

LaTeX学习

一、使用环境

在自己电脑上搭建可以进行中文编辑的LaTeX编译器有点恶心，容易劝退新人。因此，我们可以使用在线版本的LaTeX编译器，免安装，同时还提供一些常见的宏包，支持中文编辑以及输出位PDF。我用的LaTeX在线编辑器是overleaf，需要登录才能使用，可以使用Google或Facebook账号登录。

二、文档布局

文档类

当LaTeX 处理源文件时，首先需要知道的就是作者所要创建的文档类型。文档类型可由\documentclass 命令来指定。

`\documentclass[options]{class}`

class 指定想要的文档类型。表1.1 给出了一些文档类型的解释。LATEX2 ϵ 发行版中还提供了其他一些文档类，像信件和幻灯片等。通过options 参数可以定制文档类的属性。不同的选项之间须用逗号隔开。标准文档类的最常用选项如表1.2所示。

例：一个LaTeX源文件以下面一行开始：`\documentclass[11pt,twoside,a4paper]{article}`这条命令会引导LaTeX使用article格式、11磅大小的字体来排版该文件，并得到在a4纸上双面打印的效果。

| 文档名 | 用途 |
|---------|--|
| article | 排版科学期刊、演示文档、短报告、程序文档、邀请函..... |
| proc | 一个基于 article 的会议文集类。 |
| minimal | 非常小的文档类。只设置了页面尺寸和基本字体。主要用来查错。 |
| report | 排版多章节长报告、短篇书籍、博士论文..... |
| book | 排版书籍。 |
| slides | 排版幻灯片。该文档类使用大号 sans serif 字体。也可以选用 FoilTEXa来得到相同的效果。 |

表1.1 - 文档类

| 选项 | 作用 |
|---------------------------|---|
| 10pt, 11pt, 12pt | 设置文档中所使用的字体的大小。如果该项没有指定，默认使用10pt 字体。 |
| a4paper, letterpaper, ... | 定义纸张的尺寸。缺省设置为letterpaper。此外，还可以使用a5paper, b5paper, executivepaper 以及legalpaper。 |
| fleqn | 设置行间公式为左对齐，而不是居中对齐。 |
| leqno | 设置行间公式的编号为左对齐，而不是右对齐。 |
| titlepage, notitlepage | 指定是否在文档标题(document title) 后另起一页。article 文档类缺省设置为不开始新页，report 和book 类则相反。 |
| onecolumn, twocolumn | LATEX 以单栏(one column) 或双栏(two column) 的方式来排版文档。 |
| twoside, oneside | 指定文档为双面或单面打印格式。article 和report 类为单面(single sided) 格式，book 类缺省为双面(double sided) 格式。注意该选项只是作用于文档样式，而不会通知打印机以双面格式打印文档。 |
| | |

| | |
|--------------------|--|
| landscape | 将文档的打印输出布局设置为 landscape 模式。 |
| openright, openany | 决定新的一章仅在奇数页开始还是在下一页开始。在文档类型为 article 时该选项不起作用，因为该类中没有定义“章”(chapter)。 report 类默认在下一页开始新一章而 book 类的新一章总是在奇数页开始。 |

表 1.2 – 文档类选项

宏包

排版文档时，你可能会发现某些时候基本的LATEX 并不能解决你的问题。如果想插入图形(graphics)、彩色文本(coloured text) 或源代码到你的文档中，你就需要使用宏包来增强LATEX 的功能。可使用如下命令调用宏包 `\usepackage[options]{package}` 这里package 是宏包的名称，options 是用来激活宏包特殊功能的一组关键词。很多宏包随LATEX 基本发行版一起发布(见表1.3)，其他的则单独发布。你可以在所 安装的LATEX 系统中找到更多的宏包相关信息。The LATEX Companion [3]提供了关于宏包的重要信息，它包含了数百个宏包的描述及如何写作自己的LATEX2ε扩展的信息。

| 宏包名 | 作用 |
|----------|---|
| doc | 排版LATEX 的说明文档。 具体描述见doc.dtx及The LATEX Companion [3]。 |
| exscale | 提供了按比例伸缩的数学扩展字体。具体描述见ltxscale.dtx。 |
| fontenc | 指明使用哪种LATEX 字体编码(font encoding)。具体描述见ltoutenc.dtx。 |
| ifthen | 提供如下形式的命令if ...then do ...otherwise do具体描述见ifthen.dtx 及The LATEX Companion [3]。 |
| latexsym | 提供LATEX 符号字体。具体描述见latexsym.dtx 及The LATEX Companion [3]。 |
| makeidx | 提供排版索引的命令。 具体描述见第4.3 节及The LATEX Companion [3]。 |
| syntonly | 编译文档而不生成 dvi 文件（常用于查错）。 |
| inputenc | 指明使用哪种输入编码，如 ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows 或用户自定义编码。具体描述见inputenc.dtx。 |

表 1.3 – 随LATEX一起发行的宏包

页面样式

LATEX 支持三种预定义的页眉/页脚(header/footer) 样式，称为页面样式(pagestyle)。如下命令 `\pagestyle{style}` 中的style 参数确定了使用哪一种页面样式。表1.4 列出了预定义的页面样式。

| 样式名 | 说明 |
|----------|-------------------------------|
| plain | 在页脚正中显示页码。这是页面样式的缺省设置。 |
| headings | 在页眉中显示章节名及页码，页脚空白。 （本文即采用此样式） |
| empty | 将页眉页脚都设为空白。 |

表 1.4 – LATEX预定义的页面样式
可以通过如下命令来改变当前页面的页面样式`\thispagestyle{style}`。

各类LaTeX文件

使用LATEX 时，你可能很快发现自己置身于各种不同扩展名(extension) 或毫无线索的文件形成的迷宫之中。下面的列表解释了在使用LATEX 时可能遇到的文件类型。

| 文件后缀 | 作用 |
|------|----|
| | |

| | |
|------|--|
| .tex | LATEX 或TEX 源文件。可以使用 <code>latex</code> 命令编译。 |
| .sty | LATEX 宏包文件。可以使用 <code>\usepackage</code> 命令将宏包文件载入到你的LATEX文档中。 |
| .dtx | 文档化TEX 文件。这是LATEX 宏包文件的主要发布格式。如果编译.dtx 文档，将会得到其中包含的L |
| ATEX | 宏包文件的文档化宏代码。 |
| .ins | 对应.dtx 文件的安装文件。如果你从网上下载了一个LATEX 的宏包文件，其中一般会包含一个.dtx 文件和一个.ins 文件。使用LATEX 处理.ins 文件可以解开.dtx 文件。 |
| .cls | 定义文档外观形式的类文件，可以通过使用 <code>\documentclass</code> 命令选取。 |
| .fd | 字体描述文件，可以告诉LATEX 有关新字体的信息。 |

下面这些文件是使用LATEX 处理源文件时产生的：

| 后缀名 | 作用 |
|------|--|
| .dvi | 设备无关文件。这是运行LATEX 编译的主要结果。你可以使用 DVI 预览器预览其内容或使用dvips 或其他程序输出到打印机。 |
| .log | 记录了上次编译时的详细信息。 |
| .toc | 储存了所有的章节标题。下次编译时将读取该文件并生成目录。 |
| .lof | 和.toc 文件类似，可生成图形目录。 |
| .lot | 和.toc 文件类似，可生成表格目录。 |
| .aux | 用来向下次编译传递信息的辅助文件。主要储存交叉引用的相关信息。 |
| .idx | 如果文档中包含索引，LATEX 将使用该文件存储所有的索引词条。此文件需要使用makeindex 处理，详见位于57 页的第4.3 节。 |
| .ind | 处理过的.idx 文件。下次编译时将读入到你的文档中。 |
| .ilg | 和.log 文件类似，记录了makeindex 命令运行的详细信息。 |

大型项目

当处理大型文档时，最好将文档分割成为几部分。LATEX 有两个命令可以帮助你完成这项工作。

```
\include{filename}
```

你可以使用该命令将名为filename.tex 的文档内容插入到当前文档中。需要注意的是，在处理插入的filename.tex 文档前，LATEX 会另起一页。第二个命令只能在导言区使用。它可以让LATEX 仅读入某些\include 文件。

```
\includeonly{filename,filename,...}
```

这条命令在文档的导言区执行后，在所有的\include 命令中，只有文档名出现在\includeonly 的命令参数中的文档才会被导入。注意文档名和逗号之间不能有空格。

\include 命令会在新的一页上排版载入的文本。当使用\includeonly 命令时会很有帮助，因为即使一些载入的文本被忽略，分页处也不会发生变化。有些时候可能不希望在新的一页上排版载入的文本，这时可以使用命令

```
\input{filename}
```

\input 命令只是简单的载入指定的文本，没有其他限制。

如果想让LATEX 快速的检查文档中的错误，可以使用syntonly 宏包。它可以使LATEX 浏览整个文档，检查语法错误和使用的命令，但并不生成 DVI 输出。

在这种模式下，LATEX 运行速度很快，可以为你节省宝贵的时间。syntonly 宏包 的使用非常简单：

```
\usepackage{syntonly}
```

```
\syntaxonly
```

如果想产生分页，只要注释掉第二行即可(在前面加上一个百分号%)。

常用的支持中文的文档格式

```
\documentclass[16px,a4paper]{article}
```

```
\usepackage[UTF8]{ctex}
```

```
\author{ZS}
```

```
\title{mytitle}
\begin{document}
\maketitle
\section{第一部分}
\subsection{第一小点}
\subsection{第二小点}
\subsection{第三小点}
\section{第二部分}
\subsection{第一小点}
\subsection{第二小点}
\subsection{第三小点}
\end{document}
```

三、文本排版

| 命令 | 作用 |
|---|---|
| <code>\或\newline</code> | 断行 |
| <code>*</code> | 另起一行，而不另起一段 |
| <code>\newpage</code> | 强制断行后，还禁止分页 |
| <code>\linebreak[n]</code> ， <code>\nolinebreak[n]</code> ， <code>\pagebreak[n]</code> ， <code>\nopagebreak[n]</code> | 另起一页，通过可选参量n（0-4），作者可以影响命令的效果 |
| <code>\mbox{test}</code> | 保证把几个单词排在同一行 |
| <code>\fbox</code> | 与mbox类似，且会围绕内容画个框 |
| <code>\today</code> | 今日日期 |
| 两个重音（ <code>`</code> ）和（ <code>'</code> ） | 产生左右双引号 |
| 一个 <code>'</code> 和 <code>'</code> | 产生单引号 |
| <code>-</code> 、 <code>--</code> 、 <code>---</code> | 连字号、短破折号、长破折号 |
| <code>\~</code> 或 <code>\$\sim\$</code> | 波浪号 <code>~</code> |
| <code>\$\,\,\circ\,\,\mathrm{C}\,\$</code> | 摄氏度符号 |
| <code>\usepackage{textcomp}</code> 和 <code>\texteuro</code> | 欧元符号 |
| <code>\ldots</code> | 省略号 |
| <code>ff</code> 、 <code>fi</code> 、 <code>fl</code> 、 <code>ffi</code> | 为了避免连字现象，可以写成 <code>\mbox{f}</code> |
| <code>\@</code> | 用于点号前说明句子末尾，因为句号紧跟一个大写字母，它就不视为句子的结尾。一般在有缩写的地方，才出现句号紧跟大写字母的情况。 |
| <code>\section{}</code> ， <code>\subsection{}</code> ， <code>\subsubsection{}</code> ， <code>\paragraph{}</code> ， <code>\subparagraph{}</code> | 分节命令 |
| <code>\part{}</code> | 如果想把文档分成几个部分而且不影响章节编号 |
| <code>\chapter{}</code> | 使用report 或者book 类的时候，可以用另外一个高层次的分节命令。因为article 类的文档不划分为章，所以很容易把它作为一章插入书籍中。节之间的间隔，节的序号和标题的字号由LATEX 自动设置。 |
| | LATEX 在文档编译的最后一个循环中，提取节的标题和页码 |

| | |
|-------------------------------|--|
| <code>\tableofcontents</code> | (不会提取带*的节，如 <code>\section*{help}</code>)以生成目录。在其该命令出现的位置插入目录。 |
| <code>\maketitle</code> | 产生标题，标题的内容必须在调用 <code>\maketitle</code> 以前，由命令 <code>\title{...}</code> , <code>\author{...}</code> 和可选的 <code>\date{...}</code> 定义。在命令 <code>\author</code> 的参量中，可以输入几个用 <code>\and</code> 命令分开的名字。 |
| <code>\frontmatter</code> | 应接着命令 <code>\begin{document}</code> 使用。它把页码更换为罗马数字，而且章节不计数。当你使用带星的分节命令(例如， <code>\chapter*{Preface}</code>)时，这些章节就不会出现在目录里。 |
| <code>\mainmatter</code> | 应出现在书的第一章前面。它启用阿拉伯数字的页码计数器，并对页码重新计数。 |
| <code>\appendix</code> | 标志书中附录材料的开始。该命令后的各章序号改用字母标记。 |
| <code>\backmatter</code> | 应该插入与书中最后一部分内容的前面，如参考文献和索引。在标准文档类型中，它对页面没有什么效果。 |

引用、脚注等

1. 交叉引用
- 在书籍、报告和论文中，需要对图、表和文本的特殊段落进行交叉引用(cross-references)。LATEX 提供了如下交叉引用命令`\label{marker}`，`\ref{marker}` 和`\pageref{marker}`其中`marker` 是用户选择的标识符。如果在节、子节、图、表或定理后面输入`\label`命令，LATEX 把`\ref` 替换为相应的序号。`\pageref` 命令排印`\label`输入处的页码。和章节标题一样，使用的序号是前面编译所产生。
2. 脚注 命令`\footnote{footnote text}`把脚注内容排印于当前页的页脚位置。脚注命令总是置于(put)其指向的单词或句子的后面。脚注是一个句子或句子的一部分，所以应用逗号或句号结尾。
3. 强调 如果文本是用打字机键入的，用下划线 来强调重要的单词。`\underline{text}`但是在印刷的书中，用一种斜体字体排印要强调的单词。LATEX 提供命令`\emph{text}`来强调文本。
4. 环境 为了排版专用的文本，LATEX 定义了各种不同格式的环境(environment):
- ```
\begin{environment}
text
\end{environment}
```

其中`environment` 是环境的名称。只要保持调用顺序，环境可以嵌套。
- ```
\begin{aaa}
...
\begin{bbb}
...
\end{bbb}
...
\end{aaa}
```

| 参数 | 使用环境 |
|--------------------------|-------------|
| <code>itemize</code> | 适用于简单的列表 |
| <code>enumerate</code> | 适用于有排列序号的列表 |
| <code>description</code> | 用于带描述的列表 |

例如：

```
\begin{enumerate}
\item You can mix the list environments to your taste:
\begin{itemize}
\item But it might start to look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
```

```
\begin{description}
\item[Stupid] things will not become smart because they are in a list.
\item[Smart] things, though, can be presented beautifully in a list.
\end{description}
\end{enumerate}
```

| 代码 | 效果 |
|------------|---|
| flushleft | 左对齐段落 |
| flushright | 右对齐段落 |
| center | 文本居中 |
| quote | 引文、语录和句子 |
| quotation | 超过几段的长引用，因为它对段落进行缩进 |
| verse | 用于诗歌 |
| abstract | 摘要，一般用于article类文档 |
| verbatim | 原文打印，位于\begin{verbatim} 和\end{verbatim} 之间的文本将直接打印，包括所有的断行和空白，就像在打字机上键入一样，不执行任何LATEX 命令。在一个段落中，类似的功能可由\verb+text+完成。+ 仅是分隔符的一个例子。除了* 或空格，可以使用任意一个字符。这个小册子中的许多例子是用这个命令排印的。 |
| tabular | 排版带有水平和垂直表线的漂亮表格(table)，\begin{tabular}[pos]{table spec}，table spec 定义了表格的格式。用一个l 产生左对齐的列，用一个r 产生右对齐的列，用一个c 产生居中的列；用p{width} 产生相应宽度、包含自动断行文本的列；竖线产生垂直表线。如果一列里的文本太宽，LATEX 不会自动折行显示。使用p{width} 你可以定义如一般段落里折行效果的列。 |

参量pos 设定相对于环绕文本基线的垂直位置。使用字母 t、b 和 c 来设定表格靠上、靠下或者居中放置。在tabular 环境中，用& 跳入下一列，用\\开始新的一行，用\hline 插入水平表线。用\cline{j-i} 可添加部分表线，其中 j 和 i 分别表示表线的起始列和终止列的序号。|

表格的列分隔符可由@{...} 构造。这个命令去掉表列之间的间隔，代之为两个花括号间的内容。一个用途在于下面要解释的十进制数对齐问题。另一个可能应用在于用@{} 压缩表列右端空间。由于没有内建机制使十进制数按小数点对齐18，我们可以使用两列“作弊”达到这个目的：整数右，小数向左对齐。 \begin{tabular} 行中的命令@{.} 用一个“.” 取代了列间正常间隔，从而给出了按小数点列对齐的效果。不要忘记用列分隔符(&) 取代十进制小数点！使用命令\multicolumn 可在数值“列”上放置一个列标签。

对于在当前排不下的任何一个图片或表格，其解决办法是把它们“浮动”到下一页，与此同时当前页面用正文文本填充。LATEX 提供了两个浮动体(floatingbodies) 环境：一个用于图片，一个用于表格。要充分发挥这两个环境的优越性，应该大致了解LATEX 处理浮动体的内在原理。但是浮动可能成为令人沮丧的主要原因，因为LATEX 总不把浮动体放在你想要的位置。包含在figure 环境或table 环境中的任何材料都将被视为浮动内容。两个浮动环境都支持可选参数 \begin{figure}[placement specifier] 或\begin{table}[...] 称为placement specifier，它由浮动许可放置参数写成的字符串组成。这个参数用于告诉LATEX 浮动体可以被移放的位置。一个placement specifier 由一串浮动体许可放置位置(float-placing permissions) 构成。

| Spec | 浮动体许可放置位置 |
|------|----------------------------|
| h | here 在文本的确切位置上，对于小的浮动体很有用。 |
| t | 在页面的顶部(top) |
| b | 在页面的底部(bottom) |
| p | 在一个只有浮动体的专门的页面(page) 上。 |
| ! | 忽略阻止浮动体放置的大多数内部参数 |

数学公式

位置

处于段内的数学文本要放在`\与\`，`$与$`，或`\begin{math}`与`\end{math}`之间。

显示防置，占一行或更多，用`[与]`，或`\begin{displaymath}`与`\end{displaymath}`之间。

如果希望给你的方程编上号，你可以使用`quation` 环境。然后你就可以用`label` 来给一个方程加上标签并在文中的某处用`\ref` 或`amsmath` 宏包中的`\eqref` 命令来引用它。

```
\begin{equation} \label{eq:eps}
```

```
\epsilon > 0
```

```
\end{equation}
```

```
From (\ref{eq:eps}), we gather \ldots{}From \eqref{eq:eps} we do the same.
```

数学模式和文本模式都有一些不同之处。例如，在数学模式中：

1. 大多数的空格和断行没有任何意义，而且所有的空隙要么是从相应数学表达式中自然的生成，要么是用一些专门的命令来指定，如`\,`，`\quad`或`\qquad`。
2. 空白行是不允许的。每个公式只能为一段。
3. 每一个字母都会被认为是一个变量名，且会相应被排版为此种样式。如果你想要在公式中排版普通的文本（直立字体和普通字距），那么你必须要把这些文本放在`\textrm{...}` 命令中。
4. 大部分数学模式的命令只对其后的一个字符有效，因此，如果你希望一个命令对多个字符起作用，你必须把它们放在一个群组中，使用花括号：`{...}`。

基本元素

1. 指数`\^{}{}` 与下标`_{}{}`
2. 平方根`\sqrt[n]{}{}`或仅仅平方根`\surd`
3. 水平线（上`\overline{}`、下`\underline{}`）
4. 命令`\overbrace{}` 和`\underbrace_{}{}` 可以在一个表达式的上方或下方生成水平括号
5. 为了给变量增加数学重音符号，如小箭头或是`\tilde{}`，覆盖多个字符的宽“帽子”和宽`\sim`号，可以由`\widehat`和`\widetilde` 得到。`\prime` 符号则给出了一个撇号(prime)。
6. `\overrightarrow{AB}` 和`\overleftarrow{}`:向量或单个`\vec`
7. `\log` 等类似的函数名通常是用直立字体，而不是如同变量一样用斜体，因此`LATEX` 提供了以下的命令来排版这些最重要的函数名：
`\arccos\cos\csc\exp\ker\limsup\arcsin\cosh\deg\gcd\lg\ln\arctan\cot\det\hom\lim\log\arg\coth\dim\inf\liminf\max\sinh\sup\tan\tanh\min\Pr\sec\sin`
8. 对于取模函数(modulo function)，有两个命令：`\bmod` 用于二元运算“`a mod b`”，而`\pmod` 则用于表达式如“`x ≡ a (mod b)`”。
9. 分式(fraction) 可用`\frac{...}{...}`
10. `amsmath` 宏包中的`\binom` 命令可以用来排版二项式。
11. 积分号 (integral operator) 可以用`\int` 产生，求和号 (sum operator) 用`\sum`命令，而乘积号 (product operator) 要用`\prod` 命令。上限和下限用`^` 和`_` 来指定，如同上标与下标一样。