

- 前言
- 受支持的功能
- 声调记号
- 定界符（括号之类）
 - 定界符大小
- 环境
 - 数学字母、符号
 -
- 布局
 - 注解/标记
 - 换行指令
 - 垂直布局
- 重叠和间距
 - 间隔
 - 提示
- 逻辑与集合符号
- 宏定义
- 运算符
 - 巨型运算符
 - 二元运算符
 - 分数和二项式
 - 数学运算符
 - 开根
 - 关系指令
 - 箭头
 - 箭头扩展
- 样式，颜色，大小和字体
 - 类的分配
 - 颜色
 - 字体
 - 大小
 - 样式
- 符号和标点
- 单位

前言

由于在洛谷，有很多人对于 $\text{K}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 之间的关系并不清楚，导致很多人去搜 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 的资料，然后发现有许多指令无法在洛谷运行。

但是事实上， $\text{K}\text{A}\text{T}\text{E}\text{X}$ 才是洛谷上的 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 的渲染机制。因此只有 $\text{K}\text{A}\text{T}\text{E}\text{X}$ 支持的指令才能正常使用。在 $\text{K}\text{A}\text{T}\text{E}\text{X}$ 官网，其实有两张支持表，这里搬运过来以便于大家能够正确使用指令。

这张表应该涵盖了纯 $\text{KAT}_{\text{E}}\text{X}$ 所有的功能。当然， $\text{LAT}_{\text{E}}\text{X}$ 公式大全亦可作为部分参考。下面是搬运部分。

受支持的功能

这是一张由 $\text{KAT}_{\text{E}}\text{X}$ 支持的 $\text{T}_{\text{E}}\text{X}$ 公式清单。他按照逻辑顺序分类。

这有一个相似的按照字典序排列的[支持表](#)。那张表格同样写出了不受支持的公式。

声调记号

第一列	第二列	第三列
\acute{a} <code>a'</code>	\tilde{a} <code>\tilde{a}</code>	\mathring{g} <code>\mathring{g}</code>
$\acute{\acute{a}}$ <code>a''</code>	\widetilde{ac} <code>\widetilde{ac}</code>	\overgroup{AB} <code>\overgroup{AB}</code>
\acute{a} <code>a^{\prime}</code>	\utilde{AB} <code>\utilde{AB}</code>	\undergroup{AB} <code>\undergroup{AB}</code>
\acute{a} <code>\acute{a}</code>	\vec{F} <code>\vec{F}</code>	\overrightarrow{AB} <code>\overrightarrow{AB}</code>
\bar{y} <code>\bar{y}</code>	\overleftarrow{AB} <code>\overleftarrow{AB}</code>	\overrightarrow{AB} <code>\overrightarrow{AB}</code>
\breve{a} <code>\breve{a}</code>	\underleftarrow{AB} <code>\underleftarrow{AB}</code>	\underrightarrow{AB} <code>\underrightarrow{AB}</code>
\check{a} <code>\check{a}</code>	\overleftharpoonup{ac} <code>\overleftharpoonup{ac}</code>	$\overrightarrow{\tharpoonup{ac}}$ <code>\overrightarrow{\tharpoonup{ac}}</code>
\dot{a} <code>\dot{a}</code>	\overleftrightarrow{AB} <code>\overleftrightarrow{AB}</code>	\overbrace{AB} <code>\overbrace{AB}</code>
\ddot{a} <code>\ddot{a}</code>	\underleftrightarrow{AB} <code>\underleftrightarrow{AB}</code>	\underbrace{AB} <code>\underbrace{AB}</code>
\grave{a} <code>\grave{a}</code>	\overline{AB} <code>\overline{AB}</code>	$\overline{\hspace{0.5em}AB\hspace{0.5em}}$ <code>\overlinesegment{AB}</code>
$\hat{\theta}$ <code>\hat{\theta}</code>	\underline{AB} <code>\underline{AB}</code>	$\underline{\hspace{0.5em}AB\hspace{0.5em}}$ <code>\underlinesegment{AB}</code>
\widehat{ac} <code>\widehat{ac}</code>	\widecheck{ac} <code>\widecheck{ac}</code>	

另外，还有一些上下音标适合用在`\text{}`环境下。^[1]

第一列	第二列	第三列	第四列
\acute{a} <code>\'a</code>	\tilde{a} <code>\~a</code>	\acute{a} <code>\.a</code>	\grave{a} <code>\H{a}</code>
\grave{a} ^[2]	\bar{a} <code>\={a}</code>	\grave{a} <code>\"a</code>	\grave{a} <code>\v{a}</code>
\hat{a} <code>\^{a}</code>	\check{a} <code>\u{a}</code>	\grave{a} <code>\r{a}</code>	

[1]：这类特殊的音标需要用在文字环境下，即`\text{}`里面。比如`\text{\'a}`

[2]：由于会和Markdown的代码块冲突，这里说明一下，这个音标的代码其实是`\+波浪线对应按键+{}`。

另请参阅 字母 章节。

定界符（括号之类）

第一列	第二列	第三列	第四列	第五列
$()$	$()$ <code>\lparen\rparen</code>	$[]$	$\lceil\rceil$ <code>\lceil\rceil</code>	\uparrow <code>\uparrow</code>
$[\]$	$[\]$ <code>\lbrack\rbrack</code>	$\lfloor\rfloor$	$\lfloor\rfloor$ <code>\lfloor\rfloor</code>	\downarrow <code>\downarrow</code>
$\{\}$	$\{\}$ <code>\lbrace\rbrace</code>	$\{ \}$ <code>\lgroup\rgroup</code>	$\{ \}$ <code>\lgroup\rgroup</code>	\updownarrow <code>\updownarrow</code>
$\langle\rangle$	$\langle\rangle$ <code>\lang\rang</code>	$\langle\rangle$ <code>\lang\rang</code>	$\langle\rangle$ <code>\lang\rang</code>	\Uparrow <code>\Uparrow</code>
$\lvert\rvert$	$\lvert\rvert$ <code>\lvert\rvert</code>	\lvert <code>\lvert</code>	\lrcorner <code>\lrcorner</code>	\ulcorner <code>\ulcorner</code>
$\lvert\rvert$	$\lvert\rvert$ <code>\lvert\rvert</code>	\lvert <code>\lvert</code>	\llcorner <code>\llcorner</code>	\Updownarrow <code>\Updownarrow</code>
$\lvert\rvert$	$\lvert\rvert$ <code>\lvert\rvert</code>	\lvert <code>\lvert</code>	\lrcorner <code>\lrcorner</code>	\backslash <code>\backslash</code>
$\langle\rangle$	$\langle\rangle$ <code>\lt\gt</code>	$\llbracket\rrbracket$	$\llbracket\rrbracket$ <code>\llbracket\rrbracket</code>	$\{ \}$ <code>\lBrace\rBrace</code>

[3] : 由于Markdown的制表与|有冲突，因此无法在表格中显示|。

[4] : \left和\right后面直接加括号，能显示括号。（人类迷惑行为）

定界符大小[5]

(AB)
`\left(\LARGE{AB}\right)`

$((((((\big(\Big(\bigg(\Bigg($

[5] : 这里定界符主要指括号。K_AT_EX提供了一些指令来调整括号的大小。另外，这里\left,\right指令是自动调整括号大小。

第一列	第二列	第三列	第四列	第五列
\left	\big	\bigl	\bigm	\bigr
\middle	\Big	\Bigl	\Bigm	\Bigr

第一列	第二列	第三列	第四列	第五列
<code>\right</code>	<code>\bigg</code>	<code>\biggl</code>	<code>\biggm</code>	<code>\biggr</code>
	<code>\Bigg</code>	<code>\Biggl</code>	<code>\Biggm</code>	<code>\Biggr</code>

环境^[6]

第一列	第二列	第三列	第四列
$\begin{matrix} a & b \\ c & d \end{matrix}$	<code>\begin{matrix}</code> <code>a&b\\</code> <code>c&d</code> <code>\end{matrix}</code>	$\begin{matrix} a & b \\ c & d \end{matrix}$	<code>\begin{array}{cc}</code> <code>a&b\\</code> <code>c&d</code> <code>\end{array}</code>
$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	<code>\begin{pmatrix}</code> <code>a&b\\</code> <code>c&d</code> <code>\end{pmatrix}</code>	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	<code>\begin{bmatrix}</code> <code>a&b\\</code> <code>c&d</code> <code>\end{bmatrix}</code>
$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	<code>\begin{vmatrix}</code> <code>a&b\\</code> <code>c d</code> <code>\end{vmatrix}</code>	$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$	<code>\begin{Vmatrix}</code> <code>a&b\\</code> <code>c d</code> <code>\end{Vmatrix}</code>
$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$	<code>\begin{Bmatrix}</code> <code>a&b\\</code> <code>c d</code> <code>\end{Bmatrix}</code>	$\begin{array}{ c c c } \hline a & b & c \\ \hline d & e & f \\ \hdashline g & h & i \end{array}$	<code>\def\arraystretch{1.5}</code> <code>\begin{array}{c : c : c}</code> <code>a&b&c\\\hline</code> <code>d&e&f\\</code> <code>\hdashline</code> <code>g&h&i</code> <code>\end{array}</code>
$\begin{aligned} a &= b + c \\ d + e &= f \end{aligned}$	<code>\{aligned}</code> <code>a&= b + c\\</code> <code>d + e&= f</code> <code>\end{aligned}</code>	$\begin{aligned} 10x + 3y &= 2 \\ 3x + 13y &= 4 \end{aligned}$	<code>\begin{alignedat}{2}</code> <code>10&x + &3&y = 2\\</code> <code>3&x + &13&y = 4</code> <code>\end{alignedat}</code>
$\begin{gathered} a = b \\ e = b + c \end{gathered}$	<code>\begin{gathered}</code> <code>a = b\\</code> <code>e = b + c</code> <code>\end{gathered}</code>	$x = \begin{cases} a & \text{if } b \\ c & \text{if } d \end{cases}$	<code>x = \begin{cases}</code> <code>a&\{if\}b\\</code> <code>c&\{if\}d</code> <code>\end{cases}</code>

第一列	第二列	第三列	第四列
	$\begin{smallmatrix}$		
$a \ b$	$a&b\\$		
$c \ d$	$c \ d$		
	$\end{smallmatrix}$		

[6]：因为相关原因，在表格里换行不是很方便，因此这里直接嵌入的是K^AT_EX。

Unicode^[7]数学字母、符号

字段	支持的字符	字段	支持的字符
粗体	A-Z,a-z,0-9	双线体	A – Z, k
意大利斜体	<i>A-Z,a-z</i>	无衬线字体	A–Z,a–z,0–9
粗意大利斜体	<i>A-Z,a-z</i>	粗体无衬线	A–Z,a–z,0–9
手写体	A – Z	斜体无衬线	<i>A–Z,a–z</i>
花体	A – Z, a – z	打字机字体	A–Z,a–z,0–9

[7]：这里Unicode就是万国码。这里是指Unicode里支持的字体样式。关于它们的指令，请参考下面的“字体”一段。

Unicode

上面列出的字母可以在任何K^AT_EX模式下正确渲染。

此外，文本模式始终接受Brahmic，Georgian，中文，日文和韩文的字形。但是，这些字形将从系统字体（而不是K^AT_EX提供的字体）中呈现，因此它们的版式可能会发生冲突。您可以为CSS类^[8]`.latin-fallback`，`.cyrillic-fallback`，`.brahmic-fallback`，`.georgian-fallback`，`.cjk-fallback`和`.hangul-fallback`中提供规则，以为这些语言提供后备字体。这些字形的使用可能会引起小的垂直对齐问题：K^AT_EX列出了符号、大多数拉丁字母、希腊和西里尔字母的详细大小度量值，但其他接受的字形则被视为与当前K^AT_EX字体中的字母M一样高。（因此很多时候K^AT_EX用字母*M*作为参考大小——编者注）

如果K^AT_EX呈现模式的设置中，strict设置为了false或warn（默认），则K^AT_EX将在文本和数学模式下接受所有Unicode字母。所有无法识别的字符将被视为它们以文本模式显示。这样可能会遇到使用系统字体以及使用不正确的垂直对齐方式的问题。

对于波斯语中的复合字符，我们正在开发为用户提供的插件。（虽然你可能不会使用它——编者注）

[8]：因为没怎么写过CSS，所以可能说明上有一些问题。

布局

注解/标记

第一列	第二列
$\cancel{5}$	$\overbrace{a+b+c}^{\text{note}}$
$\bcancel{5}$	$\underbrace{a+b+c}_{\text{note}}$
\xcancel{ABC}	$\not =$
\sout{abc}	$\boxed{\pi = \frac{c}{d}}$
$\tag{hi} x+y^{2x}$	$x+y^{2x} \tag{hi}$
$\tag*{hi} x+y^{2x}$	$x+y^{2x} \tag{hi}$

换行指令

K^AT_EX 0.10.0版本之后将在关系或二元运算符（例如“=”或“+”）之后的内联数学中自动插入换行符。你可以通过`\nobreak`或将公式放在一对大括号中来阻止自动换行，例如`{F = ma}`。`\allowbreak`将允许在关系或运算符以外的其他位置自动换行。

硬换行符是`\\`和`\newline`。^[9]

在数学模式中，K^AT_EX不会自动插入换行符。但是当渲染选项`strict`设置为`true`时，它将忽略显示数学模式并且插入硬换行。

[9]：软换行换行不换段，硬换行换行也换段。简单来说，K^AT_EX有时会自动换行，但换行之后仍在同一段，也就是软换行。而硬换行之后，两部分内容就分离了。

垂直布局

第一列	第二列	第三列
x_n	$\stackrel{!}{=}$	$\frac{a}{b} a \atop b$
e^x	$\overset{!}{=}$	$a^b c \raisebox{0.25em}{b} c$

第一列	第二列	第三列
$\frac{o}{u} - u^o$	$\underset{!}{=}$	$\sum_{\substack{0 \leq i < m \\ 0 \leq j < n}} \sum_{\substack{0 \leq i < m \\ 0 \leq j < n}}$

如果`\raisebox`的第二个参数嵌套在`$. . . $`分隔符内，则可以包含数学运算符，当然前提是整个公式包括在`$$ $$`当中。例如`$$a\raisebox{0.25em}{$\frac{a}{b}$}c$$`的显示效果为

$$a\frac{a}{b}c$$

重叠和间距

第一列	第二列
\neq	$\mathllap{(\backslash,)} \quad (x^2) \left(x^{\smash{2}}\right)$
\neq	$\mathrlap{\backslash,/}{=} \quad \sqrt{y} \sqrt{\smash[b]{y}}$

$$\sum_{1 \leq i \leq j \leq n} x_{ij} \sum_{\mathclap{1 \leq i \leq j \leq n}} x_{ij}$$

K_AT_EX同样支持`\llap`^[10]，`\rlap`和`\clap`，但是它们仅能在文本模式中使用，而不是数学模式。

[10]：这里解释一下这个命令的用处：当我们使用巨运算符时，它的上标/下标可能过长，比如 $\sum_{1 \leq i \leq j \leq k \leq n} i \times j \times k$ ，这样`i*j*k`离运算符较远。但是当我们用了这些指令，就可以自动缩小这个距离。

$$\sum_{1 \leq i \leq j \leq k \leq n} i \times j \times k$$

间隔

指令	备注	指令	备注
<code>\,</code>	$\frac{3}{18}$ em ^[11] 的间距	<code>\kern{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\thinspace</code>	$\frac{3}{18}$ em 的间距	<code>\mkern{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\></code>	$\frac{4}{18}$ em 的间距	<code>\mskip{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\:</code>	$\frac{4}{18}$ em 的间距	<code>\hskip{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\medspace</code>	$\frac{4}{18}$ em 的间距	<code>\hspace{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\;</code>	$\frac{5}{18}$ em 间距	<code>\hspace*{distance}</code>	宽度为 <code>distance</code> 的间距
<code>\thickspace</code>	$\frac{5}{18}$ em 间距	<code></code>	长与高等于参数的间距
<code>\enspace</code>	$\frac{1}{2}$ em 间距	<code>\hphantom{content}</code>	高度等于参数的间距
<code>\quad</code>	1 em 间距	<code>\vphantom{content}</code>	高度等于参数的间距
<code>\qquad</code>	2 em 间距	<code>\!</code>	$-\frac{3}{18}$ em 间距

指令	备注	指令	备注
<code>~</code>	不换行间距	<code>\negthinspace</code>	$-\frac{3}{18}$ em 间距
<code>\<space></code>	间距	<code>\negmedspace</code>	$-\frac{4}{18}$ em 间距
<code>\nobreakspace</code>	不换行间距	<code>\negthickspace</code>	$-\frac{5}{18}$ em 间距
<code>\space</code>	普通的空格		

[11]：有关单位，请参考下方的“单位”一段。

提示

`distance`参数可以接受任何 \LaTeX 的单位。

`\kern`，`\mkern`，`\mskip`和`\hspace`都可以直接加上距离参数，例如，`\kern1em`可以生成一个宽度为1em的空格。

`\mkern`和`\mskip`在文本模式下不起作用，并且两者都将为`\mu`以外的任何单位显示警告。

逻辑与集合符号

第一列	第二列	第三列	第四列
\forall <code>\forall</code>	\complement <code>\complement</code>	\therefore <code>\therefore</code>	\emptyset <code>\emptyset</code>
\exists <code>\exists</code>	\subset <code>\subset</code>	\because <code>\because</code>	\emptyset <code>\empty</code>
\exists <code>\exist</code>	\supset <code>\supset</code>	\mapsto <code>\mapsto</code>	\varnothing <code>\varnothing</code>
\nexists <code>\nexists</code>	$ $ <code>\mid</code>	\rightarrow <code>\to</code>	\Rightarrow <code>\implies</code>
\in <code>\in</code>	\wedge <code>\land</code>	\leftarrow <code>\gets</code>	\Leftarrow <code>\impliedby</code>
\in <code>\isin</code>	\vee <code>\lor</code>	\leftrightarrow <code>\leftrightarrow</code>	\Leftrightarrow <code>\iff</code>
\notin <code>\notin</code>	\ni <code>\ni</code>	\nexists <code>\notni</code>	\neg <code>\neg</code> 或者 <code>\lnot</code>

你也可以直接输入这些：`\forall \therefore \complement \because \exists \mid \in \ni \subset \supset \wedge \vee \mapsto \rightarrow \leftarrow \leftrightarrow \neg \lnot\CHNPQR`

宏定义

显示效果	代码
$x^2 + x^2$	<code>\def\foo{x^2} \foo + \foo</code>
$y^2 + y^2$	<code>\gdef\bar#1{#1^2} \bar{y} + \bar{y}</code>
	<code>\global\def\名称#1#2...{定义}</code>

 $\backslash\text{newcommand}\backslash\text{名称}[\text{参数数量}]\{\text{定义}\}$

 $\backslash\text{renewcommand}\backslash\text{名称}[\text{参数数量}]\{\text{定义}\}$

 $\backslash\text{providecommand}\backslash\text{名称}[\text{参数数量}]\{\text{定义}\}$

你同样可以在 $\text{KAT}_{\text{E}}\text{X}$ 的渲染选项中定义宏。^[12]

宏最多接受九个参数： $\#1$ ， $\#2$ 等。

$\backslash\text{gdef}$ 和 $\backslash\text{global}\backslash\text{def}$ 宏将在数学表达式之间保留。

这些都是合法的指令名称：

 $\backslash\text{char}\backslash\text{mathchoice}\backslash\text{TextOrMath}\backslash\text{@ifstar}\backslash\text{@ifnextchar}\backslash\text{@firstoftwo}\backslash\text{@secondoftwo}\backslash\text{relax}$

@是命令的有效字符，就像你使用了 $\backslash\text{makeatletter}$ 指令。^[13]

[12]：宏定义类似于C++中的 $\backslash\text{define}$ ，可以简化代码，也可以嵌套。

可能有人不怎么清楚宏定义的具体用法，这里举个栗子：

假设我们有一段代码：

Hello world!!
Hello world!!
Hello world!!

```
$$
\color{red}{\text{Hello world!!}}\backslash\backslash
\color{red}{\text{Hello world!!}}\backslash\backslash
\color{red}{\text{Hello world!!}}\backslash\backslash
$$
```

但是这样非常麻烦，所以我们可以用宏定义简化一下。

 $\backslash\text{def}\backslash\text{HelloWorld}\backslash\color{red}\backslash\text{Hello world!!}\backslash\backslash\backslash\text{HelloWorld}\backslash\text{HelloWorld}\backslash\text{HelloWorld}$

```
$$
\def\HelloWorld{\color{red}{\text{Hello world!!}}\backslash\backslash}
\HelloWorld\HelloWorld\HelloWorld
$$
```

顺带解释一下一些细节。

- $\backslash\text{def}$ ， $\backslash\text{newcommand}$ 之类都可以直接加自定义指令的名称，而不需要加括号。比如 $\backslash\text{def}\backslash\text{kkksc03}\{\text{AK IOI!}\}$ 。

- `\gdef`, `\global\def`用处比较诡异，一般用不到。和`\def`的区别是，`\def`如果在一定嵌套中定义，只能用在有限的定义域内。而`\gdef`一类相当于直接在全局定义。可能有点模糊，这里举个例子：`{\def\t{2333} \t}` 这句话定义后的`\t`只能在括号中使用，比如`{\def\t{2333} \t}`。而括号外会失效。`{\def\t{2333}} \t`会渲染失败。而如果使用`\gdef`一类，就会作用于“全局”。`{\gdef\t{2333}} \t`成功渲染。
- 关于下面三个`command`的关系：`\newcommand`是定义一个新的指令，如果已经有同名指令，就会渲染失败。`\renewcommand`是重新定义指令。但是如果原来没有这个指令，也会渲染失败（迷惑行为）。`\providecommand`综合了上面两个指令的缺点，满足如果原来没有这个指令，就会定义一个；否则重新定义。（更迷惑了好不好）。
- 关于覆写的问题：正如上面提到的，`\def`与`\command`系列一样，只能作用于一定范围内，因此覆写的效果仅支持于这个范围。但是，如果将这些指令放在代码开头，就相当于全局定义了。`\gdef`范围更广。`\newcommand`不支持覆写；`\renewcommand`只支持覆写；`\providecommand`和`\def`都支持覆写。
- 关于多参数：首先， $\text{KAT}_{\text{E}}\text{X}$ 不支持不定参。然后，`\def`系列的参数可以这样定义：`\def{\qwq}\#1\#2\#3{\#1+\#2+\#3=?}`要强调的是，参数只能从`\#1`开始排列，并且递增。比如，你要9个参数，只能这样写：`\def{\名称}\#1\#2\#3\#4\#5\#6\#7\#8\#9{定义}`（非常迷惑）。而`command`系列相对方便，`\newcommand{\名称}[参数数量]{定义}`就可以了。
- (总结：遇事不决用`\def`，全局定义用`\def`，多参数还能用`\def`，`\def`天下第一)

[13]：解释一下`\makeatletter`指令：因为部分原因，我们在 $\text{T}_{\text{E}}\text{X}$ 中无法使用`@`作为指令名称，除非使用这个指令。而 $\text{KAT}_{\text{E}}\text{X}$ 中的`@`则不需要，所以说“就像你使用了`\makeatletter`指令”。

运算符

巨型运算符^[14]

第一列	第二列	第三列	第四列
Σ <code>\sum</code>	\prod <code>\prod</code>	\otimes <code>\bigotimes</code>	\vee <code>\bigvee</code>
\int <code>\int</code>	\coprod <code>\coprod</code>	\oplus <code>\bigoplus</code>	\wedge <code>\bigwedge</code>
\iint <code>\iint</code>	\int <code>\intop</code>	\odot <code>\bigodot</code>	\cap <code>\bigcap</code>
\iiint <code>\iiint</code>	\int <code>\smallint</code>	\uplus <code>\biguplus</code>	\cup <code>\bigcup</code>
\oint <code>\oint</code>	\oiint <code>\oiint</code>	\oiint <code>\oiint</code>	\sqcup <code>\bigsqcup</code>

当然，你也可以直接输入这些：`\int \iint \iiint \oint \oiint \oiint \sum \wedge \vee \cap \cup \odot \oplus \otimes \uplus \sqcup`

[14]：这里的巨型运算符在居中情况下自动变大，都支持上下标摆放在上下两侧。

二元运算符

第一列	第二列	第三列	第四列
$++$	$\cdot \quad \backslash cdot$	$\triangleright \quad \backslash gtrdot$	$x \pmod a \quad x \quad \backslash pmod a$
$--$	$\cdot \quad \backslash cdotp$	$\intercal \quad \backslash intercal$	$x(a) \quad x \quad \backslash pod a$
$//$	$\cdot \quad \backslash centerdot$	$\wedge \quad \backslash land$	$\triangleright \quad \backslash rhd$
$**$	$\circ \quad \backslash circ$	$\times \quad \backslash leftthreetimes$	$\ltimes \quad \backslash rightthreetimes$
$\amalg \quad \backslash amalg$	$\circledast \quad \backslash circledast$	$\ldotp \quad \backslash ldotp$	$\rtimes \quad \backslash rtimes$
$\& \quad \backslash And$	$\odot \quad \backslash circledcirc$	$\vee \quad \backslash lor$	$\setminus \quad \backslash setminus$
$\ast \quad \backslash ast$	$\ominus \quad \backslash circleddash$	$\lessdot \quad \backslash lessdot$	$\smallsetminus \quad \backslash smallsetminus$
$\bar{\wedge} \quad \backslash barwedge$	$\cup \quad \backslash Cup$	$\lhd \quad \backslash lhd$	$\sqcap \quad \backslash sqcap$
$\bigcirc \quad \backslash bigcirc$	$\cup \quad \backslash cup$	$\ltimes \quad \backslash ltimes$	$\sqcup \quad \backslash sqcup$
$\bmod \quad \backslash bmod$	$\vee \quad \backslash curlyvee$	$x \bmod a \quad x \backslash mod a$	$\times \quad \backslash times$
$\boxed{\cdot} \quad \backslash boxdot$	$\curlywedge \quad \backslash curlywedge$	$\mp \quad \backslash mp$	$\unlhd \quad \backslash unlhd$
$\boxminus \quad \backslash boxminus$	$\div \quad \backslash div$	$\odot \quad \backslash odot$	$\unrhd \quad \backslash unrhd$
$\boxplus \quad \backslash boxplus$	$\div \quad \backslash divideontimes$	$\ominus \quad \backslash ominus$	$\uplus \quad \backslash uplus$
$\boxtimes \quad \backslash boxtimes$	$\dot{+} \quad \backslash dotplus$	$\oplus \quad \backslash oplus$	$\vee \quad \backslash vee$
$\bullet \quad \backslash bullet$	$\bar{\wedge} \quad \backslash doublebarwedge$	$\otimes \quad \backslash otimes$	$\veebar \quad \backslash veebar$
$\Cap \quad \backslash Cap$	$\Cap \quad \backslash doublecap$	$\oslash \quad \backslash oslash$	$\wedge \quad \backslash wedge$
$\cap \quad \backslash cap$	$\cup \quad \backslash doublecup$	$\pm \quad \backslash pm \text{ 或者 } \backslash plusmn$	$\wr \quad \backslash wr$

当然，你也可以直接输入这些： $+-/*\cdot \pm\times\div\mp\dot{+}\wedge\vee\cap\cup\wr\cup\cap\oplus\ominus\otimes\oslash\odot\odot\odot\odot$

分数和二项式

第一列	第二列	第三列
$\frac{a}{b} \quad \backslash frac{a}{b}$	$\frac{a}{b} \quad \backslash tfrac{a}{b}$ [15]	$(\frac{a}{a+1}) \quad \backslash genfrac {}{1}{2pt} a{a+1}$
$\frac{a}{b} \{a \over b\}$	$\frac{a}{b} \quad \backslash dfrac{a}{b}$	$\frac{a}{b+1} \{a \above{2pt} b+1\}$
$\frac{a}{1+\frac{1}{b}} \quad \backslash cfrac{a}{1+\backslash cfrac{1}{b}}$		
$\binom{n}{k} \quad \backslash binom{n}{k}$	$\dbinom{n}{k} \quad \backslash dbinom{n}{k}$	$\{n\brace k \}$

第一列	第二列	第三列
$\binom{n}{k}$ <code>{n \choose k}</code>	$\binom{n}{k}$ <code>\tbinom{n}{k}</code>	$\left[n\right]_k$ <code>{n\brack k}</code>

[15]：在洛谷讨论区的K^AT_EX版本中，有个Bug：嵌套使用`\tfrac{ }{ }`会导致字体膨胀。

数学运算符

第一列	第二列	第三列	第四列
\arcsin <code>\arcsin</code>	\cotg <code>\cotg</code>	\ln <code>\ln</code>	\det <code>\det</code>
\arccos <code>\arccos</code>	\coth <code>\coth</code>	\log <code>\log</code>	\gcd <code>\gcd</code>
\arctan <code>\arctan</code>	\csc <code>\csc</code>	\sec <code>\sec</code>	\inf <code>\inf</code>
\arctg <code>\arctg</code>	\ctg <code>\ctg</code>	\sin <code>\sin</code>	\lim <code>\lim</code>
\arcctg <code>\arcctg</code>	\cth <code>\cth</code>	\sinh <code>\sinh</code>	\liminf <code>\liminf</code>
\arg <code>\arg</code>	\deg <code>\deg</code>	\sh <code>\sh</code>	\limsup <code>\limsup</code>
\ch <code>\ch</code>	\dim <code>\dim</code>	\tan <code>\tan</code>	\max <code>\max</code>
\cos <code>\cos</code>	\exp <code>\exp</code>	\tanh <code>\tanh</code>	\min <code>\min</code>
\cosec <code>\cosec</code>	\hom <code>\hom</code>	\tg <code>\tg</code>	\Pr <code>\Pr</code>
\cosh <code>\cosh</code>	\ker <code>\ker</code>	\th <code>\th</code>	\sup <code>\sup</code>
\cot <code>\cot</code>	\lg <code>\lg</code>	\argmax <code>\argmax</code>	\argmin <code>\argmin</code>
\plim <code>\plim</code>	f <code>\operatorname{f}</code>	f <code>\operatorname{*f}</code>	

参数显示在右侧的公式，如 $\sum_{i=1}^n$ 可以使用`\limits`强行显示在上下侧， \sum ，`\sum\limits_{i=1}^n`

开根

\sqrt{x} `\sqrt{x}`

$\sqrt[3]{x}$ `\sqrt[3]{x}`

关系指令

$\stackrel{!}{=}$ `\stackrel{!}{=}`

第一列	第二列	第三列	第四列
$=$	\equiv <code>\eqcirc</code>	\lesseqgtr <code>\lesseqgtr</code>	\sqsupset <code>\sqsupset</code>

第一列	第二列	第三列	第四列
$<<$	\colon \eqcolon	\lesseqgtr	\sqsupseteq
$>>$	\Eqcolon	\lessgtr	\Subset
$::$	\eqqcolon	\lesssim	\subset 或者 \sub
\approx	\Eqqcolon	\ll	\subseteq 或者 \sube
\approxeq	\eqsim	\lll	\subseteqq
\asymp	\eqslantgtr	\llless	\succ
\backepsilon	\eqslantless	\lt	\succapprox
\backsimeq	\equiv	\mid	\succcurlyeq
\backsim	\fallingdotseq	\models	\succeq
\between	\frown	\multimap	\succsim
\bowtie	\geq	\owns	\supset
\bumpeq	\geq	\parallel	\supseteq
\Bumpeq	\geqq	\perp	\supseteq 或者 \supe
\circeq	\geqslant	\pitchfork	\supseteqq
\colonapprox	\gg	\prec	\thickapprox
\Colonapprox	\ggg	\precapprox	\thicksim
\coloneq	\gggtr	\preccurlyeq	\triangleleft
\Coloneq	\gt	\preceq	\triangleq
\coloneqq	\gtrapprox	\precsim	\triangleright
\Coloneqq	\gtreqless	\propto	\varpropto
\colonsim	\gtreqqless	\risingdotseq	\vartriangle
\ColonSim	\gtrless	\shortmid	\vartriangleleft
\cong	\gtrsim	\shortparallel	\vartriangleright
\curlyeqprec	\in 或者 \isin	\sim	\vcentcolon
\curlyeqsucc	\Join	\simeq	\vdash
\dashv	\leq	\smallfrown	\Vdash
\dblcolon	\leq	\smallsmile	\Vdash
\doteq	\leqq	\smile	\Vvdash
\Doteq	\leqslant	\sqsubset	
\doteqdot	\lessapprox	\sqsubseteq	

当然，你也可以直接输入这些：

⤵⤴⤶⤷⤸⤹⤺⤻⤼⤽⤾⤿⥀⥁⥂⥃⥄⥅⥆⥇⥈⥉⥊⥋⥌⥍⥎⥏⥐⥑⥒⥓⥔⥕⥖⥗⥘⥙⥚⥛⥜⥝⥞⥟⥠⥡⥢⥣⥤⥥⥦⥧⥨⥩⥪⥫⥬⥭⥮⥯⥰⥱⥲⥳⥴⥵⥶⥷⥸⥹⥺⥻⥼⥽⥾⥿⦀⦁⦂⦃⦄⦅⦆⦇⦈⦉⦊⦋⦌⦍⦎⦏⦐⦑⦒⦓⦔⦕⦖⦗⦘⦙⦚⦛⦜⦝⦞⦟⦠⦡⦢⦣⦤⦥⦦⦧⦨⦩⦪⦫⦬⦭⦮⦯⦰⦱⦲⦳⦴⦵⦶⦷⦸⦹⦺⦻⦼⦽⦾⦿⧀⧁⧂⧃⧄⧅⧆⧇⧈⧉⧊⧋⧌⧍⧎⧏⧐⧑⧒⧓⧔⧕⧖⧗⧘⧙⧚⧛⧜⧝⧞⧟⧠⧡⧢⧣⧤⧥⧦⧧⧨⧩⧪⧫⧬⧭⧮⧯⧰⧱⧲⧳⧴⧵⧶⧷⧸⧹⧺⧻⧼⧽⧾⧿⨀⨁⨂⨃⨄⨅⨆⨇⨈⨉⨊⨋⨌⨍⨎⨏⨐⨑⨒⨓⨔⨕⨖⨗⨘⨙⨚⨛⨜⨝⨞⨟⨠⨡⨢⨣⨤⨥⨦⨧⨨⨩⨪⨫⨬⨭⨮⨯⨰⨱⨲⨳⨴⨵⨶⨷⨸⨹⨺⨻⨼⨽⨾⨿⩀⩁⩂⩃⩄⩅⩆⩇⩈⩉⩊⩋⩌⩍⩎⩏⩐⩑⩒⩓⩔⩕⩖⩗⩘⩙⩚⩛⩜⩝⩞⩟⩠⩡⩢⩣⩤⩥⩦⩧⩨⩩⩪⩫⩬⩭⩮⩯⩰⩱⩲⩳⩴⩵⩶⩷⩸⩹⩺⩻⩼⩽⩾⩿⪀⪁⪂⪃⪄⪅⪆⪇⪈⪉⪊⪋⪌⪍⪎⪏⪐⪑⪒⪓⪔⪕⪖⪗⪘⪙⪚⪛⪜⪝⪞⪟⪠⪡⪢⪣⪤⪥⪦⪧⪨⪩⪪⪫⪬⪭⪮⪯⪰⪱⪲⪳⪴⪵⪶⪷⪸⪹⪺⪻⪼⪽⪾⪿⫀⫁⫂⫃⫄⫅⫆⫇⫈⫉⫊⫋⫌⫍⫎⫏⫐⫑⫒⫓⫔⫕⫖⫗⫘⫙⫚⫛⫝̸⫝⫞⫟⫠⫡⫢⫣⫤⫥⫦⫧⫨⫩⫪⫫⫬⫭⫮⫯⫰⫱⫲⫳⫴⫵⫶⫷⫸⫹⫺⫻⫼⫽⫾⫿⬀⬁⬂⬃⬄⬅⬆⬇⬈⬉⬊⬋⬌⬍⬎⬏⬐⬑⬒⬓⬔⬕⬖⬗⬘⬙⬚⬛⬜⬝⬞⬟⬠⬡⬢⬣⬤⬥⬦⬧⬨⬩⬪⬫⬬⬭⬮⬯⬰⬱⬲⬳⬴⬵⬶⬷⬸⬹⬺⬻⬼⬽⬾⬿⭀⭁⭂⭃⭄⭅⭆⭇⭈⭉⭊⭋⭌⭍⭎⭏⭐⭑⭒⭓⭔⭕⭖⭗⭘⭙⭚⭛⭜⭝⭞⭟⭠⭡⭢⭣⭤⭥⭦⭧⭨⭩⭪⭫⭬⭭⭮⭯⭰⭱⭲⭳⭴⭵⭶⭷⭸⭹⭺⭻⭼⭽⭾⭿⮀⮁⮂⮃⮄⮅⮆⮇⮈⮉⮊⮋⮌⮍⮎⮏⮐⮑⮒⮓⮔⮕⮖⮗⮘⮙⮚⮛⮜⮝⮞⮟⮠⮡⮢⮣⮤⮥⮦⮧⮨⮩⮪⮫⮬⮭⮮⮯⮰⮱⮲⮳⮴⮵⮶⮷⮸⮹⮺⮻⮼⮽⮾⮿⯀⯁⯂⯃⯄⯅⯆⯇⯈⯉⯊⯋⯌⯍⯎⯏⯐⯑⯒⯓⯔⯕⯖⯗⯘⯙⯚⯛⯜⯝⯞⯟⯠⯡⯢⯣⯤⯥⯦⯧⯨⯩⯪⯫⯬⯭⯮⯯⯰⯱⯲⯳⯴⯵⯶⯷⯸⯹⯺⯻⯼⯽⯾⯿ⰀⰁⰂⰃⰄⰅⰆⰇⰈⰉⰊⰋⰌⰍⰎⰏⰐⰑⰒⰓⰔⰕⰖⰗⰘⰙⰚⰛⰜⰝⰞⰟⰠⰡⰢⰣⰤⰥⰦⰧⰨⰩⰪⰫⰬⰭⰮⰯⰰⰱⰲⰳⰴⰵⰶⰷⰸⰹⰺⰻⰼⰽⰾⰿⱀⱁⱂⱃⱄⱅⱆⱇⱈⱉⱊⱋⱌⱍⱎⱏⱐⱑⱒⱓⱔⱕⱖⱗⱘⱙⱚⱛⱜⱝⱞⱟⱠⱡⱢⱣⱤⱥⱦⱧⱨⱩⱪⱫⱬⱭⱮⱯⱰⱱⱲⱳⱴⱵⱶⱷⱸⱹⱺⱻⱼⱽⱾⱿⲀⲁⲂⲃⲄⲅⲆⲇⲈⲉⲊⲋⲌⲍⲎⲏⲐⲑⲒⲓⲔⲕⲖⲗⲘⲙⲚⲛⲜⲝⲞⲟⲠⲡⲢⲣⲤⲥⲦⲧⲨⲩⲪⲫⲬⲭⲮⲯⲰⲱⲲⲳⲴⲵⲶⲷⲸⲹⲺⲻⲼⲽⲾⲿⳀⳁⳂⳃⳄⳅⳆⳇⳈⳉⳊⳋⳌⳍⳎⳏⳐⳑⳒⳓⳔⳕⳖⳗⳘⳙⳚⳛⳜⳝⳞⳟⳠⳡⳢⳣⳤ⳥⳦⳧⳨⳩⳪ⳫⳬⳭⳮ⳯⳰⳱Ⳳⳳ⳴⳵⳶⳷⳸⳹⳺⳻⳼⳽⳾⳿ⴀⴁⴂⴃⴄⴅⴆⴇⴈⴉⴊⴋⴌⴍⴎⴏⴐⴑⴒⴓⴔⴕⴖⴗⴘⴙⴚⴛⴜⴝⴞⴟⴠⴡⴢⴣⴤⴥ⴦ⴧ⴨⴩⴪⴫⴬ⴭ⴮⴯ⴰⴱⴲⴳⴴⴵⴶⴷⴸⴹⴺⴻⴼⴽⴾⴿⵀⵁⵂⵃⵄⵅⵆⵇⵈⵉⵊⵋⵌⵍⵎⵏⵐⵑⵒⵓⵔⵕⵖⵗⵘⵙⵚⵛⵜⵝⵞⵟⵠⵡⵢⵣⵤⵥⵦⵧ⵨⵩⵪⵫⵬⵭⵮ⵯ⵰⵱⵲⵳⵴⵵⵶⵷⵸⵹⵺⵻⵼⵽⵾⵿ⶀⶁⶂⶃⶄⶅⶆⶇⶈⶉⶊⶋⶌⶍⶎⶏⶐⶑⶒⶓⶔⶕⶖ⶗⶘⶙⶚⶛⶜⶝⶞⶟ⶠⶡⶢⶣⶤⶥⶦ⶧ⶨⶩⶪⶫⶬⶭⶮ⶯ⶰⶱⶲⶳⶴⶵⶶ⶷ⶸⶹⶺⶻⶼⶽⶾ⶿ⷀⷁⷂⷃⷄⷅⷆ⷇ⷈⷉⷊⷋⷌⷍⷎ⷏ⷐⷑⷒⷓⷔⷕⷖ⷗ⷘⷙⷚⷛⷜⷝⷞ⷟ⷠⷡⷢⷣⷤⷥⷦⷧⷨⷩⷪⷫⷬⷭⷮⷯⷰⷱⷲⷳⷴⷵⷶⷷⷸⷹⷺⷻⷼⷽⷾⷿ⸀⸁⸂⸃⸄⸅⸆⸇⸈⸉⸊⸋⸌⸍⸎⸏⸐⸑⸒⸓⸔⸕⸖⸗⸘⸙⸚⸛⸜⸝⸞⸟⸠⸡⸢⸣⸤⸥⸦⸧⸨⸩⸪⸫⸬⸭⸮ⸯ⸰⸱⸲⸳⸴⸵⸶⸷⸸⸹⸺⸻⸼⸽⸾⸿⹀⹁⹂⹃⹄⹅⹆⹇⹈⹉⹊⹋⹌⹍⹎⹏⹐⹑⹒⹓⹔⹕⹖⹗⹘⹙⹚⹛⹜⹝⹞⹟⹠⹡⹢⹣⹤⹥⹦⹧⹨⹩⹪⹫⹬⹭⹮⹯⹰⹱⹲⹳⹴⹵⹶⹷⹸⹹⹺⹻⹼⹽⹾⹿⺀⺁⺂⺃⺄⺅⺆⺇⺈⺉⺊⺋⺌⺍⺎⺏⺐⺑⺒⺓⺔⺕⺖⺗⺘⺙⺚⺛⺜⺝⺞⺟⺠⺡⺢⺣⺤⺥⺦⺧⺨⺩⺪⺫⺬⺭⺮⺯⺰⺱⺲⺳⺴⺵⺶⺷⺸⺹⺺⺻⺼⺽⺾⺿⻀⻁⻂⻃⻄⻅⻆⻇⻈⻉⻊⻋⻌⻍⻎⻏⻐⻑⻒⻓⻔⻕⻖⻗⻘⻙⻚⻛⻜⻝⻞⻟⻠⻡⻢⻣⻤⻥⻦⻧⻨⻩⻪⻫⻬⻭⻮⻯⻰⻱⻲⻳⻴⻵⻶⻷⻸⻹⻺⻻⻼⻽⻾⻿⼀⼁⼂⼃⼄⼅⼆⼇⼈⼉⼊⼋⼌⼍⼎⼏⼐⼑⼒⼓⼔⼕⼖⼗⼘⼙⼚⼛⼜⼝⼞⼟⼠⼡⼢⼣⼤⼥⼦⼧⼨⼩⼪⼫⼬⼭⼮⼯⼰⼱⼲⼳⼴⼵⼶⼷⼸⼹⼺⼻⼼⼽⼾⼿⽀⽁⽂⽃⽄⽅⽆⽇⽈⽉⽊⽋⽌⽍⽎⽏⽐⽑⽒⽓⽔⽕⽖⽗⽘⽙⽚⽛⽜⽝⽞⽟⽠⽡⽢⽣⽤⽥⽦⽧⽨⽩⽪⽫⽬⽭⽮⽯⽰⽱⽲⽳⽴⽵⽶⽷⽸⽹⽺⽻⽼⽽⽾⽿⿀⿁⿂⿃⿄⿅⿆⿇⿈⿉⿊⿋⿌⿍⿎⿏⿐⿑⿒⿓⿔⿕⿖⿗⿘⿙⿚⿛⿜⿝⿞⿟⿠⿡⿢⿣⿤⿥⿦⿧⿨⿩⿪⿫⿬⿭⿮⿯⿰⿱⿲⿳⿴⿵⿶⿷⿸⿹⿺⿻⿼⿽⿾⿿⸀⸁⸂⸃⸄⸅⸆⸇⸈⸉⸊⸋⸌⸍⸎⸏⸐⸑⸒⸓⸔⸕⸖⸗⸘⸙⸚⸛⸜⸝⸞⸟⸠⸡⸢⸣⸤⸥⸦⸧⸨⸩⸪⸫⸬⸭⸮ⸯ⸰⸱⸲⸳⸴⸵⸶⸷⸸⸹⸺⸻⸼⸽⸾⸿⹀⹁⹂⹃⹄⹅⹆⹇⹈⹉⹊⹋⹌⹍⹎⹏⹐⹑⹒⹓⹔⹕⹖⹗⹘⹙⹚⹛⹜⹝⹞⹟⹠⹡⹢⹣⹤⹥⹦⹧⹨⹩⹪⹫⹬⹭⹮⹯⹰⹱⹲⹳⹴⹵⹶⹷⹸⹹⹺⹻⹼⹽⹾⹿⺀⺁⺂⺃⺄⺅⺆⺇⺈⺉⺊⺋⺌⺍⺎⺏⺐⺑⺒⺓⺔⺕⺖⺗⺘⺙⺚⺛⺜⺝⺞⺟⺠⺡⺢⺣⺤⺥⺦⺧⺨⺩⺪⺫⺬⺭⺮⺯⺰⺱⺲⺳⺴⺵⺶⺷⺸⺹⺺⺻⺼⺽⺾⺿⻀⻁⻂⻃⻄⻅⻆⻇⻈⻉⻊⻋⻌⻍⻎⻏⻐⻑⻒⻓⻔⻕⻖⻗⻘⻙⻚⻛⻜⻝⻞⻟⻠⻡⻢⻣⻤⻥⻦⻧⻨⻩⻪⻫⻬⻭⻮⻯⻰⻱⻲⻳⻴⻵⻶⻷⻸⻹⻺⻻⻼⻽⻾⻿⼀⼁⼂⼃⼄⼅⼆⼇⼈⼉⼊⼋⼌⼍⼎⼏⼐⼑⼒⼓⼔⼕⼖⼗⼘⼙⼚⼛⼜⼝⼞⼟⼠⼡⼢⼣⼤⼥⼦⼧⼨⼩⼪⼫⼬⼭⼮⼯⼰⼱⼲⼳⼴⼵⼶⼷⼸⼹⼺⼻⼼⼽⼾⼿⽀⽁⽂⽃⽄⽅⽆⽇⽈⽉⽊⽋⽌⽍⽎⽏⽐⽑⽒⽓⽔⽕⽖⽗⽘⽙⽚⽛⽜⽝⽞⽟⽠⽡⽢⽣⽤⽥⽦⽧⽨⽩⽪⽫⽬⽭⽮⽯⽰⽱⽲⽳⽴⽵⽶⽷⽸⽹⽺⽻⽼⽽⽾⽿⿀⿁⿂⿃⿄⿅⿆⿇⿈⿉⿊⿋⿌⿍⿎⿏⿐⿑⿒⿓⿔⿕⿖⿗⿘⿙⿚⿛⿜⿝⿞⿟⿠⿡⿢⿣⿤⿥⿦⿧⿨⿩⿪⿫⿬⿭⿮⿯⿰⿱⿲⿳⿴⿵⿶⿷⿸⿹⿺⿻⿼⿽⿾⿿

箭头

第一列	第二列	第三列
\circlearrowleft	\leftharpoonup	\Rightarrow
\circlearrowright	\leftleftarrows	\rightarrow
\curvearrowleft	\leftrightarrow	\restriction
\curvearrowright	\Leftrightarrow	\rightarrow
\Downarrow	\leftrightsquigarrow	\Rightarrow
\Downarrow	\leftrightharpoons	\rightarrowtail
\downarrow	\leftrightsquigarrow	\rightarrowtail
\dashleftarrow	\Lleftarrow	\rightarrowtail
\dashrightarrow	\longrightarrow	\rightleftarrows
\downarrow	\Longleftarrow	\rightleftharpoons
\Downarrow	\longleftrightarrow	\Rightarrow
\Downarrow	\Longleftrightarrow	\rightsquigarrow
\Downarrow	\longmapsto	\Rightarrow
\Downarrow	\longrightarrow	\Rsh
\gets	\Longrightarrow	\searrow
\Leftrightarrow	\looparrowleft	\swarrow
\Leftrightarrow	\looparrowright	\rightarrow
\leftrightarrow	\Lrarr	\twoheadleftarrow
\hookleftarrow	\Lrarr	\twoheadrightarrow
\hookrightarrow	\Lrarr	\Uarr
\iff	\Lsh	\Uarr
\impliedby	\mapsto	\Uarr
\implies	\nearrow	\Uparrow
\Larr	\nleftarrow	\Uparrow
\LArr	\nLeftarrow	\updownarrow
\larr	\nleftrightarrow	\Updownarrow

第一列	第二列	第三列
\leadsto <code>\leadsto</code>	\Leftrightarrow <code>\nLeftrightarrow</code>	\Uparrow <code>\upharpoonleft</code>
\leftarrow <code>\leftarrow</code>	\rightarrow <code>\rightarrow</code>	\Uparrow <code>\upharpoonright</code>
\Leftarrow <code>\Leftarrow</code>	\Rightarrow <code>\Rightarrow</code>	\Uparrow <code>\upuparrows</code>
\leftarrowtail <code>\leftarrowtail</code>	\nwarrow <code>\nwarrow</code>	
\leftharpoonup <code>\leftharpoonup</code>	\Rightarrow <code>\Rrightarrow</code>	

你也可以直接输入这些：←↑→↓↕↗↘↙↞↠↡↢↣↤↥↦↧↨↩↪↫↬↭↮↯↰↱↲↳↴↵↶↷↸↹↺↻↼↽↾↿⇀⇁⇂⇃⇄⇅⇆⇇⇈⇉⇊⇋⇌⇍⇎⇏⇐⇑⇒⇓⇔⇕⇖⇗⇘⇙⇚⇛⇜⇝⇞⇟⇠⇡⇢⇣⇤⇥⇦⇧⇨⇩⇪⇫⇬⇭⇮⇯⇰⇱⇲⇳⇴⇵⇶⇷⇸⇹⇺⇻⇼⇽⇾⇿⤀⤁⤂⤃⤄⤅⤆⤇⤈⤉⤊⤋⤌⤍⤎⤏⤐⤑⤒⤓⤔⤕⤖⤗⤘⤙⤚⤛⤜⤝⤞⤟⤠⤡⤢⤣⤤⤥⤦⤧⤨⤩⤪⤫⤬⤭⤮⤯⤰⤱⤲⤳⤴⤵⤶⤷⤸⤹⤺⤻⤼⤽⤾⤿⥀⥁⥂⥃⥄⥅⥆⥇⥈⥉⥊⥋⥌⥍⥎⥏⥐⥑⥒⥓⥔⥕⥖⥗⥘⥙⥚⥛⥜⥝⥞⥟⥠⥡⥢⥣⥤⥥⥦⥧⥨⥩⥪⥫⥬⥭⥮⥯⥰⥱⥲⥳⥴⥵⥶⥷⥸⥹⥺⥻⥼⥽⥾⥿⦀⦁⦂⦃⦄⦅⦆⦇⦈⦉⦊⦋⦌⦍⦎⦏⦐⦑⦒⦓⦔⦕⦖⦗⦘⦙⦚⦛⦜⦝⦞⦟⦠⦡⦢⦣⦤⦥⦦⦧⦨⦩⦪⦫⦬⦭⦮⦯⦰⦱⦲⦳⦴⦵⦶⦷⦸⦹⦺⦻⦼⦽⦾⦿⧀⧁⧂⧃⧄⧅⧆⧇⧈⧉⧊⧋⧌⧍⧎⧏⧐⧑⧒⧓⧔⧕⧖⧗⧘⧙⧚⧛⧜⧝⧞⧟⧠⧡⧢⧣⧤⧥⧦⧧⧨⧩⧪⧫⧬⧭⧮⧯⧰⧱⧲⧳⧴⧵⧶⧷⧸⧹⧺⧻⧼⧽⧾⧿⨀⨁⨂⨃⨄⨅⨆⨇⨈⨉⨊⨋⨌⨍⨎⨏⨐⨑⨒⨓⨔⨕⨖⨗⨘⨙⨚⨛⨜⨝⨞⨟⨠⨡⨢⨣⨤⨥⨦⨧⨨⨩⨪⨫⨬⨭⨮⨯⨰⨱⨲⨳⨴⨵⨶⨷⨸⨹⨺⨻⨼⨽⨾⨿⩀⩁⩂⩃⩄⩅⩆⩇⩈⩉⩊⩋⩌⩍⩎⩏⩐⩑⩒⩓⩔⩕⩖⩗⩘⩙⩚⩛⩜⩝⩞⩟⩠⩡⩢⩣⩤⩥⩦⩧⩨⩩⩪⩫⩬⩭⩮⩯⩰⩱⩲⩳⩴⩵⩶⩷⩸⩹⩺⩻⩼⩽⩾⩿⪀⪁⪂⪃⪄⪅⪆⪇⪈⪉⪊⪋⪌⪍⪎⪏⪐⪑⪒⪓⪔⪕⪖⪗⪘⪙⪚⪛⪜⪝⪞⪟⪠⪡⪢⪣⪤⪥⪦⪧⪨⪩⪪⪫⪬⪭⪮⪯⪰⪱⪲⪳⪴⪵⪶⪷⪸⪹⪺⪻⪼⪽⪾⪿⫀⫁⫂⫃⫄⫅⫆⫇⫈⫉⫊⫋⫌⫍⫎⫏⫐⫑⫒⫓⫔⫕⫖⫗⫘⫙⫚⫛⫝̸⫝⫞⫟⫠⫡⫢⫣⫤⫥⫦⫧⫨⫩⫪⫫⫬⫭⫮⫯⫰⫱⫲⫳⫴⫵⫶⫷⫸⫹⫺⫻⫼⫽⫾⫿⬀⬁⬂⬃⬄⬅⬆⬇⬈⬉⬊⬋⬌⬍⬎⬏⬐⬑⬒⬓⬔⬕⬖⬗⬘⬙⬚⬛⬜⬝⬞⬟⬠⬡⬢⬣⬤⬥⬦⬧⬨⬩⬪⬫⬬⬭⬮⬯⬰⬱⬲⬳⬴⬵⬶⬷⬸⬹⬺⬻⬼⬽⬾⬿⭀⭁⭂⭃⭄⭅⭆⭇⭈⭉⭊⭋⭌⭍⭎⭏⭐⭑⭒⭓⭔⭕⭖⭗⭘⭙⭚⭛⭜⭝⭞⭟⭠⭡⭢⭣⭤⭥⭦⭧⭨⭩⭪⭫⭬⭭⭮⭯⭰⭱⭲⭳⭴⭵⭶⭷⭸⭹⭺⭻⭼⭽⭾⭿⮀⮁⮂⮃⮄⮅⮆⮇⮈⮉⮊⮋⮌⮍⮎⮏⮐⮑⮒⮓⮔⮕⮖⮗⮘⮙⮚⮛⮜⮝⮞⮟⮠⮡⮢⮣⮤⮥⮦⮧⮨⮩⮪⮫⮬⮭⮮⮯⮰⮱⮲⮳⮴⮵⮶⮷⮸⮹⮺⮻⮼⮽⮾⮿⯀⯁⯂⯃⯄⯅⯆⯇⯈⯉⯊⯋⯌⯍⯎⯏⯐⯑⯒⯓⯔⯕⯖⯗⯘⯙⯚⯛⯜⯝⯞⯟⯠⯡⯢⯣⯤⯥⯦⯧⯨⯩⯪⯫⯬⯭⯮⯯⯰⯱⯲⯳⯴⯵⯶⯷⯸⯹⯺⯻⯼⯽⯾⯿ⰀⰁⰂⰃⰄⰅⰆⰇⰈⰉⰊⰋⰌⰍⰎⰏⰐⰑⰒⰓⰔⰕⰖⰗⰘⰙⰚⰛⰜⰝⰞⰟⰠⰡⰢⰣⰤⰥⰦⰧⰨⰩⰪⰫⰬⰭⰮⰯⰰⰱⰲⰳⰴⰵⰶⰷⰸⰹⰺⰻⰼⰽⰾⰿⱀⱁⱂⱃⱄⱅⱆⱇⱈⱉⱊⱋⱌⱍⱎⱏⱐⱑⱒⱓⱔⱕⱖⱗⱘⱙⱚⱛⱜⱝⱞⱟⱠⱡⱢⱣⱤⱥⱦⱧⱨⱩⱪⱫⱬⱭⱮⱯⱰⱱⱲⱳⱴⱵⱶⱷⱸⱹⱺⱻⱼⱽⱾⱿⲀⲁⲂⲃⲄⲅⲆⲇⲈⲉⲊⲋⲌⲍⲎⲏⲐⲑⲒⲓⲔⲕⲖⲗⲘⲙⲚⲛⲜⲝⲞⲟⲠⲡⲢⲣⲤⲥⲦⲧⲨⲩⲪⲫⲬⲭⲮⲯⲰⲱⲲⲳⲴⲵⲶⲷⲸⲹⲺⲻⲼⲽⲾⲿⳀⳁⳂⳃⳄⳅⳆⳇⳈⳉⳊⳋⳌⳍⳎⳏⳐⳑⳒⳓⳔⳕⳖⳗⳘⳙⳚⳛⳜⳝⳞⳟⳠⳡⳢⳣⳤ⳥⳦⳧⳨⳩⳪ⳫⳬⳭⳮ⳯⳰⳱Ⳳⳳ⳴⳵⳶⳷⳸⳹⳺⳻⳼⳽⳾⳿ⴀⴁⴂⴃⴄⴅⴆⴇⴈⴉⴊⴋⴌⴍⴎⴏⴐⴑⴒⴓⴔⴕⴖⴗⴘⴙⴚⴛⴜⴝⴞⴟⴠⴡⴢⴣⴤⴥ⴦ⴧ⴨⴩⴪⴫⴬ⴭ⴮⴯ⴰⴱⴲⴳⴴⴵⴶⴷⴸⴹⴺⴻⴼⴽⴾⴿⵀⵁⵂⵃⵄⵅⵆⵇⵈⵉⵊⵋⵌⵍⵎⵏⵐⵑⵒⵓⵔⵕⵖⵗⵘⵙⵚⵛⵜⵝⵞⵟⵠⵡⵢⵣⵤⵥⵦⵧ⵨⵩⵪⵫⵬⵭⵮ⵯ⵰⵱⵲⵳⵴⵵⵶⵷⵸⵹⵺⵻⵼⵽⵾⵿ⶀⶁⶂⶃⶄⶅⶆⶇⶈⶉⶊⶋⶌⶍⶎⶏⶐⶑⶒⶓⶔⶕⶖ⶗⶘⶙⶚⶛⶜⶝⶞⶟ⶠⶡⶢⶣⶤⶥⶦ⶧ⶨⶩⶪⶫⶬⶭⶮ⶯ⶰⶱⶲⶳⶴⶵⶶ⶷ⶸⶹⶺⶻⶼⶽⶾ⶿ⷀⷁⷂⷃⷄⷅⷆ⷇ⷈⷉⷊⷋⷌⷍⷎ⷏ⷐⷑⷒⷓⷔⷕⷖ⷗ⷘⷙⷚⷛⷜⷝⷞ⷟ⷠⷡⷢⷣⷤⷥⷦⷧⷨⷩⷪⷫⷬⷭⷮⷯⷰⷱⷲⷳⷴⷵⷶⷷⷸⷹⷺⷻⷼⷽⷾⷿ⸀⸁⸂⸃⸄⸅⸆⸇⸈⸉⸊⸋⸌⸍⸎

箭头扩展^[16]

第一列	第二列
\xleftarrow{abc} <code>\xleftarrow{abc}</code>	$\xrightarrow[under]{over}$ <code>\xrightarrow[under]{over}</code>
\xLeftrightarrow{abc} <code>\xLeftrightarrow{abc}</code>	\xRightarrow{abc} <code>\xRightarrow{abc}</code>
\xleftrightarrow{abc} <code>\xleftrightarrow{abc}</code>	\xLeftrightarrow{abc} <code>\xLeftrightarrow{abc}</code>
\xhookleftarrow{abc} <code>\xhookleftarrow{abc}</code>	\xhookrightarrow{abc} <code>\xhookrightarrow{abc}</code>
\xtwoheadleftarrow{abc} <code>\xtwoheadleftarrow{abc}</code>	\xtwoheadrightarrow{abc} <code>\xtwoheadrightarrow{abc}</code>
\xleftharpoonup{abc} <code>\xleftharpoonup{abc}</code>	\xrightharpoonup{abc} <code>\xrightharpoonup{abc}</code>
\xleftharpoondown{abc} <code>\xleftharpoondown{abc}</code>	\xrightharpoondown{abc} <code>\xrightharpoondown{abc}</code>
\xleftrightharpoons{abc} <code>\xleftrightharpoons{abc}</code>	\xrightleftharpoons{abc} <code>\xrightleftharpoons{abc}</code>
\xleftrightarrow{abc} <code>\xleftrightarrow{abc}</code>	\xmapsto{abc} <code>\xmapsto{abc}</code>
\xlongequal{abc} <code>\xlongequal{abc}</code>	

可扩展的箭头都可以用相同的方式接受可选参数，比如说，`\xrightarrow[under]{over}`显示出来就是 $\xrightarrow[under]{over}$ 。

[16]: 这里扩展箭头指箭头上下方可以放置文字, 且箭头长度随文字长度的增加而增加。

样式，颜色，大小和字体

类的分配^[17]

$$\mathbin{\mathbb{A}} \mathbin{\mathbb{B}} \mathbin{\mathbb{C}} \mathbin{\mathbb{D}}$$

颜色

$F = ma$ `\color{blue}` $F=ma$

请注意， KAT_EX 中的`\color`的作用类似于开关，使用后一定范围内所有的字体都会改变颜色。这与 $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ 一致，但是与MathJax不同。而 KAT_EX 得其他颜色指令中，它的颜色和内容都是参数；

$F = ma$ `\textcolor{#228B22}`{ $F=ma$ }

A `\colorbox{aqua}`{ A }[18]

A `\fcolorbox{red}{aqua}`{ A }

对于颜色的名称， KAT_EX 颜色函数将接受标准的HTML预定义颜色名称。他们还将接受CSS十六进制样式的RGB参数，比如`#ff0000`表示红色，而其中的`#`是可选的。（不过洛谷讨论区版本较旧，因此在那里必须加`#`）。

[18]：`\colorbox`和`\fcolorbox` 里面不能嵌套环境，但是`\box`能。

字体

第一列	第二列	第三列
$Ab0$ <code>\mathrm{Ab0}</code>	$Ab0$ <code>\mathbf{Ab0}</code>	<i>$Ab0$</i> <code>\mathit{Ab0}</code>
<i>$Ab0$</i> <code>\mathnormal{Ab0}</code>	$Ab0$ <code>\textbf{Ab0}</code>	<i>$Ab0$</i> <code>\textit{Ab0}</code>
$Ab0$ <code>\textrm{Ab0}</code>	$Ab0$ <code>\bf{Ab0}</code>	<i>$Ab0$</i> <code>\it</code> $Ab0$
$Ab0$ <code>\rm{Ab0}</code>	$Ab0$ <code>\bold{Ab0}</code>	$Ab0$ <code>\textup{Ab0}</code>
$Ab0$ <code>\textnormal{Ab0}</code>	$Ab0$ <code>\boldsymbol{Ab}</code>	AB <code>\Bbb{AB}</code>
$Ab0$ <code>\text{Ab0}</code>	$Ab0$ <code>\bm{Ab0}</code>	AB <code>\mathbb{AB}</code>
$Ab0$ <code>\mathsf{Ab0}</code>	$Ab0$ <code>\textmd{Ab0}</code>	$Ab0$ <code>\frak{Ab0}</code>
$Ab0$ <code>\textsf{Ab0}</code>	$Ab0$ <code>\mathhtt{Ab0}</code>	$Ab0$ <code>\mathfrak{Ab0}</code>
$Ab0$ <code>\sf</code> $Ab0$	$Ab0$ <code>\texttt{Ab0}</code>	$AB0$ <code>\mathcal{AB0}</code>
$Ab0$ <code>\tt</code> $Ab0$	AB <code>\mathscr{AB}</code>	

通过使用`\textXX`类型的字体指令，可以堆叠字体样式、粗细和形状。因此`\textsf{\textbf{H}}`将渲染出**H**。其他字体指令不会堆叠，例如`\mathsf{\mathbf{H}}`将渲染出**H**。

如果KaTeX字体没有粗体字形，你可以使用`\pmb`指令可以模拟一个。例如，`\pmb{\mu}`呈现为： μ 。（所以莫反一类文章的大量希腊字母粗体都可以用这种方法生成——编者注）。

大小

第一列	第二列
AB <code>\Huge AB</code>	AB <code>\normalsize AB</code>
AB <code>\huge AB</code>	AB <code>\small AB</code>
AB <code>\LARGE AB</code>	AB <code>\footnotesize AB</code>
AB <code>\Large AB</code>	AB <code>\scriptsize AB</code>
AB <code>\large AB</code>	AB <code>\tiny AB</code>

样式


显示效果	代码
$\sum_{i=1}^n$	<code>\displaystyle\sum_{i=1}^n</code>
$\sum_{i=1}^n$	<code>\textstyle\sum_{i=1}^n</code>
x	<code>\scriptstyle x</code> (The size of a first sub/superscript)
x	<code>\scriptscriptstyle x</code> (The size of subsequent sub/superscripts)
\lim_x	<code>\lim\limits_x</code>
\lim_x	<code>\lim\nolimits_x</code>
x^2	<code>\verb!x^2!</code> ^[19]

在居中的环境下（即在 $\$ \$$ $\$ \$$ 当中），在`\text{}`里面可以放置 $\$ \dots \$$ 用来显示数学模式。

[19]：`\verb`以两个感叹号作为括号，里面可以填充任意文字。使用等宽字体（打字机字体）。

符号和标点

第一列	第二列	第三列
% 注释	\dots <code>\dots</code>	KATeX <code>\KaTeX</code>
% <code>\%</code>	\cdots <code>\cdots</code>	LaTeX <code>\LaTeX</code>
# <code>\#</code>	\ddots <code>\ddots</code>	TeX <code>\TeX</code>
& <code>\&</code>	\ldots <code>\ldots</code>	∇ <code>\nabla</code>

第一列	第二列	第三列
	<code>\vdots</code>	∞ <code>\infty</code>
<code>\text{\textunderscore}</code>	<code>\dotsb</code>	∞ <code>\infin</code>
<code>\text{\text{--}}</code>	<code>\dotsc</code>	✓ <code>\checkmark</code>
<code>\text{\textendash}</code>	<code>\dotsi</code>	† <code>\dag</code>
<code>\text{\text{---}}</code>	<code>\dotsm</code>	† <code>\dagger</code>
<code>\text{\textemdash}</code>	<code>\dotso</code>	† <code>\text{\textdagger}</code>
<code>\text{\textasciitilde}</code>	<code>\sdot</code>	‡ <code>\ddag</code>
<code>\text{\textasciicircum}</code>	<code>\mathellipsis</code>	‡ <code>\ddagger</code>
[20]	<code>\text{\textellipsis}</code>	‡ <code>\text{\textdaggerdbl}</code>
<code>\text{\textquoteleft}</code>	□ <code>\Box</code>	‡ <code>\Dagger</code>
<code>\lq</code>	□ <code>\square</code>	∠ <code>\angle</code>
<code>\text{\textquoteright}</code>	■ <code>\blacksquare</code>	∠ <code>\measuredangle</code>
<code>\rq</code>	△ <code>\triangle</code>	∠ <code>\sphericalangle</code>
<code>\text{\textquotedblleft}</code>	▽ <code>\triangledown</code>	⤴ <code>\top</code>
" "	◀ <code>\triangleleft</code>	⊥ <code>\bot</code>
” <code>\text{\textquotedblright}</code>	▶ <code>\triangleright</code>	\$ <code>\\$</code>
<code>\colon</code>	▽ <code>\bigtriangledown</code>	\$ <code>\text{\textdollar}</code>
<code>\backprime</code>	△ <code>\bigtriangleup</code>	£ <code>\pounds</code>
<code>\prime</code>	▲ <code>\blacktriangle</code>	£ <code>\mathsterling</code>
<code>\text{\textless}</code>	▼ <code>\blacktriangledown</code>	£ <code>\text{\textsterling}</code>
<code>\text{\textgreater}</code>	◀ <code>\blacktriangleleft</code>	¥ <code>\yen</code>
<code>\text{\textbar}</code>	▶ <code>\blacktriangleright</code>	√ <code>\surd</code>
<code>\text{\textbardbl}</code>	◊ <code>\diamond</code>	° <code>\degree</code>
<code>\text{\textbraceleft}</code>	◊ <code>\Diamond</code>	° <code>\text{\textdegree}</code>
<code>\text{\textbraceright}</code>	◊ <code>\lozenge</code>	ℳ <code>\mho</code>
<code>\text{\textbackslash}</code>	◆ <code>\blacklozenge</code>	↘ <code>\diagdown</code>
¶ <code>\text{\P}</code>	★ <code>\star</code>	↗ <code>\diagup</code>
§ <code>\text{\S}</code>	★ <code>\bigstar</code>	♭ <code>\flat</code>
§ <code>\text{\sect}</code>	♣ <code>\clubsuit</code>	♮ <code>\natural</code>

第一列	第二列	第三列
© \copyright	♣ \clubs	# \sharp
® \circledR	◇ \diamondsuit	♥ \heartsuit
℞ \text{\textregistered}	◇ \diamonds	♥ \hearts
Ⓢ \circledS	♠ \spadesuit	♠ \spades
Ⓐ \text{\textcircled a}	✠ \maltese	

你也可以直接输入这些：£¥∇∞· ∠ ∆ ✠ ♥ ◇ ♣ ♭ ♮ # ✓ ... ∴ ⋯ ⋮ !

[20]：同样由于Markdown制表问题，无法显示字符`。这里的显示效果就是‘,同`lq。

单位

在KAT_EX中，单位与在T_EX中一样按比例进行处理。

但是很多地方，KAT_EX单位不同于CSS单位。







KAT _E X单位	备注	KAT _E X单位	备注
em	CSS em	bp	1/72 inch × F × G
ex	CSS ex	pc	12 KAT _E Xpt
mu	1/18 CSS em	dd	1238/1157 KAT _E Xpt
pt	1/72.27 inch × F × G	cc	14856/1157 KAT _E Xpt
mm	1 mm × F × G	nd	685/642 KAT _E Xpt
cm	1 cm × F × G	nc	1370/107 KAT _E Xpt
in	1 inch × F × G	sp	1/65536 KAT _E Xpt

其中，

- F =(围绕着HTML文字的字体大小)/(10pt)
- G 默认情况下为1.21，因为KAT_EX字体大小通常为1.21×周围字体大小。HTML网页的CSS可以覆盖此值。

样式和大小的影响范围：

单位	文本模式	script大小 ^[21]	huge大小
em 与 ex	■	■	■

单位	文本模式	script大小 ^[21]	huge大小
mu			
其他			

[21]：这里不确定如何翻译scriptscript和huge。

本文全部译自[KAT_EX官网](#)。

历时1.5天终于肝完了orz。希望能为洛谷的L^AT_EX发展做出一些贡献。可能有一些不足，希望能在博客的评论区提出来。