

L^AT_EX 简明速查手册

1. L^AT_EX 中的\usepackage{amsmath}类似于 C 语言的#include<stdio.h> 和 Python 的import numpy, 常用usepackage如下:

```
amsmath, amssymb, bm, ctex, datetime,
diagbox, enumerate, esint, extarrows,
fancyhdr, fontspec, geometry, graphicx,
listings, longtable, makecell, multicol,
tabularx, tcolorbox, tikz, xcolor
```

其中ctex包用于提供中文显示。

2. 上标 [~] 下标 _~

A_b^c : A_b^c A_{bc}^{def} : A_{bc}^{def}
如果上下标的字母不止一个, 则需要加大括号。
组合数 $\binom{n}{2}$: \binom{n}{2}或{n\choose 2}

3. 分数与根号:

高度自适应的分数 (行内较矮, 行间较高): \frac{}{}
强制较高的分数 $\frac{\pi^2}{6}$: \dfrac{\pi^2}{6}
强制较矮的分数 $\frac{\pi^2}{6}$: \tfrac{\pi^2}{6}
(在 TexStudio 中, 用 Alt+Shift+F 输入\frac{分子}{分母}, 编辑好“分子”后, 按“Ctrl+ 右方向键”可以快速选中“分母”, 比按两次右方向键再按 Delete 键更方便。
)
 $\sqrt{5}$ \sqrt{5} $\sqrt[3]{5}$ \sqrt[3]{5}

4. 运算符

+	+	≅	\cong
⊙	\odot, \bigodot	±	\pm
⊕	\oplus, \bigoplus	∓	\mp
×	\times	∈	\in
⊗	\otimes, \bigotimes	∩	\cap, \bigcap
÷	\div	∪	\cup, \bigcup
≠	\neq	∧	\wedge, \bigwedge
≤	\leq	∨	\vee, \bigvee
≤	\leqslant 需{amssymb}	⊂	\subset
≥	\geq	⊃	\supset
≥	\geqslant 需{amssymb}	⊆	\subseteq
≫	\gg	⊇	\supseteq
≪	\ll	∫	\int
≡	\equiv	∬	\iint
~	\sim	∭	\iiint
↪	\backsim	∭	\iiiint
≈	\approx	∫	\oint, \bigoint
∬	\varoiint 需{esint}		
∫	\ointctrlockwise 需{esint}		
∫	\varointclockwise 需{esint}		

(“需{esint}”是指需要\usepackage{esint})
修改不等号的样式:

```
\renewcommand\leq\leqslant
\renewcommand\geq\geqslant
```

修改不等号样式以后, 临时想使用原始样式的 \neq , 则需采用以下曲折方法,
在\renewcommand\leq\leqslant之前加上以下两行

```
\let\leqstore\leq
\let\geqstore\geq
```

即用两个新的命令\leqstore, \geqstore保存 \neq 的旧样式, 想用旧样式时, 使用\leqstore, \geqstore即可。

5. 特殊符号 (转义) 显示:

\$	\\$	&	\&
#	\#	^	\^{}或\sim
%	\%	_	_{}或\sim
{	\{		
}	\}		
\	\backslash\$ 或 \textbackslash		
@	符号无需转义, 可直接显示。还有一个统一的方法, 就是将特殊符号置于\verb 的两条竖线之内。		

6. 其它符号与形状:

∃	\exists	∂	\partial
∀	\forall	∅	\varnothing
•	\bullet	Δ	\Delta, \triangle
⋅	\cdot	∇	\nabla
⋯	\cdots	□	\square, \Box
⋮	\vdots	○	\circ
⋱	\ddots	∠	\angle
ℏ	\hbar	⊥	\perp
∞	\infty	▭	\parallelogram

(要先输入下方代码, \parallelogram命令才能生效)。

```
\usepackage{tikz}
\newcommand\parallelogram
{\mathord{\text{
\tikz[baseline]
\draw (0em, .1ex) -- ++(0.8em, 0ex)
-- ++(0.2em, 1.2ex) -- ++(-0.8em, 0ex)
-- cycle;}}}}
```

角度 45°, 45[~]\circ
射影对应 $\bar{\wedge}$ \barwedge; $\overline{\wedge}$ \overline{\wedge}
透视对应 $\bar{\bar{\wedge}}$ \bar{\bar{\wedge}}

$\overline{\overline{\wedge}}$ `\overline{\overline{\wedge}}`

7. 极限、连加、连乘、积分:

$\lim_{n \rightarrow \infty}$ `\lim_{n \to \infty}`

$\overline{\lim}_{n \rightarrow \infty}$ `\varlimsup_{n \to \infty}`

$\underline{\lim}_{n \rightarrow \infty}$ `\varliminf_{n \to \infty}`

$\sum_{n=1}^{\infty}$ `\sum_{n=1}^{\infty}`

$\prod_{n=1}^{\infty}$ `\prod_{n=1}^{\infty}`

$\int_0^{+\infty}$ `\int_0^{+\infty}`

以上代码在行内公式中效果如上,而在行间公式中的效果如下:

$$\lim_{n \rightarrow \infty}, \sum_{n=1}^{\infty}, \prod_{n=1}^{\infty}, \int_0^{+\infty}$$

如果要在行内显示跟行间一样的效果,则加上`\limits`或`\displaystyle`,如下:

`\lim\limits_{n \to \infty}`

`\sum\limits_{n=1}^{\infty}`

`\prod\limits_{n=1}^{\infty}`

`{ \displaystyle \int_0^{+\infty} }`

可以为`\lim\limits_{}`, `\sum\limits_{}`, `\prod\limits_{}`, `\displaystyle`等常用代码指定快捷键,提高效率。如果在`\begin{document}`之前加上一句`\everymath{\displaystyle}`,那么所有行内公式按行间样式显示,`\lim`,`\sum`,`\prod`无需加`\limits`,而`\int`无需加`\displaystyle`,但副作用是会让行内的连加号、连乘号十分巨大,对比如下

$$\sum_{n=1}^{\infty}, \prod_{n=1}^{\infty}, \sum_{n=1}^{\infty}, \prod_{n=1}^{\infty}$$

虽然`\usepackage{bigints}`后可以用命令

`\bigintssss, \bigintsss, \bigintss, \bigints, \bigint`

获得不同大小的积分号(而无需`\displaystyle`),但这个系列的积分号太粗了,不美观。

比如`\bigintss`: $\int_0^{+\infty} \sin(x^2)dx = \frac{\sqrt{2\pi}}{4}$.

多重极限:

$\lim_{\substack{x \rightarrow x_0 \\ y \rightarrow y_0}}$ `\lim\limits_{x \to x_0 \atop y \to y_0}`

$\lim_{\substack{w \rightarrow w_0 \\ x \rightarrow x_0 \\ y \rightarrow y_0 \\ z \rightarrow z_0}}$

`\lim\limits_{\substack{w \to w_0 \\ x \to x_0 \\ y \to y_0 \\ z \to z_0}}`

8. 括号:

`\left(\right)`, `\left[\right]`

`\left\{ \right\}`, `\left| \right|`

`\big, \bigg, \bigl, \bigr, \biggl, \biggr`

`\Big, \Bigg, \Bigl, \Bigr, \Biggl, \Biggr`

直接使用`()`,`[]`,`{}`,括号的高度不会随着括号中的内容高度而变化,比如 $(\frac{3}{4})^2, [\frac{\pi^2}{6}], \{\frac{\pi^4}{90}\}$.

使用`\left(\right)`,则能让括号随内容增高而变高,比如 $(\frac{\pi^2}{6})^2$.

使用`\bigg(\bigg)`, `\Bigg(\Bigg)`之类的,不同的命令代表不同尺寸的括号,而与括号中的内容无关。使用`\left \right`时,内部不能出现换行符`\`,若需要较高的括号,就要改用`\bigg(\bigg)`等。

如果只需要显示单侧的括号(最常见的情形是只显示左大括号),现以左侧为例,`\right`不能省略,必须与`\left`配对出现,配对方法是要把右括号改成小数点,即输入`\right.`,比如 $\left\{\frac{\pi^2}{6}\right.$ 的代码是

`\left\{\dfrac{\pi^2}{6}\right.`

而`\big`系列的括号可以直接省去一侧,无需配对出现。

$\left|\frac{a}{b}\right|$ `\left|\dfrac{a}{b}\right|`
 $\left\langle\frac{a}{b}\right\rangle$ `\left\langle\dfrac{a}{b}\right\rangle`
 $\left\lfloor\frac{a}{b}\right\rfloor$ `\left\lfloor\dfrac{a}{b}\right\rfloor`
 $\left\lceil\frac{a}{b}\right\rceil$ `\left\lceil\dfrac{a}{b}\right\rceil`

9. 行内公式 $a^2 + b^2 = c^2$ 使用`$ a^2+b^2=c^2 $`即可。

行间公式可用语法很多,比如`[\]`,`$$` `$$`,这两种环境只能输入单行公式,换行符`\`在其中无效。行间公式还可以用`\begin{xx} \end{xx}`之类,其中`xx`可以是

`align(*), alignat(*), flalign(*)`
`equation(*), gather(*), multline(*)`

带`*`的环境不给公式编号,不带`*`的环境自动给公式编号,使用`\notag`或`\nonumber`可隐藏任意一行公式的编号。`equation(*)`也只能输入单行公式,换行符`\`在其中无效,但在其中嵌入`split`环境后就能输入多行公式了,好处是多行公式只有一个编号。

`\begin{equation} \label{aaa1}`
`\begin{split}`
 $x^4 + 2x^3 + 11x^2 + 18x + 18$
 $= (x^2 + 2x + 2)(x^2 + 9)$
 $= (x^2 + x + 3)^2 + (2x + 3)^2$
`\end{split}`
`\end{equation}`

$$\begin{aligned} & x^4 + 2x^3 + 11x^2 + 18x + 18 \\ &= (x^2 + 2x + 2)(x^2 + 9) \\ &= (x^2 + x + 3)^2 + (2x + 3)^2 \end{aligned} \tag{1}$$

用`\label{aaa1}`给公式加标签,然后用`\ref{aaa1}`引用公式(的编号),`\pageref{aaa1}`引用公式所在的页码。`\usepackage{hyperref}`可以让生成的PDF文件

带有书签以及可点击跳转的超链接, 比如公式 (1), (2).

第 1 次编译可能会报如下错误:

```
Paragraph ended before \Hy@setref@link
was complete.
```

不需任何操作, 直接编译第 2 次即可成功。

`alignat`和`align`环境区别如下 (不明显, `align`整体稍微宽一点):

```
\begin{alignat*}{3}
  2x+3 &= 5678y-8765z &+ 20 \\
  4x &= y+z &+ 11112222
\end{alignat*}
```

$$\begin{array}{rcl} 2x+3 & = & 5678y-8765z +20 \\ 4x & = & y+z +11112222 \end{array}$$

```
\begin{align*}
  2x+3 &= 5678y-8765z &+ 20 \\
  5x &= y+z &+ 33334444
\end{align*}
```

$$\begin{array}{rcl} 2x+3 & = & 5678y-8765z +20 \\ 5x & = & y+z +33334444 \end{array}$$

`gather(*)`环境中不能出现对齐符号`&`, 否则报错。此环境下所有行的公式全部居中对齐。

```
\begin{gather*}
  2x+3 = 5678y-8765z + 20 \\
  6x = y+z + 55556666
\end{gather*}
```

$$\begin{array}{c} 2x+3 = 5678y-8765z + 20 \\ 6x = y+z + 55556666 \end{array} \quad (2)$$

`cases`环境对于带左大括号的情形特别有用, 比如分段函数、方程联立等,

```
\begin{align*}
  \begin{cases}
    2x+3y=7 \\
    3x+5y=8
  \end{cases}
\end{align*}
```

$$\begin{cases} 2x+3y=7 \\ 3x+5y=8 \end{cases}$$

虽然用

```
\begin{align*}
  \left\{ \begin{array}{l}
    2x+3y=7 \\
    3x+5y=8
  \end{array} \right.
\end{align*}
```

```
& 2x+3y=7 \\
& 3x+5y=8
\end{aligned}
\right.
\end{align*}
```

也能实现同样效果, 但显然是`cases`更方便。

`multline(*)`环境第一行左对齐, 中间的行居中对齐, 最后一行右对齐, 用得较少。

```
\begin{multline}
  1-line \\
  2-line \\
  3-line \\
  4-line
\end{multline}
```

$$\begin{array}{lcl} 1-line & & \\ & 2-line & \\ & 3-line & \\ & & 4-line \end{array} \quad (3)$$

公式环境中要加汉字, 则必须置于`\text{}`之内。

实现文本居中对齐使用`center`环境

```
\begin{center}
  ...
\end{center}
```

以上给公式外围加边框用的是:

```
\begin{align*}
  \boxed{
    \begin{aligned}
      \dots\dots
    \end{aligned}
  }
\end{align*}
```

或

```
\usepackage{tcolorbox}
\tcbset{before={\noindent},
  after={\noindent},colback=white}
\begin{tcolorbox}
  \vspace{-5mm}
  \begin{align*}
    \dots\dots
  \end{align*}
\end{tcolorbox}
```

10. 想让公式编号带上“章”序号或“节”序号, 可使用

```
\numberwithin{equation}{chapter}
\numberwithin{equation}{section}
```

不想让公式、表格、图片带上章号、节号，

```
\usepackage{chngcntr}
\counterwithout{equation}{chapter}
\counterwithout{equation}{section}
```

其中的equation还可以换成table和figure.

`\usepackage{chngcntr}`不能与
`\usepackage[leqno]{amsmath}` 同时使用，否则会报错：(选项冲突)

```
Option clash for package amsmath.
```

但可以用`\usepackage{amsmath}`，加`[leqno]`的效果是
让公式编号位于左侧，如果要想实现这一效果，可以使用
`\documentclass[a4paper,leqno]{article}` 更改编号样式

```
\renewcommand{\thetable}{\Roman{table}}
\renewcommand{\thefigure}{\Roman{figure}}
```

`\arabic`，阿拉伯数字；`\roman`：小写罗马数字；`\Roman`：大写罗马数字；`\alph`：小写字母；`\Alph`：大写字母。

11. 要让全文的行间公式居左 (默认是居中)，使用

```
\usepackage[fleqn]{amsmath}
```

如果同时需要公式编号在左侧，那么使用

```
\usepackage[leqno,fleqn]{amsmath}
```

如果只想要让单个公式居左，使用

```
\begin{flalign}
E=mc^2 \&\&
\end{flalign}
```

效果如下：

$$E = mc^2 \tag{4}$$

特别注意最后的两个对齐符号`&&`，如果漏掉，则没有居左的效果。

12. 矩阵和行列式：

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

```
\begin{pmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{pmatrix}
```

$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 用 `bmatrix`， $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$ 用 `vmatrix`
不带括号和竖线用`matrix`，大括号用`Bmatrix`，双竖线用`Vmatrix`。

三种省略号：`\cdots`，`\vdots`，`\ddots`

13. 函数：

```
\arg,\exp,\inf,\sup,\max,\min
\sin,\sinh,\arcsin,\cos,\cosh,\arccos
```

```
\tan,\tanh,\arctan
\log,\ln,\lg,\deg,\det,\dim
```

这些函数只能在公式环境中使用，而且字体是正体，如果不在前面加 `\`，直接输入 `sin,cos,log`，字体就是斜体。

14. 公式环境中，某些特殊含义的字母需要用正体而非斜体，比如自然对数底数 `e`，虚数单位 `i` 和微分符号 `d`，使用`\mathrm{e}`即可，比如

```
\mathrm{e}^{\mathrm{i}\theta} =
\cos\theta + \mathrm{i}\sin\theta \quad \backslash
\int_0^{+\infty} \frac{x}{e^x - 1} dx = \frac{\pi^2}{6}
-1\mathrm{d}x = \frac{\pi^2}{6}
```

$$e^{i\theta} = \cos \theta + i \sin \theta$$
$$\int_0^{+\infty} \frac{x}{e^x - 1} dx = \frac{\pi^2}{6}$$

公式环境下 `e`、`i`、`d` 都不用正体的效果是：`e,i,d`。

15. 自定义新的命令：`\newcommand{ }{ }`，效果类似于 C 语言的宏替换`#define`。比如嫌`\quad`太麻烦，可以先

```
\newcommand{\q}{\quad}
```

然后就能用`\q`代替`\quad`。在

```
\newcommand{\im}{\mathrm{i}}
```

之后，就能用`\im`实现正体的虚数单位 `i`。因为`\i`已经在某个包中定义过了，所以也可以用

```
\renewcommand{\i}{\mathrm{i}}
```

覆盖掉`\i`的定义。对 `e` 和 `d` 可类似处理，提高输入效率。个人的一些习惯如下：

```
\renewcommand{\leq}{\leqslant}
\renewcommand{\geq}{\geqslant}
\renewcommand{\vec}{\overrightarrow}
\renewcommand{\Re}{\mathrm{Re}}
\renewcommand{\Im}{\mathrm{Im}}
\renewcommand{\d}{\mathrm{d}}
\renewcommand{\i}{\mathrm{i}}
\newcommand{\e}{\mathrm{e}}
\newcommand{\q}{\quad}
\newcommand{\disp}{\displaystyle}
```

16. 希腊字母：

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
δ	<code>\delta</code>	ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>
ξ	<code>\xi</code>	π	<code>\pi</code>	ρ	<code>\rho</code>
σ	<code>\sigma</code>	τ	<code>\tau</code>	ϕ	<code>\phi</code>
φ	<code>\varphi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>

以下字母存在大写形式 (省略了一些带\var前缀的), 只需把首字母大写即可。

\Gamma, \Delta, \Theta, \Lambda, \Xi, \O, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega

17. 字母上下加符号:

\bar{a} \overline{a}	\tilde{a} \tilde{a}
\underline{a} \underline{a}	\widetilde{abc} \widetilde{abc}
\overbrace{a} \overbrace{a}	\bar{a} \bar{a}
\underbrace{a} \underbrace{a}	\vec{a} \vec{a}
\overleftarrow{a} \overleftarrow{a}	\hat{a} \hat{a}
\overrightarrow{a} \overrightarrow{a}	\widehat{abc} \widehat{abc}
$\stackrel{b}{a}$ \stackrel{b}{a}	\check{a} \check{a}
$\overset{b}{a}$ \overset{b}{a}	\breve{a} \breve{a}
$\underset{b}{a}$ \underset{b}{a}	\dot{a} \dot{a}
\acute{a} \acute{a}	\ddot{a} \ddot{a}
\grave{a} \grave{a}	\dddot{a} \dddot{a}

18. 中文加下划线: (需\usepackage{ulem})

单下划线 \underline{}	删除线 \sout{}
双下划线 \uuline{}	虚下划线 \dashuline{}
波浪线 \uwave{}	点下划线 \dotuline{}

19. 箭头:

\rightarrow \to	\leftarrow \leftarrow
\rightarrow \rightarrow	\Leftarrow \Leftarrow
\Rightarrow \Rightarrow	\Uparrow \uparrows
\longrightarrow \longrightarrow	\Downarrow \downarrows
$\xrightarrow[c,d]{a,b}$ \xrightarrow[c,d]{a,b}	
$\xrightarrow[140^\circ]{\text{稀硫酸}}$ (\xrightarrow[140^\circ]{\text{稀硫酸}}需\usepackage{extarrows})	
$\xrightarrow[140^\circ]{\text{稀硫酸}}$ \xrightarrow[140^\circ]{\text{稀硫酸}}	

20. 插入表格:

```
\begin{tabular}{|c|c|}
\hline
& \\
\hline
& \\
\hline
\end{tabular}
```

表格内换行:

```
\usepackage{makecell}
\makecell[l]{第一行 \\ 第二行 \\ ...}
```

合并单元格则使用\multicolumn和\multirow.

跨页的长表格使用\begin{longtable} ...

表格行距控制:\renewcommand{\arraystretch}{1.5}

21. 插入图片:

```
\usepackage{graphicx}
\begin{figure}
\centering
\includegraphics[width=
0.3\linewidth]{图片名}
\caption{图片标题}
\label{xxx1}
\end{figure}
```

位置控制:h(here) t(top) b(bottom) p ! H(Here), H比h更强, 如果使用h后图片还是跑到意料之外的位置, 请改用H. 使用H需\usepackage{float}.

p代表page containing only floats, such as figures and tables.

!代表allows to ignore certain parameters of LaTeX for float placement.

四种宽度:

\linewidth	当前行的宽度
\columnwidth	当前分栏的宽度
\textwidth	整个页面版芯的宽度
\paperwidth	整个页面纸张的宽度

22. 添加页眉页脚:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\lhead{左页眉}
\chead{中页眉}
\rhead{右页眉}
\lfoot{左页脚, 修订日期\today}
\cfoot{中页脚, 第\thepage页}
\rfoot{右页脚}
```

23. 添加水印:

使用{xwatermark}包会遇到报错

```
Extra \endgroup. \begin{document}
```

{background}包第一页水印的颜色比后面的页更深, 第二页水印内容也有异常; {watermark}(2004)和{draftmark}(2009)太旧, 均无法使用。

下面给出{draftwatermark}用法示例, 但这个包有时会出现水印文字重叠到一起的问题 (本文档编译时经常遇到这个问题, 但不是100%出现)。

```
\usepackage{draftwatermark}
\usepackage{everypage}
\SetWatermarkText{磁悬浮青蛙呱呱呱, 水印}
\SetWatermarkLightness{0}
\SetWatermarkAngle{80}
\SetWatermarkColor{gray}
\SetWatermarkScale{0.07}
```

24. 带圈数字,

方法一: `\usepackage{pifont},`

①②...⑨⑩ `\ding{172} ... \ding{181}`

①②...⑨⑩ `\ding{182} ... \ding{191}`

①②...⑨⑩ `\ding{192} ... \ding{201}`

①②...⑨⑩ `\ding{202} ... \ding{211}`

方法二: `\usepackage{tikz}`, 然后定义一个新的命令:

```
\newcommand*{\mycircled}[1]{\lower
.7ex\hbox{\tikz\draw (0pt, 0pt) circle
(.4em) node {\makebox[0.5em][c]
{\small #1}};}}
```

①②⑨⑩ `\mycircled{1}\mycircled{2} ...`

当`\mycircled{}`命令出现在行首的时候, 会遇到报错:

You can't use '\lower' in vertical mode.

解决方案是在`\mycircled{}`之前加一个波浪号~ (波浪号不会出现在编译结果中), 或者把`\mycircled{}`放在行内公式环境中, 比如美元符号内。

方法三: `\textcircled{}`, 数字在圈内偏上, 没有居中, 不推荐。

①②⑨ ⑩⑩⑩⑩⑩⑩ `\textcircled{1} ...`

25. 添加带编号脚注¹: `\footnote{}`.

无编号脚注: (自定义了`\myfootnote`命令)

```
\newcommand{\myfootnote}[1]{
\renewcommand{\thefootnote}{}
\footnotetext{\scriptsize#1}
\renewcommand{\thefootnote}{
\arabic{footnote}} }
```

把脚注编号改为带圈数字:

```
\renewcommand{\thefootnote}{
\ding{\numexpr171+\value{footnote}}}
\newcommand{\myfootnote}[1]{
\renewcommand{\thefootnote}{}
\footnotetext{\scriptsize#1}
\renewcommand{\thefootnote}{
\ding{\numexpr171+\value{footnote}}}}
```

26. 允许公式跨页: `\allowdisplaybreaks`

27. 新增空白页:

`\newpage, \clearpage, \cleardoublepage`

28. 目录: `\tableofcontents`

设置目录深度: `\setcounter{tocdepth}{3}`

设置在几级目录前标记序号:

`\setcounter{secnumdepth}{4}`

¹这是用 `\footnote{}` 添加的带编号脚注。

这是用 `\myfootnote{}` 添加的无编号脚注。

29. 字体大小控制:

```
\tiny, \scriptsize, \footnotesize
\small, \normalsize
\large, \Large, \LARGE, \huge, \Huge
```

文本行距控制: `\linespread{1.3}`(必须放在`\begin{document}`之前)

30. 粗体: `\textbf{}`, 使用时如果恰好换行, 在 tex 源码中让`\textbf{}`处于新一行, 则编译后的粗体前面会多一个空格, 解决方案就是不要恰好在`\textbf{}`前面换行。斜体命令`\textit{}`只对英文有效, 对中文无效。对英文同时斜体和粗体则需`\usepackage{bm}`, **AB**: `\bm{AB}`31. 用以下命令修改全局字体, 需`\usepackage{fontspec}`

设置英文字体: `\setmainfont{Microsoft YaHei}`

设置 C(中文)、J(日文)、K(韩文) 的字体:

```
% 特别注意大括号中的字体名称区分大小写
\setCJKmainfont{SimSun} % 宋体
\setCJKmainfont{FangSong} % 仿宋
```

其它一些字体名称: NSimSun(新宋体), STFangsong(华文仿宋), STZhongsong(华文中宋), STXihei(华文细黑), KaiTi(楷体), STKaiti(华文楷体), SimHei(黑体), Microsoft YaHei(微软雅黑), LiSu(隶书), STLiti(华文隶书), YouYuan(幼圆)。

设置文本颜色(`textcolor`):

```
\textcolor{red}{设置}...
```

修改局部的字体, 只需用大括号包住, 然后加上“\字体英文名称”即可。字体英文名称可以通过在 Windows 的 cmd 或 Powershell 中, 或者 Linux 的 shell 中, 输入`fc-list`来查看, 查询结果非常多, 可以用`fc-list :lang=zh`(注意是英文冒号, 而且冒号前有一个空格)筛选中文字体, 用`fc-list :lang=en`筛选英文字体。

```
\newfontfamily{\courier}{Courier New}
\newfontfamily{\tinro}{Times New Roman}
\newfontfamily{\airal}{Arial}
\newfontfamily{\calibri}{Calibri}
\newfontfamily{\cambria}{Cambria}
\newfontfamily{\consolas}{Consolas}

{\courier English Courier New font}
{\tinro English Times New Roman}
{\airal English Arial font show}
{\calibri English Calibri font show}
```

```
{\cambria English Cambria font show}
{\consolas English Consolas font}

English Courier New font show
English Times New Roman show
English Arial font show
English Calibri font show
English Cambria font show
English Consolas font show

{\songti 宋体-磁悬浮青蛙呱呱呱}
{\heiti 黑体-磁悬浮青蛙呱呱呱}
{\fangsong 仿宋-磁悬浮青蛙呱呱呱}
{\kaishu 楷书-磁悬浮青蛙呱呱呱}
{\lishu 隶书-磁悬浮青蛙呱呱呱}
{\youyuan 幼圆-磁悬浮青蛙呱呱呱}
{\yahei 雅黑-磁悬浮青蛙呱呱呱}

宋体-磁悬浮青蛙呱呱呱
黑体-磁悬浮青蛙呱呱呱
仿宋-磁悬浮青蛙呱呱呱
楷书-磁悬浮青蛙呱呱呱
隶书-磁悬浮青蛙呱呱呱
幼圆-磁悬浮青蛙呱呱呱
雅黑-磁悬浮青蛙呱呱呱
```

32. 部分 `\part{}` 章 `\chapter{}`
节 `\section{}` 小节 `\subsection{}`

33. 带编号列表:

```
\usepackage{enumerate}
\begin{enumerate}[(1)]
\item 有界变差函数
\item 可测函数
\end{enumerate}
```

不带编号列表:

```
\begin{itemize}
\item 控制收敛定理
\item Levi引理和Fatou引理
\end{itemize}
```

`\usepackage{paralist}`后,
用`\begin{compactenum}` `\end{compactenum}`
代替`\begin{enumerate}` `\end{enumerate}`,
用`\begin{compactitem}` `\end{compactitem}`
代替`\begin{itemize}` `\end{itemize}`,
用`\begin{compactdesc}` `\end{compactdesc}`
代替`\begin{description}` `\end{description}`,
可以让 item 之间的空白更小。
`{paralist}`和`{enumitem}`包存在冲突,同时使用这两个包时,会报错:

Undefined control sequence.
`\end{enumerate}`
Missing number, treated as zero.
`\end{enumerate}`

建议抛弃`{paralist}`,只使用`{enumitem}`,在

```
\begin{enumerate}
\begin{itemize}
\begin{description}
```

后面加上`[itemsep=0pt]`来实现空白更小的效果,其中的0pt还可以设成-1pt,-2pt等负数。

34. 常用长度单位: 毫米 (mm), 厘米 (cm), 点 (pt), ex, em

35. 交换图:

$$\begin{array}{ccc} V & \xrightarrow{\varphi} & U \\ \eta_1 \downarrow & & \downarrow \eta_2 \\ \mathbb{K}_n & \xrightarrow{\varphi_A} & \mathbb{K}_m \end{array}$$

```
\usepackage[all]{xy}
\begin{displaymath}
\xymatrix{
V \ar[r]^{\bm{\varphi}} \\
\ar[d]_{\bm{\eta}_1} & & \ar[d]_{\bm{\eta}_2} \\
\mathbb{K}_n \ar[r]^{\bm{\varphi}_A} & & \mathbb{K}_m
}
& U \ar[d]_{\bm{\eta}_2} & \\
& \mathbb{K}_m &
\end{array}
```

36. 空格与空白:

负空格 <code>\!</code>	词间空格 <code>\</code>
窄空格 <code>\,</code>	四倍空格 <code>\quad</code>
中等空格 <code>\:</code>	八倍空格 <code>\qqquad</code>
宽空格 <code>\;</code>	

注意,“词间空格”的斜杠后有一个看不见的空格。
取消首行缩进: `\noindent`
水平空白 `\hspace{±2cm}`
垂直空白 `\vspace{±2cm}`
缩小行间公式与上下文之间的空白 (必须放在
`\begin{document}`之前):

```
\AtBeginDocument{
\addtolength{\abovedisplayskip}{-2ex}
\addtolength{\abovedisplayshortskip}{-2ex}
\addtolength{\belowdisplayskip}{-2ex}
\addtolength{\belowdisplayshortskip}{-2ex}
}
```

37. 设置页边距:

```
\usepackage{geometry}
\geometry{a4paper,left=1cm,right=1cm,
```

top=1.5cm,bottom=1.5cm}

38. 英文字母几种变体效果如下:
\mathcal{}(只能用于大写字母, 对小写无效)
ABCDEFGHIJKLMNOPQRSTUVWXYZ
\mathscr{}(只能用于大写字母, 需{mathrsfs})
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
\mathbb{}(只能用于大写字母, 需{amssymb})
ABCDEFGHIJKLMNOPQRSTUVWXYZ
\mathfrak{}(同时适用于大小写, 需{amssymb})
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz

39. 防止 ff, fi, ffi, fl 变成连体 (Ligature): ff,fi,ffi,fl
方法一: f{}f,f{}i,f{}f{}i,f{}l
方法二: f{f},f{i},f{f}i, f{l}
连体可能导致从编译的 PDF 中复制文字或者取词翻译出问题, 是我最讨厌的 L^AT_EX 特性。禁用所有连写的命令我还没有测试成功。

40. 本手册使用了多栏环境
\usepackage{multicol}
\begin{multicols}{2}

\end{multicols}
以及带编号列表环境enumerate, 用
\columnseprule 1pt
显示中央分隔竖线并控制线宽。用
\columnsep 20pt

磁心, 晶体管, 磁鼓, 水印

控制两栏之间的间隔。显示 L^AT_EX 代码使用了两种方法, 较短的代码使用了\verb| |, 大片的代码使用了

```
\usepackage{listings}
\lstset
{ language=[LaTeX]TeX,
  backgroundcolor=\color{gray!20},
  basicstyle=\tt\normalsize,
  aboveskip=0pt,
  belowskip=0pt, }

\begin{lstlisting}

\end{lstlisting}
```

除了lstlisting, 也可以使用

```
\begin{verbatim}

\end{verbatim}
```

41. 以下三个网站可以在线写作以及编译 L^AT_EX:

```
https://www.texpage.com/
https://www.slager.cn/
https://cn.overleaf.com/
```

以下网站可以识别单个手写的 L^AT_EX 符号, 并提供可能的 L^AT_EX 代码。

```
http://detexify.kirelabs.org/classify.html
```

Mathpix snip 软件 (Win,MacOS,Linux,IOS,Android 均支持) 能识别手写或印刷的数学公式、英文和汉字, 包括矩阵和表格等, 然后生成完整的 L^AT_EX 代码, 而且准确率很高, 值得尝试。