

# My Document

wxdespair

2020 年 12 月 9 日

在命令行可以使用命令 `latex filename.tex` 对 tex 文件进行编译，并生成 `filename.aux`、`filename.dvi`、`filename.log` 三种文件；然后执行命令 `pdflatex filename.dvi`，就可以生成 pdf 文件。使用命令 `dvipdfmx filename.dvi` 也可以从 dvi 文件生成 pdf。（从 dvi 生成 pdf 的命令，前者总报错，一般使用后者，即 `dvipdfmx`）又或者直接使用软件进行快捷编译加预览。

在命令行使用 `xelatex filename.tex` 可以直接编译并生成 pdf 文件。（也会有中间文件产生）

LaTeX 文件的结构，详情查看 tex 源文件。

可以使用一对大括号来限定设置的效果范围。

行内公式： $a^2 + b^2 = c^2$ 。

行间公式：

$$a^2 + b^2 = c^2$$

带编号的行间公式：

$$AB^2 = BC^2 + AC^2. \tag{1}$$

目录 `./texlive/2020/texmf-dist/doc/latex` 下存放着 LaTeX 的宏包，每个宏包一般都有一个使用和信息介绍。也可以在命令行使用命令 `texdoc ctex` (`ctex` 为包名) 来调出一些使用手册。

在  $\text{\LaTeX}$  中，每种字体都有 5 种字体属性：字体编码、字体族、字体系列、字体形状、字体大小。字体编码包括：正文字体编码 (OT1、T1、EU1 等)、数学字体编码 (OML、OMS、OMX 等)。字体族包括：罗马字体、无衬线字体、打字机字体。字体系列包括：粗细、宽度。字体形状包括：直立、斜体、伪斜体、小型大写。

Roman Family Sana Serif Family Typewriter Family

Roman Family Sans Serif Family Typewriter Family

Medium Series **Boldface Series**

Medium Series **Boldface Series**

Upright Shape *Italic Shape Slanted Shape* SMALL CAPS SHAPE

Upright Shape *Italic Shape Slanted Shape* SMALL CAPS SHAPE

宋体 黑体 仿宋 楷书

对于中文的粗体与斜体，分别是使用黑体和楷体来分别的，使用上面的设置也是会相应是显示黑体或楷书。

tiny  
scriptsize

footnotesize

small

normalsize

large

Large

LARGE

huge

Huge

小初号 五号 二号 小二号

**利用 *newcommand* 创建的新命令显示的文字**

## 一、 章节

### 一、子章节

### 二、子章节

在编写内容时，使用双反斜杠只换行不换段，即下行没有首行缩进。使用 `par` 命令可以产生新的段落。或者直接空一行多方便。

#### 1.2.1 子子章节

## 二、特殊字符

### 一、空白字符

a b

a b

a b a b

a b

a b

a b

a b

ba

a b

a b

a b

a

b

## 二、 $\text{\LaTeX}$ 控制符

# \$ \% \{ \} \sim \\_ \^ \&amp;

## 三、排版符号

§¶†‡©£

四、T<sub>E</sub>X 标志符号

T<sub>E</sub>X L<sup>A</sup>T<sub>E</sub>X L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

## 五、引号

“ ”

## 六、连字符

—

## 七、非英文字符

œEæÆåÄøØℓß ¡ ¿

## 八、重音符号 (以 o 为例)

ò ó ô õ ö ø ò ö ö ö ö ö ö ö ö ö

## 三、插入图像

includegraphics 命令的 <options> 一般有: angle(旋转角度)、scale(缩放因子)、height(固定高度)、width(固定宽度)。在设置高度宽度时可以使用一些特殊的命名来获取当前版式的一些数据, 例如 \textheight 表示当前版式的高度值, 所以可以设置 height=0.1\textheight 来控制图像的高度为整个版式高度的 0.1。(可以同时设置多个参数, 用逗号隔开)

例如:



四、插入表格

使用 tabular 环境来绘制表格。

第一列	第二列	第三列	空
右对齐文本	居中对齐文本	左对齐文本	固定列宽为 0.45 版面宽度

还有一些其他的宏包可以用于绘制更为复杂的表格，比如：booktab、longtab、tabu 等。

五、浮动体

可以将图像或表格数据放入浮动体环境中来显示，不接合正文，使用会更加方便。  
这里有一张某某某的图像 ??，还有一个某某某的表格 ??。



图 1: 浮动体 (图像) 的标题，且自动排序

第一列	第二列	第三列	第四列
右对齐文本	居中对齐文本	居中对齐文本	左对齐文本

表 1: 浮动体 (表格) 的标题，且自动排序

关于标题控制，可以使用 caption、bication 等宏包来更复杂的设置。  
并排与子图表设置可以使用 subcaption、subfig、floatrow 等宏包。  
绕排设置可以使用 picinpar、wrapfig 等宏包。

六、数学公式

可以使用常用的  $\$ \$$  行内公式，也可以使用  $\backslash ( \backslash )$ ，又或者直接使用 math 环境均可。 $\$$  最常用。  
行间公式有  $\$ \$ \$ \$$ ，也可以用  $\backslash [ \backslash ]$ ，或者使用 displaymath 环境均可。  
equation 环境可以对公式进行自动排号，而 equation\* 环境不会对公式进行自动排号。公式也可以使用  $\backslash label\{eq:...\}$  命令来对公式进行交叉引用 (在引用的地方使用 ref)。  
添加 amsmath 宏包来编辑矩阵。矩阵有多种使用方法，一种是 gather\* 环境下使用不同的矩阵环境 (不使用这个环境，在矩阵整体前后添加  $\backslash [ \backslash ]$  也可以)：

$$\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} 0 & 1 \\ 1 & 0 \end{Bmatrix} \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} \left\| \begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \right\|$$

或者:

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}$$

还有一种是直接使用无边界矩阵，然后在两端添加整体的括号，但这种方式不会单独生成一部分，与正文一起显示 (最好在两端添加 \$ 符号):

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array}$$

可以使用 `\hdotsfor< 列数 >` 来编辑跨列的省略号:

$$\begin{pmatrix} 1 & 2 & \dots & n \\ \hdotsfor{4} \\ m & m+1 & \dots & m+n-1 \end{pmatrix}$$

使用 `smallmatrix` 环境可以编辑行内小矩阵:  $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

还有 `array` 环境也可以进行矩阵编辑 (类似于表格环境 `tabular`):

$$\begin{array}{c|c} \frac{1}{2} & 0 \\ \hline 0 & -\frac{a}{b}c \end{array}$$

同时 `array` 环境也可用于编辑复杂矩阵:

$$\underbrace{\begin{pmatrix} a & \cdots & a \\ & \ddots & \vdots \\ & & a \end{pmatrix}}_m \underbrace{\begin{pmatrix} b & \cdots & b \\ & \ddots & \vdots \\ c & \cdots & c \end{pmatrix}}_m \left. \begin{array}{l} \left. \begin{array}{l} \vdots \\ \vdots \\ \vdots \end{array} \right\} p \\ \left. \begin{array}{l} c & \cdots & c \\ \vdots & & \vdots \\ c & \cdots & c \end{array} \right\} q \end{array} \right\}$$

多行公式，使用 `gather` 环境可以编辑多行公式，并自动编号，也可以在 `\\` 前添加 `\notag` 来阻止编号。(gather\* 环境则不进行编号)

$$a + b = b + a \tag{2}$$

$$ab = ba$$

$$a \times (b + c) = ab \times ac \tag{3}$$

可以使用 `align` 和 `align*` 环境对公式进行对齐排版，前者编号，后者不编号。

使用 `split` 环境可以对一个跨行公式进行编号，编号在中间。

$$\begin{aligned} \cos 2x &= \cos^2 x - \sin^2 x \\ &= 2 \cos^2 x - 1 \end{aligned} \tag{4}$$

`cases` 环境可以实现编辑分段函数。(在数学公式中处理中文需要使用 `\text{}` 对中文进行包裹, 否则无法正常显示)

$$f(n) = \begin{cases} n/2, & \text{if } n \text{ is even} \\ 3n + 1, & \text{if } n \text{ is odd} \end{cases} \quad (5)$$

## 七、自定义命令和环境

使用 `\newcommand` 在导言区来定义新的命令。命令只能由字母组成, 不能以 `\end` 开头。

People's Republic of *China*

带参数的命令, 参数个数可以从 1 到 9, 使用时用 `#1, #2, ..., #9` 来代替输入的参数。例如输入命令: 参数 1 cp 参数 2。命令的第二个中括号可以指定参数的默认值, 默认指定首个参数的参数值。

`\renewcommand` 命令可以用来重写已有的命令。

可以由 `\newenvironment` 命令新建环境, `\renewenvironment` 重写环境