    % 订口，文字区到纸张被装订一侧的距离
14    inner=21mm,
15    % 文字区和页眉下沿之间的空白
16    headsep=7mm,
17    % 页眉的高度，一般不需要这行
18    % headheight=6mm,
19    % 文字区和页脚下沿之间的空白

```

```

20 |     footskip=12mm,
21 | }

```

代码 4.5: 布局

`\geometry` 命令会决定纸张的大小，后续不可修改，尝试修改也会被无视。边距等版式细节数值则可以在文档内部用`\newgeometry` 命令进行修改，如

```

1 | \begin{document}
2 |     % ...
3 |     % 新的边距会从当前页起效
4 |     \newgeometry{left=30mm,right=30mm,top=20mm,bottom=20mm}
5 |     % ...
6 |     % 将边距重置为\geometry命令中设置的
7 |     \restoregeometry
8 |     % ...
9 | \end{document}

```

代码 4.6: 改变布局

i 出血

上述讲解的 A5 纸张所得到的是效果图，印刷时需要为印厂提供有 3mm 出血的版本。增加出血很简单，只需要将 `paperwidth`, `paperheight` 各增加 6mm，每一处设置的 `top`, `bottom`, `inner`, `outer`, `left`, `right` 各增加 3mm 即可。`textwidth` 和 `textheight` 两个参数不用改变。

在正文排版没有 bug 的情况下，这样的修改可以在效果图四周都增加 3mm 的白边，同缩放比例，与效果图中心对齐检阅时，文字位置完全不变。如果修改后有页面发生了移动，那么请在该页的写法上寻找问题。

§3 各级标题

`book` 类可以使用所有种类的标题，直观地说，可以理解为一本大部头学术书籍能涵盖的级别。对于小说，建议只选用 `part+chapter` 或 `chapter+section`，甚至只用一级标题，不建议使用超过两级的标题。

具体标题类别如下（层深的概念在目录中会用到）：

层深	标题	含义	效果	用处
-1	<code>\part</code>	部	标题新起一张纸并独占它	50+ 页分割
0	<code>\chapter</code>	章	标题新起一张纸 标题下可以有内容	15+ 页分割
1	<code>\section</code>	节	标题与上文同页	3~5+ 页分割
2	<code>\subsection</code>	-	标题与上文同页	看着用
3	<code>\subsubsection</code>	-	标题与上文同页	看着用

表 4.1: 各级标题

新 `part` 会重置一部分前文的格式定义，如行高，可以通过把命令写进 `part` 宏的方式提升复用性。不需要未雨绸缪，发现有命令失效后再添加即可。

```

1 % 重载part宏
2 \let\origpart\part
3 \renewcommand*{\part}[2][{}{
4     \ifx\#1\
5     \origpart{#2}
6     \else
7     \origpart[#1]{#2}
8     \fi
9     % 写这里就行
10 }
```

代码 4.7: 重载 `part` 宏

§4 标题格式

对于文内标题，可以使用 `titlesec` 包内的命令来配置格式。

这个包里，标题格式用 `\titleformat` 命令来配置。

```

1 % 参数
2 \titleformat{命令}[形式]{格式}{标题序号}{间距}{前命令}[后命令]
3 % 示例
4 \titleformat{\chapter}{\huge\chapterfont}{\thechapter}{1em}{\
  thispagestyle{empty}}
```

代码 4.8: 配置标题格式

而标题与上下文的间距，使用 `\titlespacing` 命令配置。

```

1 % 参数
2 % 注：左间距的相对点是这一行文字的起始位置，即段首缩进后的
3 \titlespacing{命令}{左间距}{上间距}{下间距}[右间距]
4 % 示例
5 \titlespacing{\chapter}{0em}{.5\baselineskip}{2\baselineskip}
```

代码 4.9: 配置标题间距

标题上的编号默认为 *1.1.1* 类，即子级编号会带上上级，用以下命令来去除子级的上级编号。

```
1 \counterwithout{子级别}{亲级别}
```

代码 4.10: 编号独立

V 字体设定

字体最好在排细节之前确定。不同字体文件的细节不同，换字体有可能导致文档的换行和换页发生改变，哪怕纯中文文档也是如此。

§1 字体入门

i 字体类别

字体可以简单分为有衬线（Serif）和无衬线（Sans Serif，或简称为 Sans）两类。衬线指笔画边角处的装饰，例如宋体是典型的衬线字体，而黑体是典型的无衬线字体。

纸张上，有衬线的字体易读性更佳，一般用于正文中；无衬线字体更加醒目，可以用于封面或勒口（封皮往内折的部分）。

等宽字体（mono）指所有字母和符号都占据同样宽度的字体。通常说的“方块字文字”是等宽的，但其字体标点有可能不等宽。对字母文字而言，等宽字体的易读性并不太好，应用场景很有限。除了有意模仿打字机的文字质感外，只有编程为了竖对齐使用等宽字体。

ii 字体风格

一个字体名是一个系列，其中往往有多个风格，最重要的是各种字重和斜体。

字重：描述一个笔画有多粗。

从轻到重（从细到粗）分别有：thin, extra light, light, regular (normal), medium, semi bold, bold, heavy (black)。一个字体必然有字重关键词，留空时会自动使用 regular。原则上的加粗行为就是字重从 regular 变成 bold。如果一个字体没有 bold 字重，部分软件在加粗时会自动用算法生成一种伪粗体，典型如 word。但 $\text{\LaTeX}$ 没有这个功能，它会给你报一个字体缺失警告，文件中用原字重字体暂行填补。

斜体：Italic，严格上来说应该叫意大利体。字母文字排版时产生的风格化方案，中文其实没有斜体，承担类似功能的是楷体或仿宋。

不是所有字体都有意大利体，很多时候我们看到的也是算法计算的伪意大利体。方案是简单将字符拉斜一些，不涉及意大利体中常见的字形变化。这种伪意大利体其实才该叫斜体（Slanted）。但由于 Italic 风格本身的舶来性，加之 Slanted 几乎没什么专门的用处，大家已经习惯将 Italic 称作斜体，因此我后文也沿用这种通俗的称呼。

iii 字符集和编码表

字符集指字符本身；编码表指字符字形和字符代码的一一映射表。通常来说，一套字标准是包含字符集和编码表的。本章中仅介绍字体，因此只讨论字体对字符集的覆盖率，而不讨论编码（计算机可以在兼容的编码表中做转换，在现代排版中基本不构成问题）。

以下是一些常用的字符集：

字符集名称	内容
ASCII	基础字符集，可见部分为美式键盘上能直接输入的字符 此处的美式是与英式、法式、俄式等区分，中国能买到的键盘一般就是美式的
GB2312	中文基础字符集，含 6763 个汉字字符。兼容 ASCII
GBK	汉字扩展集，含 21886 个汉字字符。兼容 GB2312
GB18030	中文编码字符集，含 87887 个汉字字符，兼容 GBK
Unicode	国际字符表，基本含一切能用的字符和 emoji，常用标准 utf8 兼容 ASCII

简单来说，GB 和 Unicode 是两套不同的编码，同一个文字在两个标准里会对应不同的编码，不过 $\text{\LaTeX}$ 做好了兼容，因此无论字体文件使用何种编码集，都可以正常渲染出结果。

§2 ctex 中的字体

ctex 包默认根据当前操作系统选择字体配置，策略如下

表 3 $\text{CT}_{\text{E}}\text{X}$ 宏集自动配置字体策略

	macOS Old ¹	macOS New ²	Windows ³	其他
$\text{Xe}_{\text{L}}\text{\LaTeX}$	xeCJK 华文字库	xeCJK 华文字库 + 苹方	xeCJK 中易字库 + 微软雅黑	xeCJK Fandol 字库 ⁴
$\text{Lua}_{\text{L}}\text{\LaTeX}$ ⁵	LuaTeX-ja 华文字库	LuaTeX-ja 华文字库 + 苹方	LuaTeX-ja 中易字库 + 微软雅黑	LuaTeX-ja Fandol 字库
$\text{pdf}_{\text{L}}\text{\LaTeX}$	不可用	不可用	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	不可用
$\text{\LaTeX}$ + DVIPDFMx	不可用	CJK + zhmetrics 华文字库 + 苹方	CJK + zhmetrics 中易字库 + 微软雅黑 ⁶	CJK + zhmetrics Fandol 字库
$\text{up}_{\text{L}}\text{\LaTeX}$ + DVIPDFMx	不可用	zhmetrics-uptex 华文字库 + 苹方	zhmetrics-uptex 中易字库 + 微软雅黑	zhmetrics-uptex Fandol 字库

1 Yosemite (10.10) 及以前的 macOS 系统。
2 El Capitan (10.11) 及以后的 macOS 系统。
3 仅支持 Windows Vista 及以后的 Windows 操作系统。
4 由马起园、苏杰、黄晨成等人开发的开源中文字体，参见：<https://www.ctan.org/pkg/fandol>。
5 $\text{Lua}_{\text{L}}\text{\LaTeX}$ 编译时使用 LuaTeX-ja 宏包。对此，第 9 节有特别说明。
6 微软雅黑字体并不总是有效，这和选项 zhmap 的取值有关。

图 5.1: ctex 预设包，出自其文档

ctex 包内有若干套预设好的字体配置，可在导入时使用 `[fontset=X]` 选择，X 为包名，详见下表。

预设名	使用字体	版权
adobe	Adobe 公司的四款中文字体	付费商用
Fandol	Fandol 中文字体	GPL+FE 协议 ¹ 开源： 放入文档中可以随意使用 
founder	方正公司的中文字体	书宋/黑体/楷体/仿宋 四种为免费商用  原则上需要申请一份 书面授权书
mac	macOS 系统下的字体	印刷时存在侵权问题
macnew	ElCapitan 或之后的多字重 华文字体和苹方字体	同上
macold	Yosemite 或之前的华文字体	同上
ubuntu	思源宋体、思源黑体和 TeX 发行版自带的文鼎楷体	免费商用 
windows	中易字体和微软雅黑字体	付费商用

表 5.2: ctex 字体预设

TeXLive 自带字库中包含 Fandol 系列,各平台均可用 `[fontset=fandol]` 加载全套 Fandol 字体配置,简单达到跨平台统一。但是它只含 GB2312 内的字符,可能会出现缺字。如果出现这种问题的话,还是只能自行配置字体。

§3 常用字体获取

外语字体可在1001 Fonts²上挑选。我推荐EB Garamond 12³,该字体实现了拉丁、希腊、西里尔字母的全覆盖,用起来很省心。

中文字体:

思源系列开源且免费商用,给出官方地址:思源宋体⁴,思源黑体⁵。

方正系列官网需要登录等操作,这里给出第三方下载链接:方正书宋⁶、方正楷体⁷、方正仿宋⁸、方正黑体⁹。不确定网站可靠性,链接或许会失效。请注意,在找方正字体文件时请一定找 GBK 后缀(书宋字体文件 8.87M),而非 GB2312 或简体后缀(书宋字体文件 2.63M),预防字体缺字。

¹基于此字体改编、修改等所有再创作的字体产品,均必须同样继承 GPL+FE 协议开源
²1001 Fonts: <https://www.1001fonts.com/>
³EB Garamond 12: <https://www.1001fonts.com/eb-garamond-font.html>
⁴思源宋体: <https://github.com/adobe-fonts/source-han-serif/releases/latest>
⁵思源黑体: <https://github.com/adobe-fonts/source-han-sans/releases/latest>
⁶方正书宋: <https://www.fonts.net.cn/font-31610316242.html>
⁷方正楷体: <https://www.fonts.net.cn/font-31607222283.html>
⁸方正仿宋: <https://www.fonts.net.cn/font-31602268591.html>
⁹方正黑体: <https://www.fonts.net.cn/font-31609167689.html>

§4 字体配置

这里介绍两种字体配置方法。路径法适用于在多个工作环境中使用同一项目，安装法适用于在单一工作环境下维护多个项目。*Overleaf* 仅支持路径法。

i 路径法

路径法增加一个 `[Path= 路径]` 的可选参数，人为限制 $\text{\LaTeX}$ 的搜索范围，设定主字体的方法如下。其中，路径使用绝对路径和（以编译文件的位置为起点的）相对路径均可，需要用代表文件夹的 `/` 或 `\` 结尾。

`fontspec` 包命令支持使用专属名和文件名，`ctex` 包命令似乎是不支持专属名的。

```

1 | 主目录
2 | |   main.tex
3 | |   └── fonts
4 | |       myfont.otf
5 | └── ...

```

文件结构示例

在这样的文件结构下，使用以下写法设定字体。

```
1 | \setmainfamily[Path=fonts/]{myfont}
```

代码 5.1: 路径法配置主字体

文件名或路径仅使用 ASCII 编码覆盖的字符为宜，不要使用中文文件名或路径。

ii 安装法

安装法即为将字体安装进系统中，通过字体名称调用。 $\text{\LaTeX}$ 可以自动检索系统路径 `C:\Windows\Fonts\` 下的字体和 *TeXLive* 安装时带的字体。如需自定义字体，安装时需要选择为所有用户安装。

由于安装法的字体名检查起来较为不易，推荐使用专属名。（用 *Everything* 的话可以无视这句）

以下为设定主字体的方法。

```
1 | \setmainfamily{字体名称}
```

代码 5.2: 安装法配置主字体

§5 字体名称

i 字体名称查询

配置字体时，名称可以使用字体专属名和文件名（含不含后缀均可）。

字体专属名是嵌入字体文件内的，也是操作系统识别字体时所用的标识，以此配置较为稳定，可读性也较高。

字体文件名是可以随意修改的，只建议在使用专属名出错或文件名容易检查时使用。

在 Windows 中,简单的查找字体专属名称方法是运行 `fc-list` 命令,其覆盖范围与 TEX 能检索的范围一致。

可以用系统命令 `fc-list >> fonts.txt` 生成一个字体表文件,包括系统上的可用字体。增加 `:lang-zh` 参数可以指定过滤筛选中文字体。注意一些字体虽然使用时是中文,但其字体文件会被识别为日文或韩文,不会出现在结果中。确定英文系列名时,可以用 `fc-list | Select-String "系列名"` 来筛选字体列表。中文名可能会是乱码,建议只用英文名进行此项操作。

得出的结果中包含字体名,这里使用开源字体 *Vollkorn* 系列举例, `fc-list | Select-String "Vollkorn"` 得到的结果如下

```
1 C:/Windows/fonts/Vollkorn-Medium.otf: Vollkorn,Vollkorn Medium:style=
   Medium,Regular
2 C:/Windows/fonts/Vollkorn-Italic.otf: Vollkorn:style=Italic
3 C:/Windows/fonts/Vollkorn-Bold.otf: Vollkorn:style=Bold
4 C:/Windows/fonts/Vollkorn-MediumItalic.otf: Vollkorn,Vollkorn Medium:
   style=Medium Italic,Italic
5 C:/Windows/fonts/Vollkorn-BoldItalic.otf: Vollkorn:style=Bold Italic
6 C:/Windows/fonts/Vollkorn-Regular.otf: Vollkorn:style=Regular
7 C:/Windows/fonts/Vollkorn-SemiboldItalic.otf: Vollkorn,Vollkorn
   Semibold:style=Semibold Italic,Italic
8 C:/Windows/fonts/Vollkorn-Semibold.otf: Vollkorn,Vollkorn Semibold:
   style=Semibold,Regular
```

Vollkorn 查询结果

其中 `*.otf:` 和 `:style` 之间的 *Vollkorn* 即为字体在系统里的名称。对于字体的特殊风格,可以直接以字体名 风格作为字体名加载,如 *Vollkorn Semibold Italic*。切记区分大小写,有的字体里会是 **SemiBold**,有的是 **Semibold**。

注:一些字体名中含有`-`,在打印时会增加转义符显示为`\-`,使用这些字体时输入`-`即可。

另外,也可以使用FontDrop!¹网站解析单个字体文件,获得字体名。解析样例字体 *EBGaramondSC12-Regular.otf* 时结果如下。

```
1 You see EB Garamond SC
2 Name: EB Garamond SmallCaps 12 Regular. Style name: 12 Regular.
   Version 0.016 © Created by Georg Duffner with FontForge 2.0 (http
   ://fontforge.sf.net)
3 License: Copyright 2010-2013, Georg A. Duffner (<http://www.
   georgduffner.at/ebgaramond|g.duffner@gmail.com>), 2013 Siva Kalyan
   This Font Software
```

FontDrop! 解析结果

其中 `You see` 后的是字体系列名,而 `Name` 与 `. Style name` 之间的 *EB Garamond SmallCaps 12 Regular* 即为字体本身在系统里的名称(对于风格字体,即为指定时使用的名称)。

¹FontDrop!: <https://fontdrop.info>

ii 配置的字体找不到?

第一步, 确定字体确实安装在了 `C:\Windows\Fonts` 文件夹里。且字体文件名 (`.otf` 前的部分) 是纯 ASCII 字符 (有一段时间, 中文命名的字体文件可以被正常调用, 但 2026.5.12 发现不支持了, 所以还是建议修改为纯 ASCII 字符)。

第二步, 检查代码中的拼写和 `fc-list` 命令获得的一样, 特别是大小写。

第三步, 用 [FontDrop!](https://fontdrop.info)¹ 网站解析单个字体文件, 获得字体名, 和 `fc-list` 里获得的可能不一样。

第四步, 在字体专属名外面加个中括号。别问为什么, 我也不懂, 总而言之亲测有时候是有效的。

```
1 | % 修改前
2 | \newCJKfontfamily\xbsong{Source Han Serif SemiBold}
3 | % 修改后
4 | \newCJKfontfamily\xbsong{[Source Han Serif SemiBold]}
```

代码 5.3: 找不到字体的处理方案

第五步, 改用文件名配置。

第六步, 天涯何处无芳草, 字体到处都是, 换一个吧。

iii fontspec 包字体配置

TeX 自带的字体配置方案, 默认只支持外语字符生效 (不止为 ASCII、各种欧洲语言, 还包括章节符 § 和摄氏度 °C 等国际通用符号)。这些字符的渲染只有 `fontspec` 包配置的字体才能起效。

中日韩三语的文字字体, 需要使用 `ctex` 包中的命令另行配置。

我们先介绍基础字体配置方法:

```
1 | % 引入包
2 | \usepackage{fontspec}
3 | % 设置主字体
4 | \setmainfont{Vollkorn}
```

代码 5.4: 配置基础字体

基础字体可以设置三类别:

- 主字体 `\setmainfont` 用 `\rmfamily` 调用
- 无衬线字体 `\setsansfont` 用 `\sffamily` 调用
- 等宽字体 `\setmonofont` 用 `\ttfamily` 调用

一般设个主字体就够了, 其他两种字体是默认格式中使用的, 自定义格式时反正也要覆盖掉。

用 `\newfontfamily` 配置新的字体。

```
1 | % 设置字体, 并自定义字体名\medfont
2 | \newfontfamily\medfont{Vollkorn Medium}
```

代码 5.5: 自定义字体

¹FontDrop!: <https://fontdrop.info>

TEX 默认寻找同系列的字体作为其加粗和斜体，但可以自行进行指定。

```
1 % 设置字体，并自行配置加粗和斜体
2 \newfontfamily\medfont{Vollkorn Medium}[BoldFont=Vollkorn Semibold,
   ItalicFont=Vollkorn Semibold Italic]
```

代码 5.6: 指定加粗和斜体

使用非主字体时，只需要输入配置时设定的字体名即可。与字号类似，字体命令一次使用也对后文全部生效，局部使用时，应用 `{}` 将对应区域括起，其中包含字体命令和改变字体的内容。

```
1 \begin{document}
2   % 一次性使用
3   \fontspec{Vollkorn medium}[...]
4   % 预先设定好
5   \medfont 对应文本
6 \end{document}
```

代码 5.7: 使用自定义字体

iv ctex 包字体配置

中文字符，如你，我，；（中文分号），《（中文书名号），均需要使用 `ctex` 包内的命令进行字体配置才能正常显示。

如果使用 `ctex` 包字库的话，字体命令已经预定义好了。主字体默认为宋体，加粗为黑体，斜体为楷体，等宽（`\ttfamily`）为仿宋。

```
1 \songti 宋体
2 \heiti 黑体
3 \fangsong 仿宋
4 \kaishu 楷书
5 % 只对 windows、founder 和 macnew 字库生效
6 \lishu 隶书
7 \youyuan 幼圆
```

代码 5.8: ctex 默认字体使用方式

如果选择自定义，首先要引入包时用 `[fontset=none]` 避免加载默认字体集，防止产生冲突。

```
1 \documentclass[fontset=none]{ctexZ}
2 % 或
3 \usepackage[fontset=none]{ctex}
```

代码 5.9: 禁用默认中文字体配置

字体配置的命令和 `fontspec` 包的差不多，加个 `CJK` 即可配置中文、日文、或韩文的字体，配置出的字体只会对这三种语言内的字符生效。


```

1 % 主字体
2 \setCJKmainfont{Source Han Serif}[BoldFont=Source Han Serif SemiBold,
   ItalicFont=FZKai-Z03]
3
4 % 以下是六种常用的GBK免费商用中文字体名
5
6 % 思源系列字体有Source Han和Noto两个名字，是一样的
7 % 不同国家和地区使用的同一汉字具有同一Unicode编码，但字形不同，用后缀
   区分
8 % 语言包后缀：J（日语），K（韩语），SC（简体中文），HK（繁体中文-香
   港），TW（繁体中文-台湾）
9 % 国家/地区后缀：JP（日本），KR（韩国），CN（中国），HK（香港地区），
   TW（台湾地区）
10 % CN通常足够使用，SC较CN覆盖范围要更广一些
11 \newCJKfontfamily\songti{Source Han Serif CN} % 思源宋体 CN
12 \newCJKfontfamily\sheiti{Source Han Sans SC} % 思源黑体 简中
13
14 % 方正系列，GBK字符集
15 % 有S（简体）T（繁体）两种后缀的字体，对应的是GB2312字符集，不建议用
16 \newCJKfontfamily\shusong{FZShuSong-Z01}% 方正书宋
17 \newCJKfontfamily\fangsong{FZFangSong-Z02} % 方正仿宋
18 \newCJKfontfamily\heiti{FZHei-B01} % 方正黑体
19 \newCJKfontfamily\kaiti{FZKai-Z03} % 方正楷体

```

代码 5.10: 配置中文字体

其中，方正系列四个字体都只有一个字重，而思源两个系列都具有 extralight, light, regular, medium, semibold, bold, heavy 七个字重，可以进行指定。两类字体中都不含斜体。

v 外语连字（进阶）

外语衬线字体中存在连字（ligature），通过设计优化某些字符串的显示方式来提升易读性。最经典的三个案例如下图（字体为 *Vollkorn*）。

f i l f i f l f f

图 5.2: 常见连字

LaTeX 默认载入一部分通用连字 (OTF 中 tag 为 liga)，能提升外文文本的易读性。而一些字体对连字的设计比较充分，还使用了其他的连字 tag。通常类别有：clig 指上下文连字 (contextual)，能使手写体中产生连笔；dlig 指自由连字 (discretionary)，会使文字更花俏；hlig 指历史连字 (historical) 能令文字看起来较为复古；rlig 标记必需连字 (Required)，可以实现“将英语字母拼合得到其他字母”，例如 ae 拼合为 æ。

本教程的正文字体启用了历史连字，因此 &，&t 上面有一笔勾连；启用了自由连字，因此 Th 上端出现了融合。

Table 11: Options for the OpenType font feature `Ligatures`.

Feature	Option	Tag
<code>Ligatures</code>	= Required	<code>rlig</code> †
	Common	<code>liga</code> †
	Contextual	<code>clig</code> †
	Rare/Discretionary	<code>dlig</code> †
	Historic	<code>hlig</code> †
	TeX	<code>tlig</code> †
	ResetAll	

† These feature options can be disabled with `..Off` variants, and reset to default state (neither explicitly on nor off) with `..Reset`.

图 5.3: 连字配置 (出自 `fontspec` 包文档)

配置时可以添加一个 `[Ligatures=XXX]` 参数, 启用一些其他 tag 里的连字, 见上图。其中 Common 是默认打开的。

可以用FontDrop!¹这个网站来查看字体内连字。有些字体虽然设计了连字, 但并没有正确标注, 导致这些连字虽然存在但却无法使用。面对这种情况, 可以用FontForge²软件进行手动标注³ (相当繁琐, 还是建议找一个做得好的字体文件)。

§6 高级配置

本节介绍的高级配置最好是在阅览的过程中调整。

i pdf 浏览器

想看字体效果, 先要有一个好使的 pdf 浏览器。

打印的分辨率是 300dpi, 而屏幕的分辨率鲜少到达这个级别, 因此屏幕上预览的 pdf, 精度比起打印机通常是压缩了的。由于渲染机制原因, 不同 pdf 浏览器通常有着不同的压缩效果, 建议找一个使用方便, 且页面模式下与放大到 300dpi 下效果最相似的。个人经验中, Windows 端的 Acrobat DC Pro 在英文花体字上发生了笔划显著过细的现象, 而 SumatraPDF 和 Edge, Firefox, Chrome 均未出现此问题。

一些浏览器 (如 SumatraPDF、Acrobat DC Pro) 支持自定义屏幕分辨率, 合理配置后以在屏幕上显示与实物等大的内容。屏幕分辨率根据显示器尺寸 (即对角线长)、长宽比例、勾股定理、屏幕像素数即可计算, 单位 dpi 为 dots per inch, 即每英寸像素数。

SumatraPDF 的配置方式为目录-设置-高级选项, 在配置文件中把 `CustomScreenDPI = 0` 改为 `CustomScreenDPI = 屏幕分辨率`。Acrobat DC Pro 的配置方式为 `Ctrl+K` 打开首选项, 在自定义分辨率中输入屏幕分辨率/系统缩放比例 (操作系统中设置的显示缩放比例) 的值。

¹FontDrop!: <https://fontdrop.info>
²FontForge: <https://fontforge.org/en-US/>
³手动标注: <https://fontforge.org/docs/tutorial/editexample4.html>

ii 中文

中文的可选字体其实不少（[开源字体可以见此名单](#)¹）。从各方面收集到的资料来看，方正系的字体适用范围最广、知名度最高，设计应该是最佳的。

实际使用时，推荐将方正书宋（书宋即完全针对书籍印刷设计的宋体）作为主字体，指定方正楷体或仿宋作为斜体，以便于在正文中标注少量字符（如引用），这样易读性较佳。

不过，方正书宋只有一个字重免费，固然技术上可以将较粗的思源宋体或方正黑体作为粗体使用，但思源系列与方正书宋字形相似又不同，夹杂使用令人不适；黑体也不适用于书写正文。如果对加粗有明确的需求，请挑一款多字重的字体（如思源宋体）作为主字体。不过，由于字形差异足够大，依然可以搭配方正字体作为斜体。

装饰性文字（脚注、页眉页脚等）建议使用方正仿宋或楷体，但注意与正文的斜体字体有所区分。

各级标题字体建议用不同字重的思源宋体来显示。

内页不推荐使用黑体。

可以用`\ctexset{ziju= 额外倍数}`来额外改变字距，但是不建议改，属于费力不讨好。

iii 外语

外语的免费字体很多，必然支持 ASCII 字符，绝大部分都支持拉丁字母及变形，一部分支持希腊和西里尔字母，选择字体时，先统计文本所用符号，再确认字体支持即可。

有一些进阶选项，恰当使用可以提升混排效果。这里介绍几个比较实用的花招，更详细的推荐阅读 `fontspec` 包文档。

以下样例中，中文使用思源宋体 Regular；英文字体均是从网站[1001 Fonts](#)²下载的免费商用字体，全部使用 Regular 字重的默认配置。

思源宋体

Hal 低低地笑了：“我希望你把蝙蝠车留给 Tim，然后回家。……
他说愿意替你 7 个小时的班，让你休息一下。”

Neuton

Hal 低低地笑了：“我希望你把蝙蝠车留给 Tim，然后回家。……
他说愿意替你 7 个小时的班，让你休息一下。”

EB Garamond 12 Regular

Hal 低低地笑了：“我希望你把蝙蝠车留给 Tim，然后回家。……
他说愿意替你 7 个小时的班，让你休息一下。”

Spectral

Hal 低低地笑了：“我希望你把蝙蝠车留给 Tim，然后回家。……
他说愿意替你 7 个小时的班，让你休息一下。”

图 5.4: 混排样例

¹开源字体可以见此名单：<https://drxie.github.io/OSFCC/>

²1001 Fonts：<https://www.1001fonts.com>

Neuton 的字母偏小。配置时可用 `[Scale= 比例]` 参数放大。

EB Garamond 的字母与中文相当契合，但数字采用了与英文相同的三格设计（俗称 Old-Style），在数字与汉字混排时看起来会很诡异。可使用 `[Numbers=Lining]` 切换到等高数字。注意此方案起效建立在字体中含有等高数字的前提下，不是所有字体都支持两套数字。

Spectral 的字距比起中文明显偏高。可使用 `[LetterSpace= 改变值（如-3）]` 来降低字母间距。另，如果想要调整空格宽度，使用 `[WordSpace= 空格宽度]` 参数即可。

思源系列是全语言支持，但是需缩小一点并降低字母间距，才适用于正文混排。默认比例适合于形如“Chapter 1 节俭”这样多语言分隔的格式，或极少量穿插其他语言的大段文本。调整后即可达到这样的混排效果。

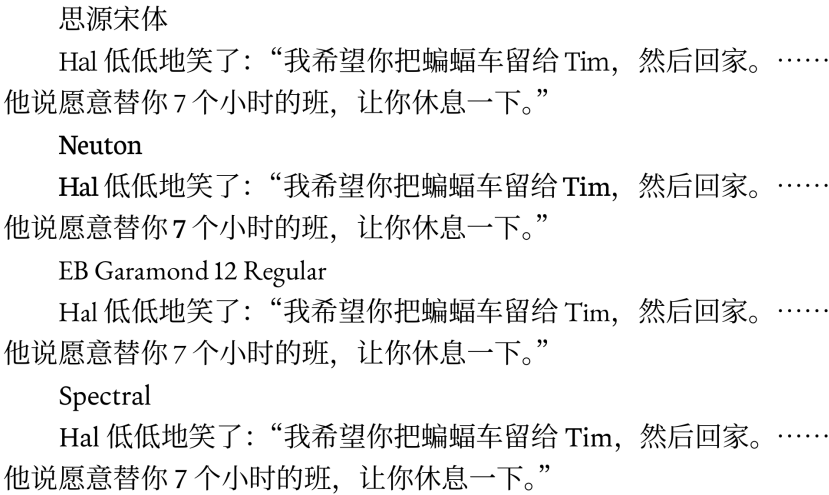


图 5.5: 混排样例（调整后）

方正系列虽然覆盖了外语的字符，但并没有做对应的设计（很多被认为是抄的），基础参数也很不合理。四种免费字体内除 ASCII 外的全部字符，均与中文字符等宽，不适合用于显示外语。因此极不建议将其作为 `fontspec` 包配置的字体。例如，这是思源宋体中的 `café`，这是方正书宋中的 `caf é`，这是 *EB Garamond 12* 中的 `café`。

VI 页面排版

§1 行高和段距

首先来明确几个词的含义：

行高：上一行文字底边到这一行文字底边的距离。

行距：即行间距，上一行文字底边到这一行文字顶边的距离。

段距：换段时，在行距基础上额外增加的距离。

i 行高

推荐使用`\setspaces`包来调整行高。

```
1 % 定义好的命令
2 \singlespacing % 单倍行高
3 \onehalfspacing % 1.5倍行高
4 \doublespacing % 双倍行高
5
6 % 自定义行高
7 \setstretch{行高倍数} % 默认行高等同于\setstretch{1.3}
```

代码 6.1: 调整行高

其中，定义好的三个命令以文字高度（即字号）为基础值，即，单倍行高会使得行高等同于文字高度。

但自定义行高的机制则不然。`\setstretch`的基础倍数为字号的120%。对于10pt的文档字号，`\setstretch{1}`会使得行高为12pt，`\setstretch{1.5}`会使得行高为18pt。

这些方式都会使`\baselineskip`（记录行高的变量）自动跟着变化，可以以此为参数设置其他场间的留白。

ii 段距

改变`\parskip`即可，默认为0pt。段距应为行高的整数倍，不然特别的丑。

```
1 | \setlength{\parskip}{段距}
```

代码 6.2: 调整段距

使用时，最好不要只给一个数，容易导致出现大量 `Overfull/Underfull \vbox` 错误，解决方法有两种。

1. 设置段距时，给出弹性空间。不要让最短段距低于0，会使得段间空白小于行间。
2. 使用命令允许每页下缘不齐。不推荐，在页面布局做不到行对齐时，该命令会使得整体看起来像狗啃的一样。

```
1 | % 设置弹性空间
2 | \setlength{\parskip}{段距 plus 最多加大值 minus 最多减少值}
3 | % 下缘不对齐
4 | \raggedbottom
```

代码 6.3: 弹性间距与不对齐

§2 缩进

TEX 默认首段不缩进，用 `indentfirst` 包来开放首段的缩进。

```
1 | \usepackage{indentfirst}
```

代码 6.4: 首段缩进

段首缩进的长度也可以调整，但不是很需要。中文文档用默认的 2 字宽是最合适的。

```
1 | \setlength{\parindent}{长度}
```

代码 6.5: 修改缩进

§3 换页

`\clearpage` 和 `\newpage` 都可以起到新开一页的效果，粗糙地使用时不会有什么区别。

精确地说，`\clearpage` 命令可以使后续内容从新的一页上开始，`\newpage` 则可以新起一栏（单栏模式下即为新开一页）。有浮动要素（如图片、表格）时，二者的处理机制也不同，如需要这部分功能请自行深入研究。

§4 新章节前的空白页

`\chapter` 及以上级别的标题会自动新起一张纸。`book` 类文档中默认为双页模式，只有右页（奇数页）为新纸，左页（偶数页）不能承担新纸的功能。所以当上文结束在右页（奇数页）时，下文出现在右页（奇数页），中间会产生一页空白页。不想要的话，可以在定义 `documentclass` 的时候增加一个参数 `openany`，允许在左页（偶数页）开启新内容。

```
1 | \documentclass[...]{...}
```

代码 6.6: 左页开始

相对的，如果在使用了 `openany` 参数后，又希望规定某些内容在右页（奇数页）上开始，则可以在前面使用 `\cleardoublepage` 命令。该命令会自动插入空白页，使后续内容从下一个右页（奇数页）铺开。

若希望后文在左页（偶数页）上开始，可以使用以下写法：

```
1 | % 新起一页
2 | \clearpage
3 | % 如果页码为奇数
```

```

4 | \ifodd\thepage
5 |     % 插入一个空白页，空白页实现见后文
6 |     \blankpage
7 | \fi

```

代码 6.7: 自动空白页

§5 手动空白页

一页强制插入的白纸包含若干要素，建议打包成一个命令，方便使用。

```

1 | \newcommand{\blankpage}{
2 |     % 新起一页
3 |     \newpage
4 |     % 页面要有内容，不然会被省掉，空box，~\，\null均可
5 |     \makebox{}
6 |     % 由于页面内有内容，需要手动去除页眉页脚
7 |     \thispagestyle{empty}
8 |     % 下一页，也可用\vfill或\clearpage实现
9 |     \newpage
10 | }

```

代码 6.8: 手动插入空白页

§6 页码更改

在文档中，可以用`\setcounter`来修改页码，后续页的页码和目录中的页码都会跟着改变。在目录后使用该命令，可以很简便地确定正文首页为第1页。

```
1 | \setcounter{page}{本页的新页码}
```

代码 6.9: 修改页码

如果某处存在若干页使用其他方式排版，希望直接插入pdf，要将这部分页码空置预留出来，可以使用`\addcounter`增加页码。

```
1 | \addcounter{page}{增加量}
```

代码 6.10: 跳过一些页码

§7 孤字和孤行（进阶）

中文排版有一条原则叫“孤字不成行，孤行不成页”。

“孤字不成行”指，一行不能只有一个字（不含标点）；“孤行不成页”指一页不能只有一行。很遗憾，这个问题是非常痛苦的“在不同丑里找平衡”大挑战，最省心美观的解法就是放弃。但如果你就不死心，我也可以给你提供一些方案。

（下文中“优雅”的标准是“一，保持内容和格式的分隔”，“二，使得不同段落、不同页面的观感尽可能一致”）

i 手动

不优雅，但好使。

孤字：把每段最后两个字（以及标点）用`\makebox{}`括起来。

孤行：引入包 `needspace`，在适当的地方加一行`\needspace{高度}`，该代码能使此后一定高度的内容留在同一页上。

评价：后患无穷，只能用于“最终 final 付梓再也不改了再动我是 XX”版本里。更关键的是，我只是说它能解决孤字孤行，没说它会好看。

ii 自动

非常优雅，但很难搞。

调整单行字数和单页行数，在让排版风格完全没有瑕疵的前提下，通过调参规避孤字和孤行。

评价：可选的参数其实就那些，好消息是可以穷举，坏消息未必举得出来。

iii 半自动

还算优雅，但不咋好使，与手动或自动搭配或可减轻一些工作量。

孤字：`ctex` 包中对于一行内能填充的空白是可规定的，默认为 `1\ccwd`，把这个数改小一些能让标点更加紧凑，部分文本将让一行纳更多的字符，或可避免孤字的出现。支持规定为数（默认单位为`\ccwd`）或长度。

```
1 | \ctexset{linestretch=伸展量}
```

代码 6.11: 行伸展量

`ctex` 包中集成了 `CJKpunct` 包，可以改变标点风格，把风格改变为开明式（句末全角、句中半角），也能让部分行多容纳一点字符，或可避免孤字。

```
1 | \punctstyle{kaiming}
```

代码 6.12: 标点风格

孤行：

拉丁排版中，不希望一个多行段落在一页上只有一行。以下代码会通过调整段距的方法来解决该问题。

```
1 | % 禁止在一段的正数第一二行间换页
2 | \clubpenalty=10000
3 | % 禁止在一段的倒数第一二行间换页
4 | \widowpenalty=10000
```

代码 6.13: 孤行

放在中文排版内，它能够解决多行段最后的孤行问题。对于单行段，可能发生“前文占据了更多的垂直空间，使得更多的内容放在了尾页”这样的间接解法。

评价：便于快速体验什么叫“在不同丑里找平衡”。

VII 引用

常见的引用基本可以分成两款。

§1 段落格式

适用于长引用，直接插入正文。首段缩进在环境里不起效，如果需要，请在第一行文字前加入`\hspace{\parindent}`。

长引用需要使用 `changepage` 包。

```
1 \begin{adjustwidth}{左侧 缩进量}{右侧 缩进量}  
2     % 字体 字号 等配置  
3     % 引用 内容  
4 \end{adjustwidth}
```

代码 7.1: 长引用

§2 文本框

适用于短引用，打包成文本框，作为浮动对象参与排版。

```
1 \begin{minipage}[位置]{宽度}  
2     % 字体 字号 等配置  
3     % 引用 内容  
4 \end{minipage}
```

代码 7.2: 短引用

默认支持 `htbp` 四个位置，`h` 即当前位置（here），`t` 即本页顶部（top），`b` 即本页底部（bottom），`p` 指放在一个仅有浮动对象（图和表）的页面上（page）。四个参数可多选，但优先级固定为 `htbp`。还有一个 `!` 的参数，指忽略美学指标。

VIII 注释

注释一般使用脚注和尾注两种，视情况自行选择即可。

§1 脚注

页底脚注以`\footnote[编号]{内容}`命令来添加。

```
1 | \footnote[编号]{内容}
```

代码 8.1: 添加脚注

引入 `footnpag` 包可以在每页重置脚注编号。

```
1 | \usepackage{footnpag}
```

代码 8.2: 每页重置脚注

脚注高度通常不是正文行高整数倍，会拉扯当页的行宽，可以用以下代码避免

```
1 | \usepackage[bottom]{footmisc}
```

代码 8.3: 脚注沉底

脚注所使用的字体可以自行设定，方法如下

```
1 | \usepackage{etoolbox}
2 | \makeatletter
3 |   \patchcmd{\@footnotetext}{\footnotesize}{指定字体和字号}{}{}
4 | \makeatother
```

代码 8.4: 设置脚注字体

使用以下代码令脚注使用罗马数字

```
1 | \renewcommand{\thefootnote}{\roman{footnote}}
```

代码 8.5: 脚注罗马数字

§2 尾注

建议使用 `endnotes` 包来管理尾注。

添加尾注使用`\endnote[编号]{内容}`，在文章任意地方使用`\theendnotes`命令输出尾注。第二次使用`\theendnotes`会输出第一次输出后新增的尾注。

尾注标题使用以下命令进行修改。

```
1 | \renewcommand{\notesname}{尾注标题}
```

代码 8.6: 修改尾注标题

尾注标题默认影响`\leftmark`（该命令的用处见页眉页脚），可以用以下命令规避。

```
1 | \renewcommand{\enoteheading}{\section*{\notesname}\mbox{}\par\vskip-\baselineskip}
```

代码 8.7: 尾注不改标记

尾注编号可以用如下代码重置：

```
1 | \makeatletter
2 | % 每章重置尾注编号，chapter可修改为其他标题级别
3 | \@addtoreset{endnote}{chapter}
4 | \makeatother
```

代码 8.8: 重置尾注

使用以下代码令尾注使用罗马数字

```
1 | \renewcommand{\theendnote}{\roman{endnote}}
```

代码 8.9: 尾注罗马数字

IX 目录

建议在决定了所有内容后，再来设计目录的格式。
用`\tableofcontents`命令，在文档中插入目录。

§1 目录标题

使用如下命令来调整目录页的标题内容，其格式默认与`\chapter`级标题相同。`ctex`文档类别中，目录页标题会翻译为“目录”，也可用如下命令来修改。

```
1 | \renewcommand{\contentsname}{目录标题}
```

代码 9.1: 修改目录页标题

§2 目录配置

用如下命令设置目录层深，层深数值见前文。

```
1 | \setcounter{tocdepth}{层深}
```

代码 9.2: 设置目录最大层深

目录格式配置包并不唯一，我使用的是 `titletoc`。
配置目录中标题格式的指令是`\titlecontents`。

```
1 | % 代码
2 | \titlecontents{标题名}[左距离]{上方代码}
3 | {序号格式}{无序号的标题格式}
4 | {页码格式}[下方代码]
5 |
6 | % 样例
7 | \titlecontents{chapter}
8 | [0em]
9 | {\vspace{1em}\Large\tocchapterfont}
10 | {\makebox[10mm][l]{\S\hspace{0.1em}\large\uppercase\expandafter{\romannumeral\thecontentslabel}}}}
11 | {}
12 | {\hspace{4mm}\tocpagefont\large\contentspage}
13 | [\vspace{0.5em}]
```

代码 9.3: 配置目录中标题格式

其中，无序号标题默认不出现在目录里，想要的话需要手动添加。可使用以下命令。

```
1 | \addcontentsline{toc}{级别}{标题名}
```

代码 9.4: 手动添加项进目录

目录中的标题名默认会显示在序号右、页码左，内容默认与正文中的标题名相同，可以不同，有些时候也需要不同。如注只能放在正文中的标题里，而不能出现在目录索引里。

```
1 | \chapter[索引中的名称]{正文中的名称}
```

代码 9.5: 手动设置索引中的标题名称

[索引中的名称] 留空可以使其在目录中只有节编号，而不显示节名称。

当改变页码字体时，可能会出现 `\hbox underfull` 警告，可以用以下代码改变默认的页码宽度。

```
1 | \makeatletter
2 |   \renewcommand{\contentspage}[1][\thecontentspage]{\hb@xt@{\
   |   @pnumwidth{#1}\hfil}}
3 |   \renewcommand{\@pnumwidth}{页码宽度}
4 | \makeatother
```

代码 9.6: 调整页码宽度

§3 局部目录

`titletoc` 包支持局部目录，用法如下。

```
1 | % 开始记录标题属于某个局部
2 | \startcontents[局部名称] % 这是一个key，不要加转义符，使用时保持一致即可
3 | % 结束记录某个局部的标题
4 | \stopcontents[局部名称]
5 |
6 | % 打印局部目录
7 | \printcontents[局部名称]
8 | {前缀} % 留空则使用主标题内的格式
9 | {开始层级} % 局部目录的开始层级
10 | [层深] % 局部目录的层深
11 | {目录代码} % 该目录表头的代码，如\chapter*{\contentsname}可增加一个标题
```

代码 9.7: 局部目录

§4 文档内跳转

虽然是按印刷品的标准的制作，但是为了检阅方便，还是建议使用 `hyperref` 包来启用超链接跳转。这样可以从目录直达某个标题。

超链接会带来格式的变化，可以使用一条命令来避免。

```
1 | % 启用超链接
2 | \usepackage{hyperref}
3 | % 超链接不影响格式
4 | \hypersetup{hidelinks}
```

代码 9.8: 超链接

§5 目录换页

目录较长时，可以在正文中手动向目录中增加换页符，确定目录换页的位置。

```
1 | \addtocontents{toc}{\protect\newpage}
```

代码 9.9: 手动增加换页符

X 页眉页脚

一般使用 `fancyhdr` 包来自定义页眉页脚。

§1 页面风格定义

用`\fancypagestyle`命令设计新的页面风格。

```
1 % 页面风格名为mystyle
2 \fancypagestyle{mystyle}{
3     % 清空既有设置
4     \fancyhf{}
5     % 设置页眉页脚
6     \fancyhead[位置]{内容}
7     \fancyfoot[位置]{内容}
8 }
```

代码 10.1: 定义页面风格

位置为 `[L,C,R]` (左中右) 和 `[O,E]` (奇偶数) 两个选项组成, 如 `LE` 表示偶数页的左侧, `C` 表示无论奇偶页都显示在中间。

§2 文档自动的标记

对于 `book` 类而言, 文档随着进行会自动更新两个标, 以便于在页眉页脚中使用。其中`\leftmark`记录章标记, `\rightmark`记录节标记。如果想要获得部类别的标题和序号, 需要自己手动记录。

标记的记录机制与目录中的相同, 默认使用正文中的名称, 也可专门定义索引时名称。

```
1 | \chapter[索引的名称]{正文中的名称}
```

代码 10.2: 标题在索引中的名称

`\Xmark` 默认为 `X` 序号。索引中的标题名称, 可以用重定义的方法来去除章标记中 `X` 序号. 的部分, 节标记同理。

```
1 | \renewcommand{\chaptermark}[1]{\markboth{#1}{} }
2 | \renewcommand{\sectionmark}[1]{\markright{#1}}
```

代码 10.3: 左右标记

`ℒTEX` 中没有预设记录 `part` 标题的变量, 可以自己设置

```
1 % 定义变量
2 \newcommand{\parttitle}{} 
```

```

3 |
4 | % 重载part宏
5 | \let\origpart\part
6 | \renewcommand*{\part}[2][]{
7 |     \ifx\\#1\\
8 |     \origpart{#2}
9 |     \renewcommand*\parttitle{#2}
10 | \else
11 | \origpart[#1]{#2}
12 | \renewcommand*\parttitle{#1}
13 | \fi
14 | }

```

代码 10.4: part 标记

也可以在正文中随时使用如下代码来修改左右两种标记。

```

1 | \markboth{新的左标记}{新的右标记}
2 | \markright{新的右标记}

```

代码 10.5: 手动标记

左标记默认比右标记高级，所以不能单独定义，不想影响右标记的话填入\rightmark即可。

§ 3 页面风格使用

在文档中使用\pagestyle{风格名}命令设置这一页起的风格。

使用\thispagestyle{风格名}设置当前页的风格，下一页自动还原。

fancyhdr包提供 fancy, plain, empty 几种内置风格，book 类会自动引用适配不同的页面，可以用上文的定义方法直接重定义。

fancy: 页眉外侧角落: 偶数页为\leftmark, 奇数页为\rightmark, 页脚中央: 均为\pagenumber。

plain: 页脚中央为\pagenumber。

empty: 什么都没有。

§ 4 问题处理

对于由于标题位置而补充的空页，其页面风格默认与前文相同，可能包含页眉页脚，可以通过引入 emptypage 包，使得空页面上无内容。

页眉下默认有一条横线，可以用以下命令来设置横线宽度，0 为去除。

```

1 | \renewcommand{\headrulewidth}{宽度}

```

代码 10.6: 页眉下横线

如果希望页眉页脚溢出在文字区外，可以使用以下写法：


```
1 | \fancyhfoffset[位置]{溢出值} % 用H/F这组位置标记来明确是页眉还是页脚
2 | % 单独设置页眉或页脚的溢出
3 | \fancyheadoffset[位置]{溢出值}
4 | \fancyfootoffset[位置]{溢出值}
```

代码 10.7: 页眉页脚水平溢出

XI 图片

处理图片需要的包是 `graphicx`。插入图片的常用代码如下。

§1 插入图片

```
1 % 封一个figure环境，环境内的命令不影响外部
2 \begin{figure}[图片位置]
3     % 水平居中
4     \centering
5     % 导入图片，文件名使用相对路径或绝对路径均可
6     \includegraphics[限制参数]{文件名}
7     % 设定图片标题
8     \caption{图片标题}
9 \end{figure}
```

代码 11.1: 插入图片

`figure` 的意义是进行内外隔离，为插图服务。对于一些极端情况（如全文档只有图片）可以不包这一层。

`graphicx` 图片位置默认支持 `htbp` 四个位置，`h` 即当前位置 (here)，`t` 即本页顶部 (top)，`b` 即本页底部 (bottom)，`p` 指放在一个仅有浮动对象（图和表）的页面上 (page)。四个参数可多选，但优先级固定为 `htbp`。还有一个的 `!` 参数，指忽略美学指标。如果额外引入 `float` 包，那么此处还可使用 `H` 作为参数，可以大概理解为“竭尽全力把它放在当前位置”。

导入图片的限制参数有很多，详见文档。最常用的是 `scale=` 放缩比例、`width=` 图片宽度、`height=` 图片高度，都是等比例放缩。

§2 图片标题

图片标题用 `caption` 这个包来风格化，用法如下。

```
1 % 导入包
2 \usepackage{caption}
3 % 设置
4 \captionsetup{
5     % key=value类配置，如
6     font=small, % small号字体
7     labelformat=empty, % 隐藏编号
8 }
```

代码 11.2: 图片标题

§3 旋转

`graphics` 内置了一个旋转功能，使用方法如下。

```
1 | \rotatebox{逆时针旋转角度（角度制）}{旋转内容}
```

代码 11.3: 旋转

顾名思义，就是把内容打包成一个盒子，然后转一下。

XII 其它

§1 规定标点宽度（如破折号）

绝大多数中文标点都能在 $\text{\LaTeX}$ 中按标准方式渲染，但是破折号基本都不会和标准使用方法一样，等同于两个字宽。这与字体和连字均有关系，有兴趣可以阅读这篇帖子¹。想完美解决很复杂，但是想简单解决真的很简单。

```
1 | % 直接把破折号用盒子框起来，默认居中
2 | \makebox[2\ccwd]{——}
3 | % rule命令画线，可以自行根据字体设计破折号的高度、粗细、长短。
4 | \makebox[2\ccwd]{\rule[0.35em]{1.6\ccwd}{0.03em}}
```

代码 12.1: 破折号

使用中，可以将实现方式定义为一个命令，然后把正文中所有破折号替换为该命令，一键解决。

注：`\mycommand`文字会发生识别错误，应该使用 `{\mycommand}`文字的写法。

§2 特殊字符换字体

对于个别字符，你可能不满意它的默认字体，（例如问号）。可以用如下写法进行修改。给它起名`\mysymbol`，在全文中将中文字符替换成 `{\mysymbol}`，这里的大括号是为了防止文字粘连。

设定命令，让`\mysymbol`用指定字体渲染。注意大括号需要有两层，限制字体应用范围。

```
1 | \newcommand{\mysymbol}{{\symfont 符号}}
```

代码 12.2: 专门字体的符号

§3 行溢出

行溢出，即 `Overfull \hbox` 警告。一行文字无法恰当地显示在规定的行宽里。

从本质上来说，警告并不是一定要消除，因为它不影响功能的实现。但是 $\text{\LaTeX}$ 中的警告与丑基本可以画等号，所以排版过程中还是建议解决一下。

调整字号、行宽等基础布局数值，都可能在某些页面产生或消除该警告。

可以增加一个 `sloppypar` 域来一劳永逸，即

```
1 | \begin{document}
2 |     \begin{sloppypar}
3 |         内容
```

¹有兴趣可以阅读这篇帖子：<https://github.com/CTeX-org/tex-kit/issues/382>

```

4 | \end{sloppypar}
5 | \end{document}

```

代码 12.3: 行溢出的处理

但这样的解决方法会导致某一行字距过大，还是影响观感。最美观的解决方法应该是直接调整文本，在确定页面参数后，通过增减这一行的几个字或标点来解决问题。不过因为只能在给自己排版时进行改动，所以大部分时候就只能接受一点点的不美观了。

§4 纸面不充盈

即 `Underfull \vbox` 警告。该警告出现是因为某页前一部分内容不足以填充页面，后一部分内容所需空间又过多，致使页面无法恰当地加载。

调整行宽、行高、段距等基础布局数值，都可能在某些页面产生或消除该警告。

调整文本内容当然可以优雅地消除警告。除此之外，还可以在合适的地方进行手动换页（`\clearpage` 或 `\newpage`）解决问题。但由于这样的手动解决方式不是很优雅，所以应当在排版过程的最后进行此调整。

§5 扉页

虽然可以选择用 `\maketitle` 命令生成一页现成的扉页，但是真的很丑。中文出版物的扉页通常是封面的黑白朴素版，内外呼应，我个人推荐按此办理。根据封面的设计来手动布局，或直接导出一页适于黑白印刷的封面 pdf 文件（注意出血），插入正文开头作为扉页。

```

1 | \usepackage[pdftypes]
2 | \begin{document}
3 | \includepdf[
4 |     pages={文档页码，以英文逗号连接},
5 |     angle=逆时针方向旋转角度
6 | ]{文件名}
7 | \end{document}

```

代码 12.4: 导入扉页

§6 子文件

善用子文件可以便于管理，实现格式与内容的分离，以及无关内容彼此分割。

最简单的方法是用 `\input` 命令，将另一个 `.tex` 文件引入当前文件，效果相当于用该文件的全部内容替换 `\input` 这一行。

```

1 | % 子文件，head.tex
2 | \documentclass[10pt]{book}
3 | \usepackage{hyperref}
4 | \hypersetup{hideliinks}
5 |

```

```

6 | % 主文件, main.tex
7 | \input{head}

```

代码 12.5: 导入子文件

对于较厚的本, 可以使用`\subfiles`包来分割出子文件, 该包可以在不影响整体编译的情况下, 独立编译子文件。由于换页的位置会直接影响效果, 每个子文件的内容, 在主文件中都应当开始于新的一张纸, 至少新的一页。

```

1 | % 以main.tex为主文档
2 | % 在同层级的Chapter文件夹内, 创建ch1.tex为子文档
3 |
4 | % 子文档
5 | \documentclass[../main]{subfiles}
6 | \begin{document}
7 | 文档内容
8 | \end{document}
9 |
10 | % 主文件, main.tex
11 | \usepackage{subfiles}
12 | \begin{document}
13 | \subfile{Chapter/ch1}
14 | \end{document}

```

代码 12.6: 分割子文件

对于一些在文档内使用生效的命令, 如`\pagestyle`, 需要在子文件的文档内容里和主文件引入子文件前都使用, 才能达到相同的渲染效果。

在子文档中, 页码自动从1开始。可以手动设置, 但不建议这样做, 因为手动调整页码的命令在主文件内同样生效, 容易产生页码不连续的问题。

`subfiles`包有时会出现一些 bug, 如果主文件和子文件渲染出来有页码之外的区别的话, 大概就是原生 bug 造成的。如果调整不出来, 可以釜底抽薪, 删除开头结尾的 `document` 相关声明, 改用`\input`引入。

§7 更多文字装饰

`ulem`包提供了更多的文字装饰方法, 如下。

```

1 | \uline{下划线}
2 | \uuline{下划两条线}
3 | \uwave{下划波浪线}
4 | \sout{删除线}
5 | \xout{划掉内容}
6 | \dashuline{下划虚线}
7 | \dotuline{下划点线}

```

代码 12.7: `ulem` 装饰

下划线 下划两条线 下划波浪线 删除线 划掉内容 下划虚线 下划点线

注意, `ulem` 包会默认将 `\emph` 命令的效果替换为下划线, 每强调一次, 强调内容下方就会多一条线。可以在引入时规避。

```
1 | \usepackage[normalem]{ulem}
```

代码 12.8: 不允许 `ulem` 改变强调格式

§8 色彩

一般用 `xcolor` 处理色彩。

先定义颜色。

```
1 | % 颜色格式为RGB、HSB、CMYK etc
2 | \definecolor{颜色名称}{颜色格式}{颜色参数}
3 | % 样例
4 | \definecolor{purple}{RGB}{189, 147, 249}
```

代码 12.9: 定义颜色

用 `\color` 设置字体颜色。

```
1 | \color{颜色名称}
```

代码 12.10: 使用颜色

用 `\colorbox` 绘制彩色背景盒子。

```
1 | \colorbox{颜色名称}{文本内容}
```

代码 12.11: 彩色背景盒子

§9 输入 EMOJI

emoji 分单色和多色两种, 有不同字体, 使用方案不同。

i 单色 emoji

单色 emoji 的使用方法近似于一般文字。🐼 上色后即成为 🐼。✅ 则为 ✅ 放入彩色背景盒子后得到的。

加载单色 emoji 字体后, 用 `\char` 或 `\symbol` 输入编码使用。emoji 的 unicode 编码可以在 emoji 字典¹上查询。

典型的单色 emoji 字体如 `Symbola`²。

```
1 | % 编码前一个英文双引号为十六进制的标志
2 | \fontspec{Symbola}{\symbol{"十六进制编码}}
3 | \fontspec{Symbola}{\char{"十六进制编码}}
```

代码 12.12: 单色 emoji

¹emoji 字典: <https://www.emoji4all.com/>

²Symbola: <https://www.webfontfree.com/cn/download/Symbola>

ii 多色 emoji

多色 emoji 使用 `emoji` 包处理。只能使用 LuaL^AT_EX (LuaLaTeX) 编译。
典型的多色 emoji 字体如 Twemoji¹。

```

1 % 引入包
2 \usepackage{emoji}
3 % 设置一个普适的emoji字体
4 \setemojifont{多色emoji字体}
5
6 % 使用emoji
7 \emoji{名称}

```

代码 12.13: 多色 emoji

每个 emoji 的名称见 `emoji` 包文档。

¹Twemoji: <https://github.com/jdecked/twemoji/releases/latest>

XIII 严格网格排版（进阶）

严格网格，能让版式像排字印刷时一样精致。

不推荐玩这个，哪怕 pdf 做得再齐，印厂也很可能给你印歪一两毫米。只是，如果你想要一份电子书给自己收藏的话，那它还是很酷的。

§1 行高段距设定

下文教程将以`\baselineskip`（行高）作为纵向的基本单位，所以第一步是将行高调整的命令放到序言区开头。

```
1 | \setstretch{行高倍数}
```

代码 13.1: 进阶行高

段距设定为行高的整倍数，不留弹性。

```
1 | \setlength{\parskip}{0pt}
```

代码 13.2: 进阶段距

§2 版心规划

可以用 `em`（单字宽度，不考虑字距）或`\ccwd`（单字宽度，考虑字距）作为单位来定义版心的宽度（`textwidth`），使得每行严格形成网格。不自行调整字距的话，这两个单位是一样的。

版心高度（`textheight`）应为规划行数 * 行高 - 行距， $\text{\TeX}$ 是可以算数的。

```
1 | % 建立长度变量
2 | \newlength{\theight}
3 | % 输入行数
4 | \setlength{\theight}{期待行数\baselineskip}
5 | % 删除一个多余的行距
6 | \addtolength{\theight}{1em}
7 | \addtolength{\theight}{-\baselineskip}
8 | % 计算好后定义版式
9 | \geometry{
10 |     % ...
11 |     textwidth=每行字数\ccwd,
12 |     textheight=\theight,
13 |     % ...
14 | }
```

代码 13.3: 进阶版心

§3 正文对齐

上面设完之后，纯正文的页面将不会出现任何警告。但各级标题（加上上下留白）的高度几乎不可能是行高的整数倍。

使用现成字号方案的话，由于不同主字号下，各级字号的大小没什么关系，这里无法给出普适性的精确方案。（需要的话，建议自己算）但可以给非正文区域上下添加弹性空白。如果一页纸上只有正文和单个空白弹性的块，且正文溢出到下页的话，这个做法可以令弹性空白很精确。

```

1 | {基础长度 plus 扩张长度 minus 压缩长度}

```

代码 13.4: 进阶弹性空白

更朴实的方案是把所有非正文用盒子或 `minipage` 括起来，固定其所占高度。

```

1 | \begin{minipage}[对齐标准, 建议用c][所占高度]{所占宽度}
2 |     内容
3 | \end{minipage}

```

代码 13.5: 进阶标题

XIV 篇末谈

§1 为什么写这篇文章？

因为我喜欢 $\text{\LaTeX}$ 。

大约在 1977 年的一天，高德纳 (*D. E. Knuth*) 或许是收到了出版社寄来的可怕样书，或许是在写完《计算机程序设计的艺术》第三卷《排序与搜索》的手稿时意识到电脑打不出这东西，总之，他深感当时的排版软件太烂了，配不上他的才华，所以选择搁置写作，给自己开发一款高质量的排版工具。1983 年 6 月， $\text{\TeX}$ 完工。1985 年，*Leslie Lamport* 给 $\text{\TeX}$ 套了一份宏，简便易用的 $\text{\LaTeX}$ 从此诞生。

这是 $\text{\TeX}$ 系列的精神内核，不是因为利益，而是因为需求。因为我需要，而且我相信别人也需要。整个 $\text{\TeX}$ 社区就是基于这样的心态所建立的，我这篇文章也是同理。

§2 为什么不用 $\text{\Typst}$ ？

因为 $\text{\Typst}$ ¹ 还是太菜了。

当然，这样说粗糙了些。具体地讲，放弃 $\text{\Typst}$ 有以下几个原因。

i 中文处理能力不足

具体地说，在几个字符范围内的处理， $\text{\Typst}$ 已经把能做的都做了，但以段为单位的中文处理能力还距离出版级较为遥远。

细心的你或许一早就发现了我讲标点压缩和字距时留下的暗坑。

标点压缩：

连续的两个标点（如：“”）合并挤压为 1.5 个字符宽

字距：

文字每行左右均顶格，恰好占满设定的行宽

那么问题来了，如果我行宽设定为 24 个字宽，某行的内容是兔子草灵感迸发，提笔写道：“此处应有一段广告，但，那么它应该占多少格呢？”

答案是 24 个。虽然只看文本，这一行应为 23.5 个字宽，但它既然是一行，就必须是 24 个字宽。为了美观，左右对齐的权重要重于标点压缩的正确性。可 $\text{\Typst}$ 目前是渲染为 23.5 个字宽。

（通常这里的解法有：把连用的标点扩到 2 个字宽；把三处标点各压一点，挤出半个字宽，多放一个字在这行；把整行字距增大一点点，凑出半个字宽 et al.）

¹ $\text{\Typst}$: <https://typst.app/>

如果说半个字宽你还觉得影响不大的话，请注意，一段长文字必然要换行，而大部分标点不可以出现在行首，但是又要占一格。自然换行处是标点的话，*Typst* 会从上一行抠一个字塞进下一行，使得前一行末尾出现整个字的空缺。

结果就是：随便粘几百字有对话的小说进 *Typst*，右缘就稀烂了。如果文本是多语言夹杂的，这个问题还会更进一步放大。

ii 格式转换路径空缺

我写文一般用 markdown，当下最简单的富文本格式。而就这么轻量级的语言，目前也没有 md2typ 的自动转化工具。注释、图片、表格这堆东西，要我自己动手抠格式，我是真的不动。

iii 功能欠缺

以下是我比较在意的几个问题：

1. 没有尾注功能，只有脚注和引用。
2. 加粗/斜体等格式中，如果发生字体缺失，自动用默认字体渲染，甚至不报警告。
3. 没有删除线。
4. 无标号时，无法实现文件内跳转。
5. 很难按左右页设置页眉页脚。

§ 3 为什么不用 INDESIGN?

因为作为外行，我在研究 *InDesign* 时，只感到它特别愚蠢。

我无法理解各层 UI 目录的嵌套设计原则，出于同样的原因，我在阅读文档时也感到了极大的困扰。

我更不能接受它糟糕的多语言支持：中文 UI 的翻译甚至连断字 (hyphenation) 和连字 (ligature) 都分不清。

我最无法接受的是它的性能。我只放了三页文档，调格式的时候都能明显感到卡顿。甚至有几次调完格式它原地崩溃了。我的电脑配置是 i5-12400F，32G 双通道内存，4060ti，还配得上三页只有字的 A5 纸吧。

§ 4 ℒ_AT_EX 怎么念?

T_EX 虽然长得很像英语 (该符号也是用英语字母绘制)，但其实是希腊词根 τ_εχ (依次为 tao, epsilon, chi) 的大写 TEX，具有 technology 和 art 双义，读作“泰-赫”。

ℒ_AT_EX 的 La 则是 *Lamport* 的缩写，它读作“拉-泰-赫”。这个读音和 *Lamport* 的读音没啥关系，多半只是为了继承 T_EX 反直觉的读法，故意取头两个字母，以让名字与乳胶 (latex) 长得很像。

§ 5 补充阅读

本文旨在快速上手排版，因而跳过了大量命令解释和高级语法的内容。如果想要深度使用 ℒ_AT_EX，仍需补课。

以下是几份我认为值得一读的参考,

*LaTeX 知识库*¹是一套非常基础的文档,对于各个基本功能都进行了详细讲解。适于完整阅读,以初步了解 $\text{\LaTeX}$ 。

*lshort-zh-cn*²是真正意义上的 $\text{\LaTeX}$ 介绍文档 *lshort* 的中文翻译版本。内容翔实丰富,基本覆盖了一切用户所需要知道的内容。适于想要认真使用 $\text{\LaTeX}$ 的用户仔细阅读。

*Overleaf Document*³是一套纯英文的进阶文档,基本涵盖了所有用得上的功能。适于比较着急的人,针对性阅读能解决自己当下问题的章节段落即可。

T_EXbook 是高德纳写的 $\text{\TeX}$ 指南,有中文翻译⁴。建议至少读完前言和第一章“此名有诗意”,可以更好地理解 $\text{\TeX}$ 的功能和目的;继续阅读,则能学到很多不知道用不用得上的知识。但,我建议等 $\text{\LaTeX}$ 不够你玩了的时候,再进行深入学习。

¹*LaTeX 知识库*: <https://ask.latexstudio.net/LearnLaTeX/>

²*lshort-zh-cn*: <https://ctan.math.utah.edu/ctan/tex-archive/info/lshort/chinese/lshort-zh-cn.pdf>

³*Overleaf Document*: <https://www.overleaf.com/learn>

⁴有中文翻译: <https://github.com/seuliang/the-texbook-cn>