

# 一份 L<sup>A</sup>T<sub>E</sub>X 学习总结

Riven Russell

2024/02/16

# 目录

I	认识 L <sup>A</sup> T <sub>E</sub> X	1
1.1	L <sup>A</sup> T <sub>E</sub> X 的起源	1
1.2	L <sup>A</sup> T <sub>E</sub> X 的优势与劣势	1
II	文档结构介绍	3
III	常见宏包介绍	4
IV	常见环境介绍	6
V	页面与页眉页脚	8
5.1	页面设置	8
5.2	页眉页脚设置	8
VI	文字与段落	9
6.1	全局字体类型设置	9
6.2	全局字号设置	10
6.3	全局段落格式设置	10
6.4	局部设置	11
6.5	换行操作	11
6.6	对齐方式	11
VII	表格和图片	12
7.1	插入表格	12
7.2	插入图片	16
VIII	代码块插入	18
IX	目录、标题和标签	21
9.1	目录设置	21
9.2	标题设置	21
9.3	标签与跳转	22
X	序列的插入	23
XI	公式编辑	24
XII	参考文献插入	25

## 第 I 部分 认识 L<sup>A</sup>T<sub>E</sub>X

本部分内容主要来自于编程爱好者协会的[L<sup>A</sup>T<sub>E</sub>X 汉语文档教程](#), 采用的环境为 TeX Live2023+TeX Studio, 二者均为开源软件, 使用搜索引擎搜索下载即可。

### 1.1 L<sup>A</sup>T<sub>E</sub>X 的起源

L<sup>A</sup>T<sub>E</sub>X 和 T<sub>E</sub>X 都是用于排版的语言。T<sub>E</sub>X 及其发展出的语言都是指令驱动的, 用于生成 pdf 文件。这些语言的基础都在于宏控制序列, 开发者可以通过编写代码来进行精确、高度自动化同时高度可定制的排版任务。无论是字体字号还是段落格式, 随笔文集还是论文, 都可以用 L<sup>A</sup>T<sub>E</sub>X 中高度集成化的指令设计解决。此外, L<sup>A</sup>T<sub>E</sub>X 可以自动为您的各种题注进行编号, 并方便地在后文中引用, 这样您就不需要在编号变动后一个个手动改动引用了, 这也意味着出错的几率大大降低。

T<sub>E</sub>X 的出现很早, 可追溯至 HTML 标记语言出现之前。L<sup>A</sup>T<sub>E</sub>X 继承和发展了 T<sub>E</sub>X, 可以简单地认为 T<sub>E</sub>X 相对“朴实无华”而 L<sup>A</sup>T<sub>E</sub>X 内置了众多常用宏。对本文读者而言只需要知道一件事: L<sup>A</sup>T<sub>E</sub>X 相比一般的 T<sub>E</sub>X 更加易用, 适合快速上手。本文接下来只讨论 L<sup>A</sup>T<sub>E</sub>X 而不再讨论 T<sub>E</sub>X。

### 1.2 L<sup>A</sup>T<sub>E</sub>X 的优势与劣势

L<sup>A</sup>T<sub>E</sub>X 在排版工作上几乎无所不能。使用社区提供的宏包, 它能完成几乎任何常见排版需求。对选择 L<sup>A</sup>T<sub>E</sub>X 顾虑最大的人群恐怕是 Microsoft Office Word 的用户。诚然 Markdown 和 HTML&CSS 都可用于排版, 但是 Markdown 和 TML&CSS 的业务还是主要集中于网络界面。Markdown 也可以做文档排版, 但是就笔者的经验而言, Markdown 使用上虽然简易, 但是很多自定义功能实现反而比 L<sup>A</sup>T<sub>E</sub>X 复杂一些。而在离线程序文档中, Markdown 写就的 README 和 L<sup>A</sup>T<sub>E</sub>X 就更没有什么比较意义了。所以, 笔者并不认为 Markdown 和 L<sup>A</sup>T<sub>E</sub>X 有太多的业务重叠。也有读者可能会使用方正和 Adobe 的产品, 但那些产品通常是收费的 (当然, Microsoft Office 也是收费的。您也可以使用某些软件的免费版, 但相比之下 L<sup>A</sup>T<sub>E</sub>X 中不存在免费版比收费版更差的情况), 而且主要面向专业的排版厂商, 个人开发者使用那种体量的企业级产品实在是没有必要。我们这里主要阐述 L<sup>A</sup>T<sub>E</sub>X 相比 Microsoft Office Word 的优劣。

相比于 Microsoft Office Word 的 GUI 为核心的设计, 在 L<sup>A</sup>T<sub>E</sub>X 下您一般不需要特别考虑把图片放在哪, L<sup>A</sup>T<sub>E</sub>X 会尝试自己解决这件事。如果对它的工作不满意, 也可以通过参数自行指定。相比 Word 下需要点来点去拖来拖去, L<sup>A</sup>T<sub>E</sub>X 只需要您已经写好的代码就能得到最终结果, 要修改效果也只需要简单地改写代码, 不需要打开额外的属性页面。

L<sup>A</sup>T<sub>E</sub>X 另一大亮点是集成包管理系统, 只需要引入几个包敲几行代码就能生成漂亮的数学公式或是化学方程式, 物理论文或是生物论文自然也可以。如果需要插图, L<sup>A</sup>T<sub>E</sub>X 也有一些常见的绘图软件, 只要需要的示意图不是太复杂, 您就能用 L<sup>A</sup>T<sub>E</sub>X 把它画出来。

相比于传统讲义设计中繁琐的示意图绘制、公式插入和特殊符号插入，您在  $\text{\LaTeX}$  下只需要几行代码。当然，这些代码需要用户自行编写。

传统计算机工程领域讲义编写中，代码高亮常常是通过截图或第三方工具实现的。这一切在  $\text{\LaTeX}$  社区提供的宏包面前都黯然失色。在  $\text{\LaTeX}$  下，您所需要的只是引入包，设置高亮格式，把代码复制进来。如果愿意，您甚至可以自己实现一门语言及其在  $\text{\LaTeX}$  下的高亮。哪怕拥有 `yacc` 工具的用户也需要自己编写代码才能实现这样的自定义语言高亮，但在  $\text{\LaTeX}$  下，用户只需要使用提供好的工具，甚至没有必要真的把这门语言实现出来。

然而， $\text{\LaTeX}$  是诞生于 CLI 时代的工具（当然也有 IDE，但底层仍然是 CLI），提供的图形化操作的用户而言， $\text{\LaTeX}$  可能不是太美好。而且， $\text{\LaTeX}$  并不是所见即所得的工具，所以您需要不断改写和编译您的源码才能得到期望的文档。但是，如果您真的是本文所面向的计算机领域工程师，我有理由相信您是乐于折腾手头的工具和设备的 `geek`——否则恐怕不会选择这个行业。如果果真如此的话，有理由相信  $\text{\LaTeX}$  能成为您今后的最佳伴侣。

## 第 II 部分 文档结构介绍

L<sup>A</sup>T<sub>E</sub>X 代码文档主要分为两部分——导言区和编辑区，导言区主要用于设置相应的全局设置，编辑区则是文档内容的编写，代码示例如下：

```
1 % 导言区
2 \documentclass{article}
3 \usepackage{ctex}      % 导入中文编写宏包
4 \usepackage{float}     % 掌管各类浮动体的宏包
5
6 %文档编辑区
7 \begin{document}
8
9     % 进行文本的编辑
10
11 \end{document}
```

### 第 III 部分 常见宏包介绍

L<sup>A</sup>T<sub>E</sub>X 实际上是以宏为基础的语言，所以严格而言，“包”在这里应当叫做宏包。宏以`\def`命令定义，此外，`\newenvironment`、`\newcommand` 和 `\renewcommand` 分别可以定义新环境、新宏和修改特定宏的作用。作为一篇简明教程，这里我们只进行“脚本小子”式的教学，讲述一些宏包的功能介绍，而略去关于宏本身的内容。

宏	介绍
<code>amsmath</code>	提供了一系列功能强大的工具，用于编写复杂的数学公式和数学结构
<code>amsthm</code>	提供了定义和排版定理、引理、证明等数学结构的功能
<code>ctex</code>	为中文文档提供了完善的支持，包括中文字符的处理、中文文档类、章节标题格式等
<code>ifthen</code>	提供了条件判断命令，可以在文档中根据不同条件执行不同的操作。
<code>titlesec</code>	用于自定义和调整章节标题的格式和样式，包括字体、间距、对齐方式等。
<code>titletoc</code>	用于定制目录的样式和格式，包括添加额外的目录内容、修改目录标题样式等。
<code>SIunits</code>	提供了一套用于输入和排版科学单位的命令，以确保单位的一致性和标准化。
<code>tikz</code>	强大的绘图工具，可以绘制各种类型的图形、图表和图示，支持高度自定义。
<code>extarrows</code>	提供了一些额外的箭头符号，用于数学公式和图表中的指示和标记。
<code>indentfirst</code>	自动缩进文档中每个段落的第一行。
<code>geometry</code>	用于设置页面布局和页边距，可定制页面的大小、边距、页眉页脚等。
<code>multirow</code>	提供了在表格中创建跨行单元格的功能。
<code>fancyhdr</code>	用于自定义页面的页眉和页脚，可以添加页眉页脚内容和样式。
<code>lastpage</code>	提供了一个命令，用于获取文档的总页数，方便在文档中引用。
<code>layout</code>	显示当前页面布局的详细信息，包括页面尺寸、边距等。
<code>listings</code>	用于排版代码清单，支持多种编程语言的语法高亮和格式设置。
<code>xcolor</code>	用于设置文档中的颜色，支持各种颜色模型和色彩空间。
<code>multicol</code>	用于创建多栏布局，可以在文档中同时显示多列内容。
<code>subcaption</code>	用于支持子图和子表格，并提供了一些相关的命令和环境。
<code>graphicx</code>	用于插入和处理图片，支持多种图片格式，并提供了一些图片调整和处理的命令
<code>algoritm2e</code>	用于排版算法和伪代码，提供了一些用于排版算法的命令和环境
<code>dirtree</code>	用于绘制目录结构的树状图，方便显示文件和文件夹的组织结构。

menukeys	用于排版键盘快捷键和菜单，支持自定义快捷键的样式和格式。
fontspec	提供了对字体的高级控制功能，支持使用系统安装的字体，并提供了一些字体设置命令。
fontenc	用于指定字体编码，影响文档中字符的显示，常用于支持特定字符集。
tipa	提供了一些国际音标的支持，用于排版国际音标符号。
metalogo	用于排版各种 Logo，如 LaTeX、TeX 等。
hyperref	用于创建超链接，可以在文档中添加链接到网页、章节、图片等的交互式链接。
textgreek	提供了使用希腊字母的命令，方便在文档中输入希腊字母符号。
chemfig	用于绘制化学结构式，支持排版复杂的化学分子结构。
mhchem	用于排版化学方程式，提供了一套用于输入和排版化学式的命令和环境。
array	提供了增强的表格功能，包括更灵活的列格式和表格样式设置。
float	提供了对浮动对象（如图片和表格）的控制，可以设置浮动对象的位置和样式。
circuitikz	用于绘制电路图，支持绘制各种类型的电路图元件。
pgfplots	用于绘制高质量的图表，支持绘制二维和三维图表，并提供了丰富的绘图功能。
nomenc	用于生成术语表，可以方便地生成和管理文档中使用的术语列表。
glossaries	用于生成术语表和缩略词表，支持自动索引和排序术语，并提供了一些样式设置选项。
longtable	提供了跨页的长表格功能，使得在文档中排版大型表格时更加灵活和方便。

## 第 IV 部分 常见环境介绍

```
1 % 正文区（唯一）
2 \begin{document}
3 \end{document}
4
5 % 图标区
6 \begin{figure}[H]
7 \end{figure}
8
9 % 表格区
10 \begin{table}[H]
11 \end{table}
12
13 % 标题页
14 \begin{titlepage}
15 \end{titlepage}
16
17 % 也是表格区
18 \begin{tabular}
19 \end{tabular}
20
21 % 长表格
22 \begin{longtable}
23 \end{longtable}
24
25 % 左对齐
26 \begin{flushleft}
27 \end{flushleft}
28
29 % 居中
30 \begin{center}
31 \end{center}
32
33 % 右对齐
34 \begin{flushright}
35 \end{flushright}
36
37 % 自定义区
38 \begin{<自定义的环境名称>}
```



39 `\end{<自定义的环境名称>}`

值得注意的是，环境是可以自己定义的，比如`\def`和`amsthm`宏包中的`\newtheorem`命令都可以定义新的环境。

## 第 V 部分 页面与页眉页脚

### 5.1 页面设置

```
1 \documentclass[a4paper]{article}      % 设置纸张大小未A4纸张
2 \usepackage{geometry}                 % 导入页面格式设置的代码
3 % 下面是两种设置页边距的方式
4 \geometry{left=2.5cm,right=2.5cm,top=2.5cm,bottom=2.5cm}
5 \geometry{margin=2.5cm}
```

### 5.2 页眉页脚设置

```
1 \usepackage{fancyhdr}                % 自定义设置页眉页脚的宏包
2 \footskip = 10pt                     % 设置页脚边距
3 \renewcommand{\headrulewidth}{1pt}   % 设置页眉线的宽度
4 \renewcommand{\headwidth}{16cm}      % 设置页眉线的长度
5 \pagestyle{fancy}                    % 将全局页眉页脚设置为fancy格式
6 \thispagestyle{empty}                % 删除该页的页眉页脚
7 \lhead{一份学习文档}                 % 设置页眉左侧
8 \chead{nihao}                        % 设置页眉中间
9 \rhead{我不知道写啥}                 % 设置页眉右侧
10 \cfoot{\thepage}                    % 设置页脚中间 (\thepage表示当前
    页码)
11 \rfoot{}                             % 设置页脚右侧
12 \lfoot{}                             % 设置页脚左侧
```

值得注意的是，页眉页脚的格式在 L<sup>A</sup>T<sub>E</sub>X 中共存在四种模式：

1. empty: 页面不显示页眉页脚。
2. plain: 页面显示页脚，但页眉为空。
3. headings: 页面显示页眉和页脚，页眉中包含章节标题或节标题。
4. fancy: 这是导入 fancyhdr 宏包后可以使用的格式，即自定义页眉页脚。

## 第 VI 部分 文字与段落

这里首先要明确一下再 latex 中的各类长度单位，具体如下所示：

pt	点阵宽度，1/72.27in
bp	点阵宽度，1/72in
in	点寸
cm	厘米
mm	毫米
em	当前字号下大写字母 M 的宽度，常用于水平距离的设定
ex	当前字号下小写字母 m 的高度，常用于垂直距离的设定

### 6.1 全局字体类型设置

```
1 \usepackage{fontspec} % 自定义字体宏包
2
3 % 中文字体设置
4 \setCJKfamilyfont{Songti}{宋体} % 导入本地宋体字体
5 \setCJKfamilyfont{Kaiti}{楷体} % 导入本地楷体字体
6 \setCJKfamilyfont{Heiti}{黑体} % 导入本地楷体字体
7 \newcommand{\Song}{\CJKfamily{Songti}} % 设置\Song为宋体的指
   令，方便下文的局部使用
8 \newcommand{\Kai}{\CJKfamily{Kaiti}} % 设置\Kai为楷体的指
   令，方便下文的局部使用
9 \newcommand{\Hei}{\CJKfamily{Heiti}} % 设置\Hei为黑体的指
   令，方便下文的局部使用
10 \setCJKmainfont{宋体} % 设置全局默认字体
    为宋体
11
12 % 英文字体设置
13 \newfontfamily{\TimesNewRoman}{Times New Roman} % 导入本地Times New
    Roman字体
14 \setmainfont{Times New Roman} % 设置全局
15
16 % 设置一个字体复位的命令
17 \newcommand\resetfont{\Song \TimesNewRoman} % 方便字体变更后一键
    复位
```

## 6.2 全局字号设置

```
\documentclass[12pt]{article} % 全局字体为12磅，即小四号字体
```

字体大小对照换算表

字号	磅数	毫米	英寸	像素	宋体	黑体	楷体
初号	42	14.8	0.58	56	宋体初	黑体初	楷体初
小初	36	12.7	0.50	48	宋体小初	黑体小初	楷体小初
一号	26	9.2	0.36	34	宋体一号	黑体一号	楷体一号
小一	24	8.5	0.33	32	宋体小一	黑体小一	楷体小一
二号	22	7.8	0.31	29	宋体二号	黑体二号	楷体二号
小二	18	6.3	0.25	24	宋体小二	黑体小二	楷体小二
三号	16	5.6	0.22	21	宋体三号	黑体三号	楷体三号
小三	15	5.3	0.21	20	宋体小三	黑体小三	楷体小三
四号	14	4.9	0.19	18	宋体四号	黑体四号	楷体四号
小四	12	4.2	0.17	16	宋体小四	黑体小四	楷体小四
五号	10.5	3.7	0.15	14	宋体五号	黑体五号	楷体五号
小五	9	3.2	0.13	12	宋体小五	黑体小五	楷体小五
六号	7.5	2.6	0.10	10	宋体六号	黑体六号	楷体六号
小六	6.5	2.3	0.09	8	宋体小六	黑体小六	楷体小六
七号	5.5	1.9	0.08	7	宋体七号	黑体七号	楷体七号
八号	5	1.8	0.07	6	宋体八号	黑体八号	楷体八号

pt 磅或点数，是point简称 1磅=0.03527厘米=1/72英寸

inch 英寸，1英寸=2.54厘米=96像素（分辨率为96dpi）

px 像素，pixel的简称（本表参照显示器96dpi显示进行换算。像素不能出现小数点，一般是取小显示）

GSDN @ 酒逢知己千杯少

## 6.3 全局段落格式设置

```
1 \usepackage[indentfirst] % 加载
   indentfirst 宏包，用于使得文档中所有章节后的第一个段落也进行缩进。
2 \setlength{\parindent}{2em} % 设置段落的缩进
   为 2em，即段落开头缩进 2 个字符宽度。
3 \setlength{\leftskip}{1em} % 将段落左侧的距离
   离设置为 1em
4 \setlength{\rightskip}{1em} % 将段落右侧的距离
   离设置为 1em
5 \setlength{\parskip}{1ex plus 0.5ex minus 0.2ex} % 设置段落之间的
   垂直距离为 1ex，允许在需要时拉伸至多 0.5ex 或压缩至多 0.2ex
6 \fontsize{12}{14} % 一个设置字号和
   行间距的命令（但是一般不使用）
7 \linespread{1.25} % 将行距设置为当前
   字号的 1.25 倍。
```

这个地方需要注意的是`\linespread` 命令，他后面的参数是基础行距的倍数，而基础行距在不使用`\fontsize` 指令更改的情况下即为当前字号的大小。

## 6.4 局部设置

```
1 \noindent{无首行缩进的一段}  
2 {\linespread{2.0}\selectfont 两倍行间距的一段\par}  
3 \vspace{6pt}           % 插入垂直间距  
4 \hspace{6pt}           % 插入水平间距  
5 {\SinSun \zihao{-4} 小四宋体的字体\par}  
6 \textbf{这段文字将被加粗}  
7 \textit{这段文字将倾斜}  
8 \usepackage{ulem}  
9 \uline{这段文字将加入下划线}
```

## 6.5 换行操作

在 L<sup>A</sup>T<sub>E</sub>X 中，常见的换行操作有 4 种：

1. `\`和`\newline` 实现段内换行
2. 一行空行和`\par` 实现段落的切换

## 6.6 对齐方式

1. `\raggedright`：左对齐
2. `\raggedleft`：右对齐
3. `\centering`：居中

## 第 VII 部分 表格和图片

### 7.1 插入表格

```
1 \usepackage{array,multirow}           %表格宏包和行合并宏包
2
3 % 定义了三种新的列类型：左对齐、居中对齐和右对齐。这些列类型可以在表
  格中指定列宽
4 \newcolumntype{L}[1]{>{\raggedright\let\newline\\\arraybackslash\
  hspace{0pt}}m{#1}}
5 \newcolumntype{C}[1]{>{\centering\let\newline\\\arraybackslash\hspace
  {0pt}}m{#1}}
6 \newcolumntype{R}[1]{>{\raggedleft\let\newline\\\arraybackslash\
  hspace{0pt}}m{#1}}
7
8 % 表格环境
9 \begin{table}[H]
10     \centering                         % 表格整体居中
11     \caption{\SimHei\zihao{-5}一张随意画的\LaTeX 表格} % 表格的标题
12     \begin{tabular}{|C{2cm}|C{2cm}|C{2cm}|C{2cm}|} % 设置四列表格，
      每一列两厘米宽
13         \hline                         % 添加横线
14         (0,0) & \multicolumn{2}{c|}{row 0} & ~ \\ % 列合并
15         \hline
16         \multirow{2}{*}{column 0}      % 行合并
17         & \multicolumn{2}{c|}{\multirow{2}{*}{center}} & ~ \\
18         \cline{4-4}                     % 在指定位置添加
      横线
19         ~ & \multicolumn{2}{c|}{} & ~ \\
20         \hline
21         ~ & ~ & ~ & ~ \\ \hline
22     \end{tabular}
23 \end{table}
```

该表格效果长这样：

表 7.1: 一张随意画的 L<sup>A</sup>T<sub>E</sub>X 表格

(0,0)	row 0		
column 0	center		

下面是一段三线表长表的示例:

```

1 \usepackage{longtable}
2 \caption{文本文字样式控制序列}\label{table::font_control_sequences}
3 \begin{center}
4   \begin{longtable}{C{2.5cm}C{2.5cm}C{4cm}C{2cm}C{1.5cm}}
5     \hline
6     宏 & 功能 & 参数 & 效果 & 来源 \\
7     \hline
8     \addbs{textup} & 无效果字体 & \{要显示的文字\} & \textup{text}
9     } & \LaTeX \\
10    \addbs{textit} & 意大利斜体字体 & \{要显示的文字\} & \textit{
11    text} & \LaTeX \\
12    \addbs{textsl} & Slanted斜体字体 & \{要显示的文字\} & \textsl
13    {text} & \LaTeX \\
14    \addbs{textsc} & 小写转大写后输出 & \{要显示的文字\} & \
15    textsc{text} & \LaTeX \\
16    \addbs{textrm} & 罗马字体族 & \{要显示的文字\} & \textrm{text
17    } & \LaTeX \\
18    \addbs{textsf} & 无衬字体 & \{要显示的文字\} & \textsf{text}
19    & \LaTeX \\
20    \addbs{texttt} & 打印字体 & \{要显示的文字\} & \texttt{text}
    & \LaTeX \\
    \addbs{newfontfamily} & 指定字体格式 & \{控制序列\}\{字体名\}
    & \newfontfamily{\fontcn}{Courier New} \fontcn N/A &
    fontspec \\
    \addbs{setCJKfamilyfont} & 导入中日韩 (CJK) 字体, 赋予一个别名
    & \{别名\}\{字体名\} & \setCJKfamilyfont{kf}{楷体} N/A
    & ctex \\
    \addbs{CJKfamily} & 设置中日韩 (CJK) 语言环境字体 & \{字体名称\}
    & \CJKfamily{kf} 文字 & ctex \\
    \addbs{textbf} & 文字加粗 & \{要显示的文字\} & \textbf{text
    文字} & \LaTeX \\
    \addbs{underline} & 添加下划线 & \{要显示的文字\} & \
    underline{text 文字} & \LaTeX \\
    \addbs{textcolor} & 彩色文字 & [色彩空间]\{浮点数色彩值\}\{要
  
```

```

    显示的文字\} & \textcolor[rgb]{1, 0, 0}{text 文字} &
    xcolor \\
21 \addbs{tiny} & 最小的预设字号 & N/A & \tiny text 文字 & \
    LaTeX \\
22 \addbs{footnotesize} & 脚注标准字号 & N/A & \footnotesize
    text 文字 & \LaTeX \\
23 \addbs{small} & 小号字体 & N/A & \small text 文字 & \LaTeX \\
24 \addbs{normalsize} & 默认字号 & N/A & \normalsize text 文字 &
    \LaTeX \\
25 \addbs{large} & 较大的字号 & N/A & \large test 文字 & \LaTeX
    \\
26 \addbs{Large} & 介于large与LARGE间的字号 & N/A & \Large text
    文字 & \LaTeX \\
27 \addbs{LARGE} & 介于Large与huge间的字号 & N/A & \LARGE text
    文字 & \LaTeX \\
28 \addbs{huge} & 介于LARGE与Huge间的字号 & N/A & \huge text 文
    字 & \LaTeX \\
29 \addbs{Huge} & 最大的预设字号 & N/A & \Huge text 文字 & \
    LaTeX \\
30 \addbs{zihao} & 中国人习惯的汉字字号 & \{字号\} & \zihao{3}
    三号字 & \LaTeX \\
31 \hline
32 \end{longtable}
33 \end{center}

```

宏	功能	参数	效果	来源
\textup	无效果字体	{要显示的文字}	text	L <sup>A</sup> T <sub>E</sub> X
\textit	意大利斜体 字体	{要显示的文字}	<i>text</i>	L <sup>A</sup> T <sub>E</sub> X
\textsl	Slanted 斜体 字体	{要显示的文字}	<i>text</i>	L <sup>A</sup> T <sub>E</sub> X
\textsc	小写转大写 后输出	{要显示的文字}	TEXT	L <sup>A</sup> T <sub>E</sub> X
\textrm	罗马字体族	{要显示的文字}	text	L <sup>A</sup> T <sub>E</sub> X
\textsf	无衬字体	{要显示的文字}	text	L <sup>A</sup> T <sub>E</sub> X
\texttt	打印字体	{要显示的文字}	text	L <sup>A</sup> T <sub>E</sub> X
\newfontfamily	指定字体格 式	{控制序列}{字体名}	N/A	fontspec



<code>\setCJKfamilyfont</code>	导入中日韩 (CJK) 字体, 赋予一个别名	{别名}{字体名}	N/A	ctex
<code>\CJKfamily</code>	设置中日韩 (CJK) 语言环境字体	{字体名称}	文字	ctex
<code>\textbf</code>	文字加粗	{要显示的文字}	<b>text</b> 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\underline</code>	添加下划线	{要显示的文字}	<u>text</u> 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\textcolor</code>	彩色文字	[色彩空间]{浮点数色彩值}{要显示的文字}	<b>text</b> 文字	xcolor
<code>\tiny</code>	最小的预设字号	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\footnotesize</code>	脚注标准字号	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\small</code>	小号字体	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\normalsize</code>	默认字号	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\large</code>	较大的字号	N/A	test 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\Large</code>	介于 large 与 LARGE 间的字号	N/A	text 文 字	L <sup>A</sup> T <sub>E</sub> X
<code>\LARGE</code>	介于 Large 与 huge 间的字号	N/A	text 文 字	L <sup>A</sup> T <sub>E</sub> X
<code>\huge</code>	介于 LARGE 与 Huge 间的字号	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\Huge</code>	最大的预设字号	N/A	text 文字	L <sup>A</sup> T <sub>E</sub> X
<code>\zihao</code>	中国人习惯的汉字字号	{字号}	三号字	L <sup>A</sup> T <sub>E</sub> X

下面是一段短的三线表的代码:

```

1 \begin{figure}[h]
2   \centering
3   \begin{tabular}{cc}
4     \hline

```

```

5      宏 & 内容 \\
6      \hline
7      \addbs{alph} & 小写字母 \\
8      \addbs{Alph} & 大写字母 \\
9      \addbs{arabic} & 阿拉伯字母 \\
10     \addbs{roman} & 小写罗马数字 \\
11     \addbs{Roman} & 大写罗马数字 \\
12     \hline
13     \end{tabular}
14 \end{figure}

```

宏	内容
\alph	小写字母
\Alph	大写字母
\arabic	阿拉伯字母
\roman	小写罗马数字
\Roman	大写罗马数字

## 7.2 插入图片

```

1 \usepackage{graphicx}
2 % 插入单一图片
3 \begin{figure}[H]
4     \centering
5     \caption{logo}\label{figure::logo}
6     \includegraphics[height = 8mm, width = 8mm]{figure/logo.png}
7     % 插入本地图片
8 \end{figure}
9 % 插入存在子图的图片
10 \usepackage{subcaption} % 提供了对子标题的支持
11 \begin{figure}[H]
12     \centering % 图片居中
13     \caption{两个logos}\label{figure::two_logos} % 图片总标题（还
14     设置了标签，便于跳转）
15     \subcaptionbox{GNU logo\label{figure::GNUlogo}}{\includegraphics[
16         height = 3cm, width = 3cm]{figure/GNU_logo.png}}
17     % 创建了一个子标题盒子，在里面插入了子标题和内容
18     \subcaptionbox{Freedo\label{figure::Freedo}}{\includegraphics[
19         height = 10cm, width = 10cm]{figure/Freedo.png}}

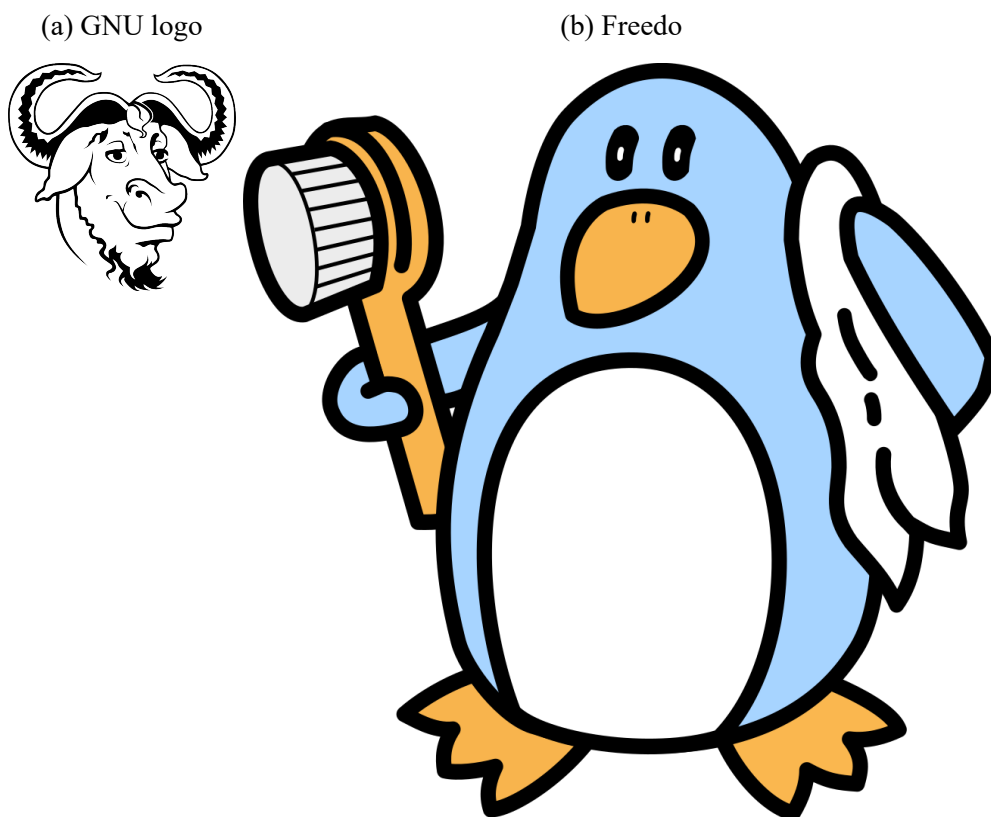
```

16 `\end{figure}`

图 7.1: 单一图片



图 7.2: 含子图图片



## 第 VIII 部分 代码块插入

代码短的插入主要依靠 listings 这个宏包，下面是一段代码示例：

```
1 \usepackage{listings}
2 % 左侧有序号的
3 \begin{lstlisting}[style = GNU_Cpp20_DevCpp] % 加载环境，参数style是自定义的格式，需要自定义这个格式
4     // A simple C++ example
5     #include<iostream>
6     using std::cout;
7     using std::endl;
8     int main() {
9         cout << "Hello world" << endl;
10        return 0;
11    }
12 \end{lstlisting}
13
14 % 左侧无序号的
15 \begin{lstlisting}[style = latex_texworks, numbers = none] % 同上，但是利用numbers参数取消了序号
16     \title{\bf \docTitle \ A Brief Chinese Tutorial on \LaTeX}
17 \end{lstlisting}
```

这里尤其要主要的是有关代码风格的问题，latex 中默认是没有代码风格的，只显示黑白文字，因此需要自己设置，下面给出一些代码高亮的设置代码，方便后续使用：

L<sup>A</sup>T<sub>E</sub>X 代码高亮

```
1 \definecolor{bgcolor_gray}{rgb}{0.9, 0.9, 0.9}
2 \definecolor{deep_green}{rgb}{0, 0.5, 0}
3 \lstdefinestyle{LaTeX_TeXworks}{
4     language=[LaTeX]TeX,
5     basicstyle=\ttfamily\small, % 设置基本字体样式
6     backgroundcolor=\color{gray!10}, % 设置背景颜色
7     morekeywords = {includegraphics, subcaptionbox, multirow
8         },
9     keywordstyle=\color{blue}, % 设置关键字颜色
10    commentstyle=\color{green!50!black}, % 设置注释颜色
11    stringstyle=\color{orange}, % 设置字符串颜色
12    numbers=left, % 行号显示在左侧
13    numberstyle=\tiny\color{gray}, % 设置行号字体样式和颜色
14    stepnumber=1, % 行号递增步长
15    numbersep=8pt, % 行号与代码之间的距离
16    tabsize=4, % 制表符大小
```

```
16     showspaces=false, % 是否显示空格
17     showstringspaces=false, % 是否显示字符串中的空格
18     captionpos=b, % 标题位置: 底部
19     breaklines=true, % 自动换行
20     frame=tb, % 框架位置: 上下
21     framerule=0pt, % 框架宽度
22     frameround=ftft, % 框架边角形状: 上下左右
23     framesep=3pt, % 框架与代码之间的距离
24     aboveskip=10pt, % 代码与上方内容的距离
25     belowskip=10pt, % 代码与下方内容的距离
26     showtabs = false,
27     extendedchars = true,
28     inputencoding = utf8,
29 }
```

## 1. C++ 代码高亮

```
1  %% Tips: C++20 关键字: alignas, alignof, and, and_eq, asm,
   atomic_cancel, atomic_commit, atomic_noexcept, auto,
   bitand, bitor, bool, break, case, catch, char, char8_t,
   char16_t, char32_t, class, compl, concept, const,
   consteval, constexpr, constexpr, constinit, const_cast, continue, co_
   _await, co_return, co_yield, decltype, default, delete,
   do, double, dynamic_cast, else, enum, explicit, export,
   extern, false, float, for, friend, goto, if, inline, int,
   long, mutable, namespace, new, noexcept, not, not_eq,
   nullptr, operator, or, or_eq, private, protected, public,
   reflexpr, register, reinterpret_cast, requires, return,
   short, signed, sizeof, static, static_assert, static_cast,
   struct, switch, synchronized, template, this, thread_
   _local, throw, true, try, typedef, typeid, typename, union,
   unsigned, using, virtual, void, volatile, wchar_t,
   while, xor, xor_eq
2  %% 宏在morekeywords里不起作用, 似乎是所有控制序列都被转义了
3  \lstdefinlanguage[ISO]{C++20}{language = [ISO]C++,
   morekeywords = {alignas, alignof, atomic_cancel, atomic_
   _commit, atomic_noexcept, char8_t, char16_t, char32_t,
   concept, consteval, constexpr, constexpr, co_await, co_
   _return, co_yield, decltype, noexcept, nullptr, reflexpr,
   requires, static_assert, synchronized, thread_local}}
4  \lstdefinlanguage[GNU]{C++20}{language = [GNU]C++,
   morekeywords = {alignas, alignof, atomic_cancel, atomic_
```

```

    commit, atomic_noexcept, char8_t, char16_t, char32_t,
    concept, consteval, constexpr, constexpr, co_await, co_
    return, co_yield, decltype, noexcept, nullptr, reflexpr,
    requires, static_assert, synchronized, thread_local}}
5 \lstdefinlanguage[ANSI]{C++20}{language = [ANSI]C++,
    morekeywords = {alignas, alignof, atomic_cancel, atomic_
    commit, atomic_noexcept, char8_t, char16_t, char32_t,
    concept, consteval, constexpr, constexpr, co_await, co_
    return, co_yield, decltype, noexcept, nullptr, reflexpr,
    requires, static_assert, synchronized, thread_local}}
6 \lstdefinlanguage[Visual]{C++20}{language = [Visual]C++,
    morekeywords = {alignas, alignof, atomic_cancel, atomic_
    commit, atomic_noexcept, char8_t, char16_t, char32_t,
    concept, consteval, constexpr, constexpr, co_await, co_
    return, co_yield, decltype, noexcept, nullptr, reflexpr,
    requires, static_assert, synchronized, thread_local}}
7 %% C++20 语法高亮
8 \lstdefinestyle{GNU_Cpp20_DevCpp}{
9     language = [GNU]C++20,
10    captionpos = b,
11    tabsize = 4,
12    breaklines = true,
13    showspaces = false,
14    showtabs = false,
15    frame = none,
16    backgroundcolor = \color{bgcolor_gray},
17    basicstyle = \CourierNew \SimSun,
18    keywordstyle = \color{black} \bf,
19    identifierstyle = \color{blue!50},
20    commentstyle = \color{deep_green},
21    numbers = left,
22    numberstyle = \small,
23    stepnumber = 1,
24    firstnumber = 1,
25    extendedchars = true,
26    inputencoding = utf8,
27 }
```

## 第 IX 部分 目录、标题和标签

### 9.1 目录设置

```
1 \usepackage{titletoc}
2 \setcounter{tocdepth}{2}
3 % \titlecontents{章节名称}[左端距离]{标题字体、与上文间距等}{标题序号}
   % {空}{引导符和页码}[与下文间距]
4 \titlecontents{part}[0em]{\SimHei \TimesNewRoman \vspace{5pt}}{\
   contentslabel{2em}}{}{\sim\titlerule*[0.6pc]{\textbullet}\sim\
   contentspage}
5 \titlecontents{section}[3em]{\SimHei \TimesNewRoman}{\contentslabel{2
   em}}{}{\sim\titlerule*[0.6pc]{.}\sim\contentspage}
6 \titlecontents{subsection}[4em]{\SimHei \TimesNewRoman}{\
   contentslabel{3em}}{}{\sim\titlerule*[0.6pc]{.}\sim\contentspage}
7 \titlecontents{subsubsection}[5em]{\SimHei \TimesNewRoman}{\
   contentslabel{2em}}{}{\sim\titlerule*[0.6pc]{.}\sim\contentspage}
```

### 9.2 标题设置

```
1 \usepackage{amsmath,algorithm2e,subcaption,titlesec,listings,nomenc}
2 % 编号跟随：使得各类标题随着章节变动而重置计数器
3 \numberwithin{section}{part}
4 \numberwithin{footnote}{part}
5 \numberwithin{table}{part}
6 \numberwithin{figure}{part}
7 \numberwithin{algocf}{part}
8 % 编号样式和内容
9 \renewcommand{\thealgocf}{\arabic{part}.\arabic{algocf}}
10 \renewcommand{\thefootnote}{\arabic{footnote}}
11 \renewcommand{\thepart}{\Roman{part}}
12 \renewcommand{\thesection}{\arabic{part}.\arabic{section}}
13 \renewcommand{\thesubsection}{\thesection.\arabic{subsection}}
14 \renewcommand{\thesubsubsection}{\thesubsection.\arabic{subsubsection}
   }}
15 \renewcommand{\thefigure}{\arabic{part}.\arabic{figure}}
16 \renewcommand{\thesubfigure}{\alph{subfigure}}
17 \renewcommand{\thetable}{\arabic{part}.\arabic{table}}
```

```
18 \renewcommand{\thesubtable}{\alph{subtable}}
19 % 显示样式和内容
20 \titleformat{\part}{\raggedright \zihao{-3} \SimHei \TimesNewRoman}{
    第 \thepart 部分}{1em}{}[]
21 \titleformat{\section}{\zihao{-4} \SimHei \TimesNewRoman}{\thesection
    }{1em}{}
22 \titlespacing{\section}{3cm}{*1}{*1}
23 \titleformat{\subsection}{\large \SimHei \TimesNewRoman}{\
    thesubsection}{1em}{}
24 \titleformat{\subsubsection}{\SimHei \TimesNewRoman}{\
    thesubsubsection}{1em}{}
25 \renewcommand{\proofname}{\bf \SimHei 证明}
26 \renewcommand{\contentsname}{\bf \SimHei 目录}
27 \renewcommand{\lstlistingname}{\SimHei 代码}
28 \renewcommand{\tablename}{\SimHei 表}
29 \renewcommand{\figurename}{\SimHei 图}
30 \renewcommand{\listalgorithmcfname}{\SimHei 算法列表}
31 \renewcommand{\algorithmcfname}{\SimHei 算法}
32 \renewcommand\nomname{术语表}
```

### 9.3 标签与跳转

```
1 标签\label{content::label}
2 \caption{图片}\label{figure::figure}
3 \caption{table}\label{tabel::tabel}
4 \ref{figure::figure} % 用于后文中的标签跳转
```



## 第 X 部分 序列的插入

```
1 \usepackage{enumitem}
2 \begin{enumerate}[label=(\arabic*)]
3     % 设置了编号的环境
4     % label 标签规定了序号的格式，该例子中格式为(1)
5     % 还有其他格式，诸如：
6     %             大写字母 (\Alph*)
7     %             小写字母 (\alph*)
8     %             罗马数字 (\Roman*)
9     %             小写罗马数字 (\roman*)
10    %             点号 (\textbullet)
11
12    \item                % 一个 item 是一个编号
13    \item
14    \item
15 \end{enumerate}
```

## 第 XI 部分 公式编辑

```
1 \usepackage{amsthm}
2 \newtheoremstyle{Chinese_mathenv}{3pt}{3pt}{\KaiTi}{\SimHei \
   TimesNewRoman}{ }{.5em}{}
3 \theoremstyle{Chinese_mathenv}
4 \newtheorem{axiom}{公理}[section]
5 \newtheorem{theorem}{定理}[section]
6 \newtheorem{corollary}{推论}[axiom]
7 \newtheorem{definition}{定义}[section]
8 \newtheorem{lemma}{引理}[theorem]
9 \newtheorem*{caution}{注意}
10 \newtheorem*{solve}{解}
11 \newtheorem*{solution}{方案}
```

其他部分的编辑与 Markdown 极度相似，因此本文便不再赘述。详见[给自己看的 Markdown 语法大全](#)

## 第 XII 部分 参考文献插入