

# L<sup>A</sup>T<sub>E</sub>X 使用简要说明

hanhj

2020 年 2 月 14 日

## 目录

## 1 安装 latex 软件

为了使用 latex 需要安装如下软件:

texlive : 基础

texlive-xetex : 编译器

tex-cjk-chinese, texlive-lang-chinese : 中文支持

texmaker:latex : latex编辑器, 可选

帮助:

texdoc 包名, 可以查看本地包的帮助信息。

本地的包放在两处, 一处是发行的包放在 `/usr/share/texlive/texmf-dist/tex/latex/`。

一处是插件, 放在 `/usr/share/texmf/tex/latex/`下面。

## 2 第一个 latex 文档

```
\documentclass{article}
\begin{document}
Hello world
\end{document}
```

将以上文档保存为 test.tex

编译: `pdflatex test.tex` 默认输出 pdf

在 linux 下查看 pdf 文档有 evince, okular, xpdf 等

## 3 中文支持

- 使用 CJK 来支持中文: CJK 是 chinese, Japan, Korea 中日韩三国。  
CJK 是德国人写的一种中文挂件。

```
\documentclass{article}
\usepackage{CJK} %使用CJK
\begin{document}
\begin{CJK}{UTF8}{gbsn} %指定编码UTF8,字体gbsn
你好,世界
```

```
\end{CJK}  
\end{document}
```

编译: `pdflatex test.tex`

关于编译: latex 文档的编译工具有 latex, xelatex, pdflatex.

传统的方法: 先用 latex 生成 dvi 文件, 然后用 dvips 转成 ps 文件, 再用 ps2pdf 转成 pdf 文件。

```
latex xx.tex
```

```
dvips xx.dvi
```

```
ps2pdf xx.ps
```

第二步也可省略, 用 dvipdf 直接将 dvi 文件转换成 pdf, `dvipdf xx.dvi`。

更直接的方法是用 pdflatex, 直接将 tex 文件转换成 pdf。 `pdflatex xx.tex`

另外一种直接的方法是用 xelatex, 也可直接将 tex 文件转换成 pdf。但是这种方法需要在文档中使用 xetex 或 ctex 来支持中文 (参见下文)。

```
xelatex xx.tex
```

- 使用 xeTeX 来支持中文

```
\documentclass{article}  
\usepackage{xeCJK}  
\setCJKmainfont{AR PL UMing CN} %设置中文字体  
\begin{document}  
你好, 世界  
\end{document}
```

编译: `xelatex test.tex`

- 使用 ctex 来支持中文。ctex 是中国人写的从底层支持中文的一种方法。

```
\documentclass{ctexart}  
\usepackage{ctex}  
\begin{document}  
...  
\end{document}
```

编译: xelatex test.tex。

- 中文字体

- 当使用 CJK 时

可选用的字体在: /usr/share/texmf/tex/latex/CJK/UTF8/ 目录下, 有 bkai 楷体, gbsn 宋体, gkai 楷体, bsmi, goth, min。

`\begin{CJK}{UTF8}{gbsn}` %指定编码UTF8, 字体gbsn。

在这里可以选择不同字体, 如 gkai, bkai 等。

- 当使用 ctex 时

ctex 内置支持几种字体宋体, 黑体, 仿宋, 楷书, 在 /usr/share/texlive/texmf-dist/tex/latex/ctex/fontset/ctex-fontset-fandol.def 中定义了一些预定义:

`\songti \heiti \fangsong \kaishu`

可以在 document 之前使用这些预定义。例如:

```
\documentclass{ctexart}
\usepackage{ctex}
\heiti
\begin{document}
...
\end{document}
```

注意: 需要用 xelatex 来编译。

- 另外可以使用 ubuntu 中的系统字体。

首先: 用 `fc-list :lang=zh` 查找当前系统中的中文字体。然后使用

`setCJKmainfont{字体}` 来设置。

比如 `fc-list` 命令输出有下面一行: /usr/share/fonts/truetype/arphic/uming.ttc: AR PL UMinG TW MBE:style=Light 。然后在 tex 文件中加入:

```
\documentclass{article}
\usepackage{xeCJK}
setCJKmainfont{AR PL uming TW MBE} %就可以使用这种字体了。
\begin{document}
```

```
...  
\end{document}
```

注意：需要用`xelatex`编译。

## 4 文档结构

你已经按照上面的说明，建立第一个 latex 文档，并支持中文了。  
现在解释一下第一个文档：

```
\documentclass{article}  
\begin{document}  
Hello world  
\end{document}
```

Latex 文档由控制命令和正文组成。其中控制命令由 `\` 后面跟一个命令组成。比如：

```
\documentclass{article}
```

命令只由字母组成，`latex` 以命令之后的空格，数字或非字母的字符作为该命令的结束。命令可以带参数，用 `{...}` 括起来，也可以带选项 `[...]`，比如：

```
\documentclass[UTF8]{article} %表示文档采用utf8编码
```

### 4.1 文档结构

引言部分	{	<code>\documentclass{article}</code>	这个是定义文档格式
		<code>\usepackage{graphicx}</code>	这个是引用宏包
		...	其他一些宏包
		...	
正文部分	{	<code>\begin{document}</code>	表示正文开始
		<code>\title{...}</code>	标题
		<code>\maketitle</code>	生成标题
		...	
		<code>\end{document}</code>	表示正文结束

文档类 `documentclass` 的参数包括：`article`(论文，期刊)，`proc` (会议文集)，`report`，`book` 等。

其配置项包括：fleqn(行间公式左对齐，默认中间对齐)，10pt，11pt（文中字体大小，默认 10pt）等等。

宏包是为了增强 latex 的功能所发布的命令包。比如通过 graphicx 包可以插入图片。

命令是对全文的一些处理：比如 setCJKmainfont 设置字体，maketitle 设置文档首页内容（包括标题，作者，日期）。

## 4.2 保留字符

在 latex 中有一些字符是作为保留用的特殊字符，通常在正文中不会被打印出来：

#	自定义命令中的参数
&	代表表格中的连接符
%	代表注释
\$	代表数学符号
{ }	代表参数
^	代表上标
_	代表下标
\	代表命令

表 1: 保留字符

如果在正文中想打印这些字符，在前面加上 \，特别的对于 \ 要用 \backslash 来表示，^ 要用 \wedge 来表示。

## 5 基本排版

### 5.1 标题

正文的标题可以用下面的文字生成：

```
\begin{document}
\title{文章的标题}
\author{作者}
\today
```

```
\maketitle
...
\end{document}
```

这样就可以生成标题了。...是正文其他部分。

## 5.2 节

节是内容的比较大的集中部分

```
\section{节的标题}
```

节会产生编号 (如果在 section 后跟 \* 则不会产生编号)。在节中还可以分成字节, 子子节, 分别用 \subsection, 和 \subsubsection 来产生。

## 5.3 段

段是一段内容的集合, 是比节小一点的内容的集合。

```
\par
```

在两段之间空两行, 也可以达到分段的目的。除非强制分行, 否则在一段内只有一行, 不会另起一行。

如果想强制分行, 可以在行尾加上 \\. 或者 \newline.

段的开始, 根据设定可以产生缩进。可以手动设置:

```
\setlength{\parindent}{20em}
```

或者用引言部分用包 \usepackage{indentfirst}。

## 5.4 其他一些分段命令

paragraph: 用来产生一段, 与 section 类似, 但是不会产生编号。

part: 用来分部。比 section 更大的分段, 它不会影响部与部之间的节编号。

newpage: 另起一页。

linebreak[n], nolinebreak[n]: 新增 n 行

pagebreak[n], nopagebreak[n]: 新增 n 页



## 5.5 强调

为了突出某些内容，可能需要对文字作出一些强调。强调包括下划线，斜体，黑体。

`\underline{...}` 下划线

`\emph{...}` 斜体

`\textit{...}` 这也是斜体

`\textbf{...}` 黑体

下划线, 斜体, *hello*, 黑体

## 5.6 交叉引用

在文档内可能需要对某些内容做标记，当别处引用此内容时，可以快速能够跳到该处。

`\label{mark}` 在需要做标记的地方

`\ref{mark}` 在引用的地方，显示标记号

`\pageref{mark}` 在引用的地方，显示标记的页号

## 5.7 脚注

`\footnote{...}` 在需要做脚注的地方

<sup>1</sup>脚注

## 5.8 摘要

论文中常在开头做摘要

`\abstract{...}`

## 5.9 环境

前面描述的命令都是单条命令，在 latex 中还有一种叫做环境的命令，它用 `begin` 开头，`end` 结尾。形如：

---

<sup>1</sup>this is a footnote

```
\begin{command}
...
\end{command}
```

意思是这条命令将影响到所包括的内容。document 就是一条环境命令。除此以外还有如下的一些常用的环境命令：

<i>flushleft, flushright</i>	左对齐，右对齐
<i>itemize</i>	无序列表
<i>enumerate</i>	有序列表
<i>tabular</i>	表格
<i>array</i>	数学公式中的多行公式
<i>verbatim</i>	代码

表 2: 常用环境命令

itemize 用法：

```
\begin{itemize}
\item xxx
\item xxx
...
\end{itemize}
```

enumerate 用法：

```
\begin{enumerate}
\item xxx
...
\end{enumerate}
```

verbatim 用法：

```
\begin{verbatim}
随便写，就像code一样
\end{verbatim }
```

tabular 用法：

```

\begin{tabular}[文字位置]{列对齐}
row1_column1 & row1_column2 ...
...
\end{tabular}

```

下面是一段示例：

A	B
a1	b1
a2	b2

```

\begin{tabular}[t]{|l|l|}
\hline
A & B \\
\hline
a1 & b1 \\
\hline
a2 & b2 \\
\hline
\end{tabular}

```

文字位置用 t 表示对上对齐，b 表示对下对齐，c 表示居中。

列对齐用 l 表示左对齐，r 表示右对齐，c 表示居中。中间用 | 表示画竖线。

hline 表示画横线，cline{n,n} 表示在第 n 列中画竖线。

array 用法：

下面是一段示例：

```

\left[
\begin{array}{l}
I_x < dz \\
U_x < dz
\end{array}
\right.

```

## 5.10 数学公式

数学公式在 latex 中有两种打印方式：行内或行间。行内就是在一行之内，用  $\$...\$$  来表示。行间就是位于两行之间，用

```
\[  
...  
\]
```

或者用

```
\begin{math}  
...  
\end{math}
```

或者用

```
\begin{equation}  
...  
\end{equation}
```

比如：  $a^2 + b^2 = c$  这是行内公式。

$$x^2 + y^2 = z$$

这是行间公式。

`math` 与 `equation` 的区别在于 `equation` 会自动给公式加上编号（默认在右侧），而 `math` 不会。

数学公式的写法基本与自然写法一样，只不过要记住公式中各个元素的表达方式。比如  $\alpha$  用 `\alpha` 来表示， $\sum$  用 `\sum` 来表示。具体元素的表达方法参见 [?], 多用几次就熟悉了。这里提几个修饰符：

数学公式默认的居中排布。如果想左对齐或右对齐，可以在开始的地方设置

```
\documentclass[fleqn]{article}
```

还可以设置缩进宽度。

```
\setlength{\mathindent}{0pt}
```

$\wedge$	上标, 可以用来表示次方, 积分或求和公式的上限。
$—$	下标, 可以用来表示公式中的下标, 积分或求和公式中的下限。
$\overline{m+n}$	上划线, <code>\overline</code>
$\underline{m+n}$	下划线, <code>\underline</code>
$\overbrace{a+b+c}$	上括号, <code>\overbrace</code>
$\underbrace{a+b+c}$	下括号, <code>\underbrace</code>
$\frac{a}{b}$	分号, <code>\frac{a}{b}</code>
空格	<code>\quad</code> or <code>\qquad</code>
公式中的字符串	<code>\mathrm{...}</code>

表 3: 数学公式

## 5.11 表和图

文档中常有表和图, 需要给它们编号, `table` 和 `figure` 就是为表和图加上编号的命令。 `table`, `figure` 在 latex 中称为浮动体。

```
\begin{figure}[option]
```

插入图片

```
\end{figure}
```

```
\begin{table}[option]
```

插入表格

```
\end{table}
```

使用默认的 `figure` 往往不能将图片放到期望的位置。这时可以用选项 `[!htp]`。

h	here 明确的位置
t	top 页面顶部
b	bottom 页面底部
p	page 在一个只有浮动体的页面
!	取消大多数浮动体的默认设置

表 4: 浮动体位置

为了插入图片, 需要使用 `graphicx` 包。

```

\usepackage{graphicx}
\begin{document}
...
\includegraphics[option]{filename} %这里插入图片

\end{document}

```

filename 可以不带后缀名, 让 latex 自己在当前目录中找图片。如果用 latex 编译, 支持的文件格式为 eps。如果用 pdflatex 编译, 支持的文件格式包括 png, pdf, jpg, mps。不幸的是 pdflatex 并不支持 eps, 所以如果用 pdflatex 来编译, 需要将 eps 文件转换成其所支持的文件。

includegraphics 的选项包括:

scale	缩放
width	宽度
height	高度
angle	角度

表 5: graphicx 的选项

高度和宽度的单位可以是:mm,cm,in(1 英寸 =25.4mm),pt(point  $1point \approx \frac{1}{3}mm$ ),em(当前字母 M 的宽度), ex(当前字母 x 的高度)。

表和图的索引可以用:

```

\listoftables
\listoffigures

```

来生成, 一般放在文档最后。

## 5.12 索引

在文档中对一些关键词进行索引, 将该索引表放在文档的最后, 可以方便的在文档中查找关键词是非常有用的。

为了生成索引, 需要在引言部分引入 makeidx 包, 并且调用 makeindex 命令。在正文中需要产生索引的地方调用 index 命令, 在需要打印索引表的地方调用 printindex 命令。

```

\usepackage{makeindex} %引入makeidx包
...
\makeindex %调用makeindex命令, 生成index
\begin{document}
...
\index{key} %产生索引
...

\printindex %打印索引表
\end{document}

```

在编译的时候,需要两次编译。即 latex xx.tex,makeindex xx.idx,latex xx.tex。第一次编译生成索引文件 idx, 用 makeindex 程序将 idx 文件转换成 ind 文件, 第二次编译的时候就会将 ind 文件包括进来, 在 printindex 处打印。

### 5.13 引用

论文中经常有引用的参考文献, 用如下方法实现:  
定义参考文献:

```

\begin{thebibliography}
\bibitem{标号} 文献描述
...

```

然后在引用的地方:

```

\cite{标号}

```

### 5.14 在 pdf 文档中加入左侧书签

需要引入 hyperref 宏包。由于 hyperref 使用了扩展 CJK, 所以需要 xelatex 来支持中文。

```

\usepackage{xeCJK}
\setCJKmainfont{AR PL UMing CN} %宋体, 可以选用其他字体
\usepackage{hyperref}
...

```

## 6 扩展 latex

以上是 latex 的基本用法, 为了提高效率, 增加功能有必要对基本 latex 进行扩展。

### 6.1 用户自定义命令

为了提高效率, 可以将常用的命令定义成用户自定义命令, 从而可以减少敲键盘的次数。用户自定义命令用 `newcommand`, `renewcommand` 来定义。前者适合原系统没有定义的命令, 后者适合系统已经定义的命令, 用来重新定义。(类似于 vim 中的 map) 命令。命令定义需要放在引言处。

```
\newcommand{\command}[n]{define}
\renewcommand{\command}[n]{define}
```

`command` 是新定义的命令名, `define` 是需要执行的动作, `n` 是参数个数, 参数个数从 1 到 9, 如果没有定义, 默认是 0。参数从命令后开始 1,2...。在定义中引用参数使用 `#1`, `#2`...。下面是一个例子: 假设将 `\la` 定义成 `LATEX`

```
\newcommand{\la}[1]{\LaTeX{#1}}
```

这是使用 `\laAAA` 的效果 `LATEXAAA`。

### 6.2 用户自定义环境命令

```
\newenvironment{command}[n]{action on begin}{action after end}
```

下面的例子假设在输入字符串的两边加上 `[]`

```
\newenvironment{tl}
{
\textbf{[]}%定义begin动作
}
{
\textbf{[]}%定义end动作
}
```

这里使用了 `tl` 环境命令

```
\begin{tl>Hello\end{tl}
```

的效果: `[Hello]`



### 6.3 定义用户自己的包

上述命令放在导言区，每次需要包括比较麻烦，可以类似 c 的文件，将上述内容放在一个 sty 文件中，然后使用 usepackage 命令包括进来就比较方便了。

sty 文件内容与上述内容类似，只不过需要在第一句加上 ProvidesPackage{包名}。下面是一个 sty 文件的例子：

```
\ProvidesPackage{mylatex}
\newsavebox{\newname}
\newcommand{\la}[1]{\LaTeX\textbf{#1}}
\newenvironment{tl}
{
\textbf{[]}%
}
{
\textbf{[]}%
}
```

该文件放在编译文件的同一个目录，或者放在/.texmf/tex/latex 目录下。  
同时设置环境变量 TEXMFHOME: export TEXMFHOME=/.texmf/tex/latex。  
上面的目录是用户私有的，也可以放在系统的包目录:/usr/share/texmf/tex/latex 中。

### 6.4 其他包

在网络上 [www.ctan.org](http://www.ctan.org) 网站上可以搜索到其他有用的包。

从上面下载包到本地，一般是压缩文件。解压文件到 /usr/share/texmf/tex/latex/目录下。包里面有 ins 文件，是安装文件，用 latex 来编译，可以生成必要的 sty 等文件。如果没有 ins 文件，也会有 dtx 文件是发行文件，用 latex 编译该文件会生成 sty 文件，这个就是我们需要的包文件。

执行 mktexlsr, 或 texhash 命令更新 latex 文件名数据库，就可以使用了。

#### 生成帮助：

用 latex 来编译 dtx 文件，然后会生成 dvi 文件，再用 dvips 转成 pdf 文件，必要的话可能还需要 makeindex 来生成索引文件。以上是生成帮助的手动方法，有些包将以上过程封装在一个脚本中，也是一样的。

## 参考文献

- [1] latex 官方文档

表格