

Typst 大学数学

这是 Jim Hefferon 的《本科生 L^AT_EX 数学》的 Typst 版本，适用的 typst 版本为 0.13.1。原始版本可以在此链接中找到：
<https://gitlab.com/jim.hefferon/undergradmath>。

记号的含义

可以复杂实现，需要更简单的方法。

Rule One 任何数学内容，哪怕只有一个字符，都需要使用数学环境。因此，对于「 x 的值为 7」，输入 $\$x\$$ 的值为 $\$7\$$ 。

模板 您的文档至少应包含以下内容。

```
-- document body here --
```

常见结构

```
x^2 x^2      √2, √3 sqrt(2), root(n, 3)
x_{i,j} x_(i, j)  2/3, 2/3 2 / 3, 2 \ / 3 or 2 slash 3
```

书法字母 使用 $\$cal(A)\$$ 。

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

手写字母

通过将 `text()` 的 `stylistic-set` 参数更改为相应的集合，可以获取手写字母：

```
#show math.equation: set text(stylistic-set: 1)
$ cal(A B C D E F G H I J K L M N O P Q R S T U V W
X Y Z) $
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

希腊字母

α	alpha	ξ, Ξ	xi, Xi
β	beta	\omicron	omicron
γ, Γ	gamma, Gamma	π, Π	pi, Pi
δ, Δ	delta, Delta	ϖ	pi.alt
ϵ	epsilon.alt	ρ	rho
ε	epsilon	ϱ	rho.alt
ζ	zeta	σ, Σ	sigma, Sigma
η	eta	ς	sigma.alt
θ, Θ	theta, Theta	τ	tau
ϑ	theta.alt	υ, Υ	upsilon, Upsilon
ι	iota	ϕ, Φ	phi.alt, Phi
κ	kappa	φ	phi
λ, Λ	lambda, Lambda	χ	chi
μ	mu	ψ, Ψ	psi, Psi
ν	nu	ω, Ω	omega, Omega

集合与逻辑

\cup	union	\mathbb{R}	RR, bb(R)	\forall	forall
\cap	inter	\mathbb{Z}	ZZ, bb(Z)	\exists	exists
\subset	subset	\mathbb{Q}	QQ, bb(Q)	\neg	not
\subseteq	subset.eq	\mathbb{N}	NN, bb(N)	\vee	or
\supset	supset	\mathbb{C}	CC, bb(C)	\wedge	and
\supseteq	supset.eq	\emptyset	diameter	\vdash	tack.r
\in	in	\emptyset	nothing	\models	models
\notin	in.not	\aleph	alef	\setminus	without

想要否定一个运算符，如 $\not\subset$ ，要写成 `subset.not`。集合的补集 A^c 写法是 $A^{(\text{sans}(c))}$ (A° 的写法是 $A^{(\text{complement})}$ ，而 \overline{A} 的写法是 `overline(A)`)。

备注 在使用 `diameter` 代表 `\varnothing` 可能会导致一些困惑。然而，在 L^AT_EX 中，通过字符变体 `cv01` 提供

的 `\varnothing` 也是 `diameter` (参见 `newcm` §14.5)。因此，使用默认数学字体 *New Computer Modern Math* 的简单解决方案是定义一个新符号 `\varnothing` 为 `\let \varnothing = \math.diameter`。其他解决方案可以在 `Typst Examples Book` 中找到。

装饰符号

```
f' f', f prime      \dot{a} dot(a)      \tilde{a} tilde(a)
f'' f prime.double  \ddia{a} diaer(a)   \macron{a} macron(a)
\Sigma^* Sigma^*    \hat{a} hat(a)      \rightarrow(a) arrow(a)
```

如果修饰的字母是 i 或 j ，那么某些修饰需要使用 `dotless.i` 和 `dotless.j`，例如 $\dot{\imath}$ ，可以使用 `arrow(dotless.i)`。一些作者在表示向量时使用粗体：`bold(x)`。

输入 `overline(x + y)` 会生成 $\overline{x + y}$ ，而 `hat(x + y)` 会给出 $\widehat{x + y}$ 。可以在表达式中添加注释，例如在这里（还有 `overbrace(...)`）。

```
\underbrace{x + y}{|A|} underbrace(x + y, |A|)
```

点号 在列表中使用低点号表示为 $\{0, 1, 2, \dots\}$ ，输入为 `\{0, 1, 2, \dots\}`。在求和或乘积中使用居中点号表示为 $1 + \dots + 100$ ，输入为 `1 + \dots.h.c + 100`。您还可以使用垂直点号 `\dots.v`，对角线点号 `\dots.down` 和反对角线点号 `\dots.up`。

函数名称 直接输入！

<code>sin</code>	<code>sin</code>	<code>sinh</code>	<code>sinh</code>	<code>arcsin</code>	<code>arcsin</code>
<code>cos</code>	<code>cos</code>	<code>cosh</code>	<code>cosh</code>	<code>arccos</code>	<code>arccos</code>
<code>tan</code>	<code>tan</code>	<code>tanh</code>	<code>tanh</code>	<code>arctan</code>	<code>arctan</code>
<code>sec</code>	<code>sec</code>	<code>coth</code>	<code>coth</code>	<code>min</code>	<code>min</code>
<code>csc</code>	<code>csc</code>	<code>det</code>	<code>det</code>	<code>max</code>	<code>max</code>
<code>cot</code>	<code>cot</code>	<code>dim</code>	<code>dim</code>	<code>inf</code>	<code>inf</code>
<code>exp</code>	<code>exp</code>	<code>ker</code>	<code>ker</code>	<code>sup</code>	<code>sup</code>
<code>log</code>	<code>log</code>	<code>deg</code>	<code>deg</code>	<code>lim inf</code>	<code>liminf</code>
<code>ln</code>	<code>ln</code>	<code>arg</code>	<code>arg</code>	<code>lim sup</code>	<code>limsup</code>
<code>lg</code>	<code>lg</code>	<code>gcd</code>	<code>gcd</code>	<code>lim</code>	<code>lim</code>

如果您想要使用的函数名不存在，您可以使用 `math.op` 来定义。例如，定义 `cosec` 函数：

```
#let cosec = math.op("cosec")
$cosec x = 1/(\sin x) $
```

$$\operatorname{cosec} x = \frac{1}{\sin x}$$

其他符号

$<$	<code><, lt</code>	\angle	<code>angle</code>	\cdot	<code>dot</code>
\leq	<code><=, lt.eq</code>	\sphericalangle	<code>angle.arc</code>	\pm	<code>plus.minus</code>
$>$	<code>>, gt</code>	ℓ	<code>ell</code>	\mp	<code>minus.plus</code>
\geq	<code>>=, gt.eq</code>	\parallel	<code>parallel</code>	\times	<code>times</code>
\neq	<code>!=, eq.not</code>	45°	<code>45 degree</code>	\div	<code>div</code>
\ll	<code><<, lt.double</code>	\cong	<code>tilde.equiv</code>	$*$	<code>*, ast</code>
\gg	<code>>>, gt.double</code>	\ncong	<code>tilde.nequiv</code>	$ $	<code>divides</code>
\approx	<code>approx</code>	\sim	<code>~, tilde</code>	$\}$	<code>divides.not</code>
\asymp	<code>asympt</code>	\simeq	<code>tilde.eq</code>	$n!$	<code>n!</code>
\equiv	<code>equiv</code>	\simeq	<code>tilde.not</code>	∂	<code>diff</code>
\prec	<code>prec</code>	\oplus	<code>plus.circle</code>	∇	<code>nabla</code>
\preceq	<code>prec.eq</code>	\ominus	<code>minus.cirle</code>	\hbar	<code>planck.reduce</code>
\succ	<code>succ</code>	\odot	<code>dot.circle</code>	\circ	<code>compose</code>
\succeq	<code>succ.eq</code>	\otimes	<code>times.circle</code>	\star	<code>star</code>
\propto	<code>prop</code>	\oslash	<code>\u{2298}</code>	$\sqrt{\quad}$	<code>sqrt("")</code>
\doteq	<code>\u{2250}</code>	\upharpoonright	<code>harpoon.tr</code>	\checkmark	<code>checkmark</code>

使用 `a divides b` 表示整除, $a \mid b$, 使用 `a divides.not b` 表示不能整除, $a \nmid b$. 使用 `|` 来表示集合构建符号, $\{a \in S \mid a \text{ is odd}\}$ 可以表示为 `{a in S | a "is odd"}`.

箭头

\rightarrow	<code>->, arrow.r</code>	\mapsto	<code> ->, arrow.r.bar</code>
\nrightarrow	<code>arrow.r.not</code>	\mapsto	<code>arrow.r.long.bar</code>
\longrightarrow	<code>-->, arrow.r.long</code>	\leftarrow	<code><-, arrow.l</code>
\Rightarrow	<code>=>, arrow.r.double</code>	\longleftrightarrow	<code><->, arrow.l.r</code>
\nRightarrow	<code>arrow.r.double.not</code>	\downarrow	<code>arrow.b</code>
\Longrightarrow	<code>==>, arrow.r.double.long</code>	\uparrow	<code>arrow.t</code>
\rightsquigarrow	<code>arrow.squiggly</code>	\Downarrow	<code>arrow.t.b</code>

第一列中的右箭头有相应的左箭头, 例如 `arrow.l.not`, 还有一些其他匹配的向下箭头等。

可变大小的运算符 求和符号 $\sum_{j=0}^3 j^2$ `sum_(j = 0)^3 j^2` 和积分符号 $\int_{x=0}^3 x^2 dx$ `integral_(x = 0)^3 x^2 dif x` 在行间模式会展开。

$$\sum_{j=0}^3 j^2 \quad \int_{x=0}^3 x^2 dx$$

下面这些同理。

\int	<code>integral</code>	\iint	<code>integral.double</code>
\iiint	<code>integral.triple</code>	\oint	<code>integral.cont</code>
\bigcup	<code>union.big</code>	\bigcap	<code>inter.big</code>

括号

$()$	<code>()</code>	$\langle \rangle$	<code>angle.l angle.r</code>	$ $	<code>abs("")</code>
$[]$	<code>[]</code>	$\lfloor \rfloor$	<code>floor("")</code>	$ $	<code>norm("")</code>
$\{ \}$	<code>\{ \}</code>	$\lceil \rceil$	<code>ceil("")</code>		

使用 `lr` 函数来固定大小。

$$\left[\sum_{k=0}^n e^{k^2} \right] \quad \text{lr}([\text{sum_}(k = 0)^n e^{(k^2)}], \text{size: } \#50\%)$$

为使它们与括号中的公式一起增长, 也可以使用 `lr` 函数。

$$\langle i, 2^{2^i} \rangle \quad \text{lr}(\text{angle.l } i, 2^{(2^i)} \text{ angle.r})$$

如果直接输入为代码点, 则括号默认会按比例缩放, 而如果以符号表示法输入, 则括号不会自动缩放。

$$\left(\frac{1}{n^\alpha} \right) \quad (1 / n^{(\alpha)})$$
$$\left(\frac{1}{n^\alpha} \right) \quad \text{paren.l } 1 / n^{(\alpha)} \text{ paren.r}$$

`lr` 函数还允许对不匹配的定界符和单侧括号进行缩放。

$$\left. \frac{df}{dx} \right|_{x_0} \quad \text{lr}(\text{frac}(dif f, dif x) |)_{(x_0)}$$

数组、矩阵 使用 `mat` 函数可以创建一个矩阵。可以将一个数组传递给它。

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \$ \text{mat}(a, b; c, d) \$$$

在 `Typst` 中, `array` 是一组数值, 而在 \LaTeX 中, `array` 是没有括号的矩阵, 相当于在 `Typst` 中使用 `$mat(delim: \#none, ..)$`。

对于行列式, 可以使用 `|A|`, 文本运算符 `det` `det` 或者 `mat(delim: "|", ..)`。

使用 `cases` 函数可以轻松定义分段函数。

$$f_n = \begin{cases} a & \text{if } n = 0 \\ r \cdot f_{n-1} & \text{else} \end{cases} \quad \$ f_n = \text{cases}($$
$$a \ \&"if" \ n = 0,$$
$$r \ \text{dot } f_ (n - 1) \ \&"else"$$
$$) \$$$

数学中的间距 将 $\sqrt{2}x$ 改进为带有细小间距的 $\sqrt{2}x$, 要写成 `sqrt(2) thin x`。稍宽一些的间距是 `med` 和 `thick` (它们的比例是 3 : 4 : 5)。更大的间距是 `quad` 和 `wide`, 效果分别是 \rightarrow \leftarrow 和 \rightarrow \leftarrow , 在行间公式的不同部分之间非常有用。使用 `h` 函数可以获取任意间距。例如, 使用 `\#h(-0.1667em)` 可以得到 \LaTeX 中的 `\!`。

行间公式 将行间公式以块级形式使用 `$... $`, 其中数学内容和 `$` 之间至少有一个空格分隔。

$$S = k \cdot \lg W \quad \$ S = k \ \text{dot } \lg W \$$$

你可以写成多行。

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$
$$\$ \sin(x) = x - x^3 / 3! \ \backslash$$
$$+ x^5 / 5! - \text{dots.h.c } \$$$

用 `&` 来对齐公式

$$\nabla \cdot \boldsymbol{D} = \rho \quad \$ \nabla \ \text{dot } \text{bold}(D) \ \&= \ \rho \ \backslash$$
$$\nabla \cdot \boldsymbol{B} = 0 \quad \nabla \ \text{dot } \text{bold}(B) \ \&= \ 0 \ \$$$

(对齐的左侧或右侧可以为空)。通过 `\#set math.equation(numbering: ..)` 给公式加编号。

微积分例子 最后三个是行间公式形式。

$$f: \mathbb{R} \rightarrow \mathbb{R} \quad f: \mathbb{R} \rightarrow \mathbb{R}$$
$$9.8 \text{ m/s}^2 \quad 9.8 \text{ thin "m/s"}^2$$
$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad \lim_{(h \rightarrow 0)} (f(x+h) - f(x)) / h$$
$$\int x^2 dx = x^3/3 + C \quad \text{integral } x^2 \text{ dif } x = x^3 \ \backslash / 3 + C$$
$$\nabla = i \frac{d}{dx} + j \frac{d}{dy} + k \frac{d}{dz} \quad \nabla = \text{bold}(i) \ \text{dif} / (\text{dif } x) + \text{bold}(j) \ \text{dif} / (\text{dif } y) + \text{bold}(k) \ \text{dif} / (\text{dif } z)$$

离散数学例子 对于模运算, 可以使用 `equiv` 来输入 \equiv , 用 `mod` 来输入文本运算符 `mod`。

对于组合, 可以使用 `binom(n, k)` 中的二项式符号 $\binom{n}{k}$ 。在行间模式下会自动调整大小。

对于排列, 可以使用 `n^(underline(r))` 来输入 $n^{\underline{r}}$ 符号 (有些作者用 $P(n, r)$ 或 ${}_nP_r$ 来表示, 可以使用 `"_n P_r`。

统计学例子

$$\sigma^2 = \sqrt{\sum (x_i - \mu)^2 / N} \quad \text{sigma}^2 = \text{sqrt}(\text{sum}(x_i - \mu)^2 \ \backslash / \ N)$$
$$E(X) = \mu_X = \sum (x_i - P(x_i)) \quad E(X) = \mu_X = \text{sum}(x_i - P(x_i))$$
$$\frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad 1 / \text{sqrt}(2 \ \text{sigma}^2 \ \text{pi}) \ e^{-(x - \mu)^2 / (2 \ \text{sigma}^2)}$$

更多 参见 Typst 文档: <https://typst.app/docs>.