

# 实用主义家用 L<sup>A</sup>T<sub>E</sub>X 入门

*L<sup>A</sup>T<sub>E</sub>X for Pragmatists*

---

INNOCENT, PANADOL  
二〇二三年十二月八日

中山大学 笃行工作室 & SPS 物理协会



GitHub 链接



QQ 交流群

# 目录

- 从零安装数学公式
- 文档结构
- Debugging
- 私货

# 从零

---

萬物是藉著他造的；凡被造的，沒有一樣不是藉著他造的。

—JOHN 1:3. *The Word Became Flesh*

# 实用主义者的自白

# 实用主义者的自白

在一切一切开始之前……我们用最简洁的方式来介绍 LATEX, 我们将会:

# 实用主义者的自白

在一切一切开始之前……我们用最简洁的方式来介绍 LATEX, 我们将会:

- 给出写出一篇 (最基本的) 文档的方法, 将更具体的 LATEX 特性阐释后置;

# 实用主义者的自白

在一切一切开始之前……我们用最简洁的方式来介绍 LATEX，我们将会：

- 给出写出一篇（最基本的）文档的方法，将更具体的 LATEX 特性阐释后置；
- 也许你会说：“**我还不知道 LATEX 怎么下载呢！**”别急，我们将使用在线的 TeXPage；

在一切一切开始之前……我们用最简洁的方式来介绍 LATEX，我们将会：

- 给出写出一篇（最基本的）文档的方法，将更具体的 LATEX 特性阐释后置；
- 也许你会说：“**我还不知道 LATEX 怎么下载呢！**”别急，我们将会使用在线的 TeXPage；
- 在这个阶段，我说的话可能是有失偏颇的……但说得太多可能会引发初学者的畏难情绪，而这在一开始是没有必要的；

在一切一切开始之前……我们用最简洁的方式来介绍 LATEX，我们将会：

- 给出写出一篇（最基本的）文档的方法，将更具体的 LATEX 特性阐释后置；
- 也许你会说：“**我还不知道 LATEX 怎么下载呢！**”别急，我们将会使用在线的 TeXPage；
- 在这个阶段，我说的话可能是有失偏颇的……但说得太多可能会引发初学者的畏难情绪，而这在一开始是没有必要的；
- 总而言之，先让它动起来再说。

# 简要的介绍

- L<sup>A</sup>T<sub>E</sub>X 是什么？

# 简要的介绍

- L<sup>A</sup>T<sub>E</sub>X 是什么?
  - 粗略来说, 是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装;

# 简要的介绍

- L<sup>A</sup>T<sub>E</sub>X 是什么?
  - 粗略来说, 是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装;
  - “排版公式很厉害且写论文 / 提交作业要用的”的排版系统;

- **L<sup>A</sup>T<sub>E</sub>X 是什么？**

- 粗略来说，是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装；
- “排版公式很厉害且写论文 / 提交作业要用的”的排版系统；
- 从嘴里说出来时，L<sup>A</sup>T<sub>E</sub>X（大概）是 T<sub>E</sub>X、L<sup>A</sup>T<sub>E</sub>X 内核、文档类、宏包的总称；

- **L<sup>A</sup>T<sub>E</sub>X 是什么？**

- 粗略来说，是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装；
- “排版公式很厉害且写论文 / 提交作业要用的”的排版系统；
- 从嘴里说出来时，L<sup>A</sup>T<sub>E</sub>X（大概）是 T<sub>E</sub>X、L<sup>A</sup>T<sub>E</sub>X 内核、文档类、宏包的总称；
- L<sup>A</sup>T<sub>E</sub>X 语法是数学家之间的交流方式.

- L<sup>A</sup>T<sub>E</sub>X 是什么?
  - 粗略来说, 是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装;
  - “排版公式很厉害且写论文 / 提交作业要用的”的排版系统;
  - 从嘴里说出来时, L<sup>A</sup>T<sub>E</sub>X (大概) 是 T<sub>E</sub>X、L<sup>A</sup>T<sub>E</sub>X 内核、文档类、宏包的总称;
  - L<sup>A</sup>T<sub>E</sub>X 语法是数学家之间的交流方式.
- T<sub>E</sub>X 是什么?

- L<sup>A</sup>T<sub>E</sub>X 是什么?
  - 粗略来说, 是 Leslie Lamport 对 T<sub>E</sub>X 的一种封装;
  - “排版公式很厉害且写论文 / 提交作业要用的”的排版系统;
  - 从嘴里说出来时, L<sup>A</sup>T<sub>E</sub>X (大概) 是 T<sub>E</sub>X、L<sup>A</sup>T<sub>E</sub>X 内核、文档类、宏包的总称;
  - L<sup>A</sup>T<sub>E</sub>X 语法是数学家之间的交流方式.
- T<sub>E</sub>X 是什么?
  - Donald E. Knuth 和他的学生开发的排版文字和数学公式而开发的排版引擎;

# 简要的介绍

- $\text{\LaTeX}$  是什么?

- 粗略来说, 是 Leslie Lamport 对  $\text{\TeX}$  的一种封装;
- “排版公式很厉害且写论文 / 提交作业要用的”的排版系统;
- 从嘴里说出来时,  $\text{\LaTeX}$  (大概) 是  $\text{\TeX}$ 、 $\text{\LaTeX}$  内核、文档类、宏包的总称;
- $\text{\LaTeX}$  语法是数学家之间的交流方式.

- $\text{\TeX}$  是什么?

- Donald E. Knuth 和他的学生开发的排版文字和数学公式而开发的排版引擎;
- $\text{\LaTeX}$  的底层, 真正负责排版的东西.

# 演示: L<sup>A</sup>T<sub>E</sub>X, 启动!

按照惯例, 我们先来给出我们的第一篇文档 (还有最基本的中英混排的例子).

# 演示: L<sup>A</sup>T<sub>E</sub>X, 启动!

按照惯例, 我们先来给出我们的第一篇文档 (还有最基本的中英混排的例子).

```
\documentclass{article}  
% 这里是导言区, % 用来表示注释  
\begin{document}  
Hello, world!  
\end{document}
```

# 演示: L<sup>A</sup>T<sub>E</sub>X, 启动!

按照惯例, 我们先来给出我们的第一篇文档 (还有最基本的中英混排的例子).

```
\documentclass{article}  
% 这里是导言区, % 用来表示注释  
\begin{document}  
Hello, world!  
\end{document}
```

- 其中, \documentclass 是一个控制序列, {article} 是其的一个必要参数, 参数值是 article, 表示调用 article 文档类.

# 演示: L<sup>A</sup>T<sub>E</sub>X, 启动!

按照惯例, 我们先来给出我们的第一篇文档 (还有最基本的中英混排的例子).

```
\documentclass{article}  
% 这里是导言区, % 用来表示注释  
\begin{document}  
Hello, world!  
\end{document}
```

- 其中, `\documentclass` 是一个控制序列, `{article}` 是其的一个必要参数, 参数值是 `article`, 表示调用 `article` 文档类.
- 导言区是用来设置整篇文档的格式的, 他们往往不会决定文档中输出的**内容**.

# 演示: L<sup>A</sup>T<sub>E</sub>X, 启动!

按照惯例, 我们先来给出我们的第一篇文档 (还有最基本的中英混排的例子).

```
\documentclass{article}
% 这里是导言区, % 用来表示注释
\begin{document}
Hello, world!
\end{document}
```

- 其中, `\documentclass` 是一个控制序列, `{article}` 是其的一个必要参数, 参数值是 `article`, 表示调用 `article` 文档类.
- 导言区是用来设置整篇文档的格式的, 他们往往不会决定文档中输出的**内容**.
- `\begin{document}` 与 `\end{document}` 之间的是最基本的 `document` 环境, 用来填写主要内容的地方.

# 演示：中文支持

## 演示：中文支持

先上最新最热的配置：CT<sub>E</sub>X 宏集 + X<sub>E</sub>LATE<sub>X</sub> 程序。如果你不清楚这两个名词意味着什么，请看下面的代码：

# 演示：中文支持

先上最新最热的配置：CT<sub>E</sub>X 宏集 + X<sub>E</sub>L<sub>A</sub>T<sub>E</sub>X 程序。如果你不清楚这两个名词意味着什么，请看下面的代码：

```
\documentclass{ctexart}
% 使用 XeLaTeX 编译
\begin{document}
你好 :)
\end{document}
```

# 演示：中文支持

先上最新最热的配置：CT<sub>E</sub>X 宏集 + X<sub>E</sub>L<sub>A</sub>T<sub>E</sub>X 程序。如果你不清楚这两个名词意味着什么，请看下面的代码：

```
\documentclass{ctexart}
% 使用 XeLaTeX 编译
\begin{document}
你好 :)
\end{document}
```

- CT<sub>E</sub>X 宏集是面向中文排版的大框架，而并非是 CT<sub>E</sub>X 套装，  
ctexart 是专为中文排版理过的 article 类；

# 演示：中文支持

先上最新最热的配置：CT<sub>E</sub>X 宏集 + X<sub>E</sub>LATE<sub>X</sub> 程序。如果你不清楚这两个名词意味着什么，请看下面的代码：

```
\documentclass{ctexart}
% 使用 XeLaTeX 编译
\begin{document}
你好 :)
\end{document}
```

- CT<sub>E</sub>X 宏集是面向中文排版的大框架，而并非是 CT<sub>E</sub>X 套装，`ctexart` 是专为中文排版理过的 `article` 类；
- X<sub>E</sub>LATE<sub>X</sub> 总体说来是一个编译工具，在 TeXPage 里，我们可以看到 LuaLATE<sub>X</sub>、pdfLATE<sub>X</sub> 等类似物，但对中文排版来说，X<sub>E</sub>LATE<sub>X</sub> 是最合适的选择；

# 演示：中文支持

先上最新最热的配置：CT<sub>E</sub>X 宏集 + X<sub>e</sub>LATE<sub>X</sub> 程序。如果你不清楚这两个名词意味着什么，请看下面的代码：

```
\documentclass{ctexart}
% 使用 XeLaTeX 编译
\begin{document}
你好 :)
\end{document}
```

- CT<sub>E</sub>X 宏集是面向中文排版的大框架，而并非是 CT<sub>E</sub>X 套装，`ctexart` 是专为中文排版理过的 `article` 类；
- X<sub>e</sub>LATE<sub>X</sub> 总体说来是一个编译工具，在 TeXPage 里，我们可以看到 LuaLATE<sub>X</sub>、pdfLATE<sub>X</sub> 等类似物，但对中文排版来说，X<sub>e</sub>LATE<sub>X</sub> 是最合适；
- 过于古典的方法可能会导致一些问题，比如「复制出乱码」。

# 演示：数学公式 (一)

## 演示：数学公式（一）

- 当我们第一次使用 LATEX 时，大抵上都是为了输入数学公式.

# 演示：数学公式（一）

- 当我们第一次使用 L<sup>A</sup>T<sub>E</sub>X 时，大抵上都是为了输入数学公式。
- L<sup>A</sup>T<sub>E</sub>X 上的数学环境可以分为两种：
  - 行内模式 ( $\$ \dots \$$ ) :  $\mathcal{L}^2(G) \cong \bigoplus_{\pi \in \widehat{G}} \mathcal{E}_\pi$ ;
  - 行间模式 ( $\backslash[ \dots ]$  或 `equation` 环境) :

$$\mathcal{B}(\mathbb{R}) = \bigcup_{\xi < w_1} \Sigma_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Delta_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Pi_\xi^0(\mathbb{R}).$$

# 演示：数学公式（一）

- 当我们第一次使用 L<sup>A</sup>T<sub>E</sub>X 时，大抵上都是为了输入数学公式。
- L<sup>A</sup>T<sub>E</sub>X 上的数学环境可以分为两种：
  - 行内模式 ( $\$ \dots \$$ ) :  $\mathcal{L}^2(G) \cong \bigoplus_{\pi \in \widehat{G}} \mathcal{E}_\pi$ ;
  - 行间模式 ( $\backslash[ \dots ]$  或 `equation` 环境) :

$$\mathcal{B}(\mathbb{R}) = \bigcup_{\xi < w_1} \Sigma_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Delta_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Pi_\xi^0(\mathbb{R}).$$

- `amsmath` 宏包是科学文献写作的事实标准，最好始终调用：

# 演示：数学公式（一）

- 当我们第一次使用 LATEX 时，大抵上都是为了输入数学公式。
- LATEX 上的数学环境可以分为两种：
  - 行内模式 ( $\$ \dots \$$ ) :  $\mathcal{L}^2(G) \cong \bigoplus_{\pi \in \widehat{G}} \mathcal{E}_\pi$ ;
  - 行间模式 ( $\begin{array}{l} \dots \end{array}$  或 equation 环境) :

$$\mathcal{B}(\mathbb{R}) = \bigcup_{\xi < w_1} \Sigma_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Delta_\xi^0(\mathbb{R}) = \bigcup_{\xi < w_1} \Pi_\xi^0(\mathbb{R}).$$

- amsmath 宏包是科学文献写作的事实标准，最好始终调用：

```
\documentclass{ctexart}
\usepackage{amsmath} % 用 \usepackage 调用宏包
\begin{document}
  \begin{array}{l}
    \hbar = c = 1.
  \end{array}
\end{document}
```

## 演示：数学公式 (二)

## 演示：数学公式 (二)

- 括号. (), [], \{\}\}. 自动控制大小是 \left...、\right...:

```
\[
\left(\frac{1}{3}\right]
+\left(\frac{1}{x}\right]^2
\]
```

## 演示：数学公式 (二)

- 括号. ()、[]、\{\}.\ 自动控制大小是 \left...、\right...:

```
\[
\left(\frac{1}{3}\right) + \left\{\frac{1}{x}\right\}^2
+\left(\frac{1}{x}\right)^2
```

## 演示：数学公式（二）

- 括号. ()、[]、\{\}.\ 自动控制大小是 \left...、\right...:

```
\[
\left(\frac{1}{3}\right) + \left\{\frac{1}{x}\right\}^2
+\left(\frac{1}{x}\right)^2
\]
```

- 上下标 (^{...}、\_{...})、分式 (\frac{分子}{分母})、根式 (\sqrt[次数]{内容}):

```
\[
\int_0^1 x^7 dx =
\frac{1}{8} =
\left( \frac{\sqrt[3]{8}}{8} \right)^2
\right)^3.
\]
```

## 演示：数学公式 (二)

- 括号. ()、[]、\{\}.\ 自动控制大小是 \left...、\right...:

```
\[
\left(\frac{1}{3}\right) + \left\{\frac{1}{x}\right\}^2
+\left(\frac{1}{x}\right)^2
\]
```

- 上下标 (^{...}、\_{...})、分式 (\frac{分子}{分母})、根式 (\sqrt[次数]{内容}):

```
\[
\int_0^1 x^7 dx =
\frac{1}{8} =
\left(\frac{\sqrt[3]{8}}{2}\right)^3.
\]
```

$$\int_0^1 x^7 dx = \frac{1}{8} = \left(\frac{\sqrt[3]{8}}{2}\right)^3.$$

## 演示：数学公式 (三)

- 矩阵。在调用 amsmath 宏包的情形下，使用 `pmatrix` 环境，其中 `&` 是分列，`\backslash` 是换行：

```
\begin{pmatrix}
  1 & 0 & 0 & 0 & 0 \\
  0 & 0 & -1 & 0 & 0 \\
  0 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0
\end{pmatrix}.
```

## 演示：数学公式 (三)

- 矩阵。在调用 amsmath 宏包的情形下，使用 `pmatrix` 环境，其中 `&` 是分列，`\backslash` 是换行：

```
\begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & 0 & -1 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0
\end{pmatrix}.
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

## 演示：组织文本（一）

- 节与段：`\section{...}`；小小节：`\subsection{...}`、段落：`\paragraph{...}`；

## 演示：组织文本（一）

- 节与段：`\section{...}`；小小节：`\subsection{...}`、段落：`\paragraph{...}`；
- 定理与证明：

# 演示：组织文本（一）

- 节与段：`\section{...}`；小小节：`\subsection{...}`、段落：`\paragraph{...}`；
- 定理与证明：

```
\usepackage{amsthm}
\newtheorem{theorem}{定理}
\begin{document}
\begin{theorem}[Riemann假设]
    Riemann  $\zeta$  函数的非平凡零点实部均为  $1/2$ 。
\end{theorem}

\begin{proof}
    因为这是假设，我们不妨假设显然成立。
\end{proof}
\end{document}
```

## 演示：组织文本（二）

- 列举环境：itemize、enumerate、description；

```
% \usepackage{enumitem}
% 处理 enumerate 环境编号格式
\begin{enumerate}[label=(\arabic*)]
\item Mark~Srednicki;
\item A.~Zee;
\begin{itemize}
\item M.~Peskin and D.~Schroede;
\item S.~Weinberg;
\end{itemize}
\begin{description}
\item [L.~Ryder] Quantum Field Theory;
\item [David Tong] QFT;
\end{description}
\end{enumerate}
```

# 演示：图片

```
% \usepackage{graphicx}
% 用来 \includegraphics
% \usepackage{hyperref}
% 超链接
\begin{figure}
    \centering
    \includegraphics[width=4cm]{images/li-a-ling.jpg}
    \caption{李阿玲}
    \label{fig:li-a-ling}
\end{figure}
```

尽可能用「如 `\autoref{fig:li-a-ling}` 所示」这种语言而不是「见上图、下表」。

图片来源: 

# 演示：图片

```
% \usepackage{graphicx}  
% 用来 \includegraphics  
% \usepackage{hyperref}  
% 超链接  
\begin{figure}  
    \centering  
    \includegraphics[width=4cm]{  
        images/li-a-ling.jpg}  
    \caption{李阿玲}  
    \label{fig:li-a-ling}  
\end{figure}
```

尽可能用「如 `\autoref{fig:li-a-ling}` 所示」这种语言而不是「见上图、下表」。

图片来源: 



图 1: 李阿玲

尽可能用「如图 1 所示」这种语言而不是「见上图、下表」。

# 演示：表格

```
% \usepackage{tabulararray}    % 个人喜好
% \usepackage{siunitx}        % 排版单位
% \UseTblrLibrary{booktabs} % 三线表
\begin{table}
\centering
\caption{...}
\label{tab:masses-particles}
\begin{tblr}{lll}
\toprule
Particle & Mass & \\
\midrule
electron & \qty{.5}{MeV} & \\
... & ... & \\
Higgs & Boson & \qty{125}{GeV} \\
\bottomrule
\end{tblr}
\end{table}
```

数据来源: DAVID TONG. *Quantum Field Theory*. 

# 演示：表格

```
% \usepackage{tabulararray}      % 个人喜好
% \usepackage{siunitx}          % 排版单位
% \UseTblrLibrary{booktabs}    % 三线表
\begin{table}
\centering
\caption{...}
\label{tab:masses-particles}
\begin{tblr}{lll}
\toprule
Particle & Mass & \\
\midrule
electron & & \\[.5mm]
& & \\
... & & \\[.5mm]
Higgs Boson & & \\[125mm]
\bottomrule
\end{tblr}
\end{table}
```

表 1: The rough masses of some elementary (*and not so elementary*) particles

Particle	Mass
electron	0.5 MeV
Muon	100 MeV
Pions	140 MeV
Proton, Neutron	1 GeV
Tau	2 GeV
W, Z Bosons	80–90 GeV
Higgs Boson	125 GeV

数据来源: DAVID TONG. *Quantum Field Theory*. ↗

## 演示：参考文献 (BIBTEX)

- 先声明：BIBTEX 是一个古典的方法，在这里使用它只是容易演示且资料丰富；
- 先要有一个 .bib 文件；
- \bibliographystyle{...}、\cite{...}、\bibliography{...}；

## 演示：参考文献 (BIBTEX)

- 先声明：BIBTEX 是一个古典的方法，在这里使用它只是容易演示且资料丰富；
- 先要有一个 `.bib` 文件；
- `\bibliographystyle{...}`、`\cite{...}`、  
`\bibliography{...}`；
- 文献的 BIBTEX 怎么找？（自己写？）

## 演示: 参考文献 (BIBTEX)

- 先声明: BIBTEX 是一个古典的方法, 在这里使用它只是容易演示且资料丰富;
- 先要有一个 .bib 文件;
- `\bibliographystyle{...}`、`\cite{...}`、  
`\bibliography{...}`;
- 文献的 BIBTEX 怎么找? (自己写?)
- 想要输出所有文献但懒得引用? (`\nocite{*}`).
- **要编译多次**, 否则就等着爆 [??] 吧.

## 演示: 参考文献 (BIBTEX)

- 先声明: BIBTEX 是一个古典的方法, 在这里使用它只是容易演示且资料丰富;
- 先要有一个 .bib 文件;
- \bibliographystyle{...}、\cite{...}、\bibliography{...};
- 文献的 BIBTEX 怎么找? (自己写?)
- 想要输出所有文献但懒得引用? (\nocite{\*}).
- **要编译多次**, 否则就等着爆 [??] 吧.

```
\cite{simon2015operator}
\bibliographystyle{alpha}
\nocite{*}
\bibliography{cite}
```

# 安裝

---

真的要安装吗？

overleaf 与 ShareLaTeX



+



Overleaf

ShareLaTeX

红配绿，那啥……

真的要安装吗？

overleaf 与 ShareLaTeX



+



Overleaf

ShareLaTeX

对于免费版使用者的限制见 (比如 20 s 编译时间?)

红配绿，那啥……

或者

- 越来越多的用户正在使用的 TeXPage (对中文的支持更好些, 产品定价 9):



清华大学  
Tsinghua University



北京大学  
PEKING UNIVERSITY



THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY



THE UNIVERSITY  
OF HONG KONG



中国科学院  
CHINESE ACADEMY OF SCIENCES



浙江大学  
ZHEJIANG UNIVERSITY



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



北京交通大学  
BEIJING JIAOTONG UNIVERSITY



天津大学  
Tianjin University

或者

- 越来越多的用户正在使用的 TeXPage (对中文的支持更好些, 产品定价 6):



清华大学  
Tsinghua University



北京大学  
PEKING UNIVERSITY



THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY



THE UNIVERSITY  
OF HONG KONG



中国科学院  
CHINESE ACADEMY OF SCIENCES



UMass Chan  
MEDICAL SCHOOL



浙江大学  
ZHEJIANG UNIVERSITY



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



北京交通大学  
BEIJING JIAOTONG UNIVERSITY



天津大学  
Tianjin University

- 合理的建议: 先不要急着安装, 用用在线平台试试手;

或者

- 越来越多的用户正在使用的 TeXPage (对中文的支持更好些, 产品定价 6):



清华大学  
Tsinghua University



北京大学  
PEKING UNIVERSITY



THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY



THE UNIVERSITY  
OF HONG KONG



中国科学院  
CHINESE ACADEMY OF SCIENCES



UMass Chan  
MEDICAL SCHOOL



浙江大学  
ZHEJIANG UNIVERSITY



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY



北京师范大学  
BEIJING JIAOTONG UNIVERSITY



天津大学  
Tianjin University

- 合理的建议: 先不要急着安装, 用用在线平台试试手;
- 以上这些云端部署的情形可以免去安装、升级等一系列烦恼, 更方便多人协作, 如果你是本地人的话……

## 请选择你的发行版

- 发行版是引擎 + 宏包 + 字体 + 文档 + 各种各样的东西……
- 也就是“全家桶”.

## 请选择你的发行版

- 发行版是引擎 + 宏包 + 字体 + 文档 + 各种各样的东西……
- 也就是“全家桶”.

### TeX live

-   
- 官方维护，居家必备；
- 不方便出门携带  
(指安装包体积很大)；
- 每年都要更新.

# 请选择你的发行版

- 发行版是引擎 + 宏包 + 字体 + 文档 + 各种各样的东西……
- 也就是“全家桶”.

TeX live



- 官方维护, 居家必备;
- 不方便出门携带(指安装包体积很大);
- 每年都要更新.

MiKTeX



- 宏包可以等到要用的时候再装;
- 安装宏包时可能卡网络问题.

# 请选择你的发行版

- 发行版是引擎 + 宏包 + 字体 + 文档 + 各种各样的东西……
- 也就是“全家桶”.

## T<sub>E</sub>X live

-   
- 官方维护, 居家必备;
- 不方便出门携带(指安装包体积很大);
- 每年都要更新.

## MiK<sub>T</sub>E<sub>X</sub>

-   
- 宏包可以等到要用的时候再装;
- 安装宏包时可能卡网络问题.

## CT<sub>E</sub>X

- 
- 2022 年, 吴凌云提交了 CT<sub>E</sub>X 的更新, 所以现在也许能用, 但暂不推荐;
- 处理历史文档或投稿部分国内期刊时可以考虑使用.

# 请选择你的发行版

- 发行版是引擎 + 宏包 + 字体 + 文档 + 各种各样的东西……
- 也就是“全家桶”.

## T<sub>E</sub>X live



- 官方维护, 居家必备;
- 不方便出门携带(指安装包体积很大);
- 每年都要更新.

## MiK<sub>T</sub>E<sub>X</sub>



- 宏包可以等到要用的时候再装;
- 安装宏包时可能卡网络问题.

## CT<sub>E</sub>X



- 2022 年, 吴凌云提交了 CT<sub>E</sub>X 的更新, 所以现在也许能用, 但暂不推荐;
- 处理历史文档或投稿部分国内期刊时可以考虑使用.

TinyT<sub>E</sub>X: 字面意思上地, 是 T<sub>E</sub>X live 的微缩版, 假设用户不惧怕或反感使用命令行.

## Easy TeXWorks & TeX Studio

开箱即用，专为 TeX 使用者定制。

## **Easy** TeXWorks & TeX Studio

开箱即用, 专为 TeX 使用者定制.

## **Normal** Visual Studio Code

最新最热的代码编辑器 (之一).

## **Easy** TeXWorks & TeX Studio

开箱即用, 专为 TeX 使用者定制.

## **Normal** Visual Studio Code

最新最热的代码编辑器 (之一).

## **Hard** Vim

获得乱码的最好方式是打开 Vim 直到保存退出.

**Easy** TeXWorks & TeX Studio

开箱即用, 专为 TeX 使用者定制.

**Normal** Visual Studio Code

最新最热的代码编辑器 (之一).

**Hard** Vim

获得乱码的最好方式是打开 Vim 直到保存退出.

**Lunatic** 针

小心出国杳无音讯.

# 如何安装？

- 就交给实操演示吧！

# 如何安装?

- 就交给实操演示吧!
  - 没有发簪? 可以看看清华 (¶)、上交 (¶) 和中科大 (¶) 的镜像网.

# 如何安装?

- 就交给实操演示吧!
  - 没有发簪? 可以看看清华 (⌚)、上交 (⌚) 和中科大 (⌚) 的镜像网.
  - 或者……不惧怕命令行读者的教程 ⌚

# 如何安装?

- 就交给实操演示吧!
  - 没有发簪? 可以看看清华 (⌚)、上交 (⌚) 和中科大 (⌚) 的镜像网.
  - 或者……不惧怕命令行读者的教程 ⌚

公理 (安装路径)

路径尽量不要用中文、空格、特殊符号.

# 开箱

定理 (命令行)

很多人都不想看到命令行.

# 开箱

## 定理 (命令行)

很多人都不想看到命令行.

### 证明.

已由 C 证明, 我们会将命令行的部分放到最后面. 需要注意的是, 编译的选项很多, 远不是一个 ▶ 能够涵盖的. □

## 定理(命令行)

很多人都不想看到命令行.

### 证明.

已由  证明, 我们会将命令行的部分放到最后面. 需要注意的是, 编译的选项很多, 远不是一个  能够涵盖的. 

我们在此暂时忽略易混淆的引擎、格式之类, 仅仅列举现今最常用的几种 T<sub>E</sub>X 程序对比:

## 定理(命令行)

很多人都不想看到命令行.

### 证明.

已由  证明, 我们会将命令行的部分放到最后面. 需要注意的是, 编译的选项很多, 远不是一个  能够涵盖的. 

我们在此暂时忽略易混淆的引擎、格式之类, 仅仅列举现今最常用的几种  $\text{\TeX}$  程序对比:

- pdf $\text{\LaTeX}$ : 不直接支持 Unicode, 但是支持 micro-typography;
- Xe $\text{\LaTeX}$ : 支持 Unicode 和 OpenType, 目前中文社区的通用程序;
- Lua $\text{\LaTeX}$ : 支持 Unicode 和 OpenType, 内联 Lua.

## 数学公式

---

很多常见的公式服务依赖于一些轻量级渲染引擎，比如 MathJax、KaTeX。但是它们实际上使用的是  $\text{\LaTeX}$  语法变种，也就是说并没有使用  $\text{\LaTeX}$  后端。所以不要期望有完全一致的输出。

# 如何在数学环境下输入中文

- 想要在 LATEX 输入中文，直接输入是做不到的，最有性价比的方法是使用 `\text{...}`，这需要调用 amsmath 宏包：  
`\[\hbar=1,\quad \text{真的}.\]`

$\hbar = 1,$  真的.

# 如何在数学环境下输入中文

- 想要在 LATEX 输入中文，直接输入是做不到的，最有性价比的方法是使用 `\text{...}`，这需要调用 amsmath 宏包：  
`\[\hbar=1,\quad \text{真的}.\]`

$$\hbar = 1, \quad \text{真的.}$$

- 数学环境中的空格会被忽略，但 `\text{...}` 中的不会。
  - `$A \text{开集}$`: A开集；
  - `$A\text{ 开集}$`: A 开集.

- 数学环境的结构分为以下几种：
  - 原子. 如  $A, \Delta, f$  之类;
  - 开闭结构. 如  $(, ]$ ,  $\langle$  之类;
  - 算子. 如  $\sin, \tan$  之类, 最主要的特征是**正体**;
  - 大算子. 如  $\sum, \oplus, \iiint$  之类;
  - 标点. 基本的逗号, 分号和句号;
  - 关系符或者二元算符.  $<, +, \rightarrow$  之类.

# 数学环境的细节 (一)

- 括号. 大括号在 L<sup>A</sup>T<sub>E</sub>X 中起分组作用, 输入应用  $\{\}, \{\}$ , 绝对值可以用  $|\dots|$ , 范数可以用  $\|\dots\|$ .

$$\{A\} = \|\Psi\| + |x|.$$

- 括号. 大括号在 LATEX 中起分组作用, 输入应用 `\{, \}`, 绝对值可以用 `|...|`, 范数可以用 `\| ... \|`.

$$\{A\} = \|\Psi\| + |x|.$$

- 大大大大括号.
  - 用 `\left, \right` 是最省事的方法:

$$\left( 1, \sum_{n \geq 1} \frac{1}{n^2} \right)$$

`\left(1, \sum_{n \geq 1} \frac{1}{n^2} \right)`

- 手动解决方案: `\big, \Big, \bigg, \Bigg` 四挡.

## 数学环境的细节 (二)

- 张量. 可以用 `tensor` 宏包.

`\tensor{\Lambda}{^\mu _\sigma}` 而不是  
`\Lambda^\mu_\sigma`:

$$\Lambda^\mu_\sigma \neq \Lambda^\mu_\sigma.$$

## 数学环境的细节 (二)

- 张量. 可以用 `tensor` 宏包.

`\tensor{\Lambda}{^\mu_\sigma}` 而不是  
`\Lambda^\mu_\sigma`:

$$\Lambda^\mu_\sigma \neq \Lambda^\mu_\sigma.$$

- 超大根式可以用 `^{p/q}` 之类描述:

$$\sqrt[3]{\sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2}} \quad \left( \sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2} \right)^{1/3}.$$

## 数学环境的细节 (二)

- 张量. 可以用 `tensor` 宏包.

`\tensor{\Lambda}{^\mu_\sigma}` 而不是  
`\Lambda^\mu_\sigma`:

$$\Lambda^\mu_\sigma \neq \Lambda^\mu_\sigma.$$

- 超大根式可以用 `^{p/q}` 之类描述:

$$\sqrt[3]{\sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2}} \quad \left( \sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2} \right)^{1/3}.$$

- 分式 (`\frac{分子}{分母}`). 在行间或角标请使用 `a/b`. 你也不想遇到这种吧:

## 数学环境的细节 (二)

- 张量. 可以用 `tensor` 宏包.

`\tensor{\Lambda}{^\mu_\sigma}` 而不是  
`\Lambda^\mu_\sigma`:

$$\Lambda^\mu_\sigma \neq \Lambda^\mu_\sigma.$$

- 超大根式可以用 `^{p/q}` 之类描述:

$$\sqrt[3]{\sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2}} \quad \left( \sum_{i=1}^N \frac{1}{(p_i + p'_i)^2 - m^2} \right)^{1/3}.$$

- 分式 (`\frac{分子}{分母}`). 在行间或角标请使用 `a/b`. 你也不想遇到这种吧:

$$\left(x + \frac{e}{e-1}\right) e^{\frac{x e^{\frac{e}{e-1}}}{e-1}}, \quad \bar{E} = \frac{\int_0^\infty E \exp(-E/(k_B T)) dE}{\int_0^\infty \exp(-E/(k_B T)) dE} = k_B T.$$

$$\left(x+\frac{e}{e-1}\right)\exp\Bigl(\frac{x^{e/(e-1)}}{e-1}\Bigr).$$

$$\overline{E}=\int_0^\infty E \exp\Bigl(-\frac{E}{k_{\rm B}T}\Bigr){\rm d}E\,\Bigg/\int_0^\infty \exp\Bigl(-\frac{E}{k_{\rm B}T}\Bigr){\rm d}E=k_{\rm B}T.$$

## 数学环境的细节 (三)

- 数学环境下的空格 (还记得吗? 数学环境下一般空格无效)<sup>1</sup>. 细 (\,)、中 (\:, \>)、粗 (\;): \quad, \qquad:

\,  $D X$

\>  $D X$

\;  $D X$

\quad  $D X$

\qquad  $D X$

<sup>1</sup>值得一提的是, 在 2020-10-01 以后, 或者调用 amsmath 后, 也可以在文本模式下使用这些命令.

## 数学环境的细节 (三)

- 数学环境下的空格 (还记得吗? 数学环境下一般空格无效)<sup>1</sup>.  
细 (\,)、中 (\:, \>)、粗 (\;): \quad, \quadquad:

\,       $D X$

\>       $D X$

\;       $D X$

\quad       $D X$

\quadquad       $D X$

- 不建议在行内插入过于复杂的公式.

<sup>1</sup>值得一提的是, 在 2020-10-01 以后, 或者调用 amsmath 后, 也可以在文本模式下使用这些命令.

# 数学字体的切换

- 数学字体的切换. 最好选择 amsfonts 宏包, 更现代的选择是 unicode-math 宏包.
  - 主要处理:  $\mathsf{math}$ ...{} 和  $\mathsf{sym}$ ...{} (unicode-math 宏包);

# 数学字体的切换

- 数学字体的切换. 最好选择 amsfonts 宏包, 更现代的选择是 unicode-math 宏包.
  - 主要处理:  $\mathsf{math}$ ...{} 和  $\mathsf{sym}$ ...{} (unicode-math 宏包);
  - 粗体:  $\mathbf{mathbf}$ , ( $\mathsf{sym}$ ) $\mathbf{bfit}$ ,  $\mathbf{boldsymbol}$ ;
  - 花体:  $\mathcal{mathcal}$ ,  $\mathscr{mathscr}$  (需要一些花体宏包, 如  $\mathcal{mathalpha}$ );
  - 哥特体和黑板粗体:  $\mathfrak{mathfrak}$ ,  $\mathbb{mathbb}$ .
  - 但是, 数学字体这些东西并不是从天上掉下来的. 正如前文所言, 数学环境调用的字体与正文**不同**, 因此不受正文字体环境影响.

# 数学字体的切换

- 数学字体的切换. 最好选择 amsfonts 宏包, 更现代的选择是 unicode-math 宏包.
  - 主要处理:  $\mathsf{math}$ ...{} 和  $\mathsf{sym}$ ...{} (unicode-math 宏包);
  - 粗体:  $\mathbf{mathbf}$ ,  $(\mathsf{sym})\mathbf{fit}$ ,  $\mathbf{boldsymbol}$ ;
  - 花体:  $\mathcal{mathcal}$ ,  $\mathscr{mathscr}$  (需要一些花体宏包, 如  $\mathcal{mathalpha}$ );
  - 哥特体和黑板粗体:  $\mathfrak{mathfrak}$ ,  $\mathbb{mathbb}$ .
  - 但是, 数学字体这些东西并不是从天上掉下来的. 正如前文所言, 数学环境调用的字体与正文**不同**, 因此不受正文字体环境影响.
  - Times New Roman: 使用 newtx 宏包.

# 更高更妙的物理 (一)

- Dirac 符号 (`physics2` 宏包):

$$A^\dagger = \sum_i \lambda_i^* |u_i\rangle\langle v_i|.$$

(有人曾经用  $|\phi\rangle$  来表示  $|\phi\rangle$ , 我不说是谁)

- Dirac 符号 (`physics2` 宏包):

$$A^\dagger = \sum_i \lambda_i^* |u_i\rangle\langle v_i|.$$

(有人曾经用  $|\phi\rangle$  来表示  $|\phi\rangle$ , 我不说是谁)

- 合理的单位制 (`siunitx` 宏包).

- `\hbar=1.0545\times 10^{-34}\,\text{J}\cdot\text{s}`:  $\hbar = 1.0545 \times 10^{-34} J \cdot s.$
- `\hbar=\qty[options...]{1.0545e-34}{\text{J}\cdot\text{s}}`:  
 $\hbar = 1.0545 \times 10^{-34} J \cdot s.$

# 更高更妙的物理 (一)

- Dirac 符号 (`physics2` 宏包):

$$A^\dagger = \sum_i \lambda_i^* |u_i\rangle\langle v_i|.$$

(有人曾经用  $|\phi\rangle$  来表示  $|\phi\rangle$ , 我不说是谁)

- 合理的单位制 (`siunitx` 宏包).

- $\hbar=1.0545\times 10^{-34}\text{J}\cdot\text{s}$ :  $\hbar = 1.0545 \times 10^{-34} \text{ J} \cdot \text{s}$ .
- $\hbar=\text{qty}[options...]{1.0545e-34}{\text{J.s}}$ :  
 $\hbar = 1.0545 \times 10^{-34} \text{ J} \cdot \text{s}$ .

- 微分算子分不清? 是  $\mathcal{D}$ ,  $d$  还是  $\mathrm{d}$  呢?

- 古老的国标 GB/T 3012.11-93 成为推荐性, 用什么都是可以哒!
- `fixdif` 宏包, 更方便考虑间距:  $\int^d \frac{\mathrm{d}D(d)}{\mathrm{d}d} \mathrm{d}d = D(d) + C$ .

## 更高更妙的物理 (二)

- Feynman 图 (tikz-feynman 宏包).

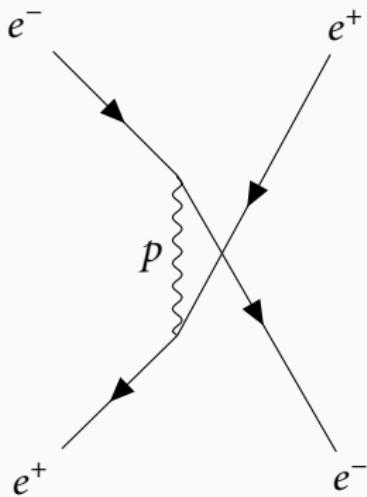


图 2: u-channel

$\mathbf{bm} = \text{\boldsymbol{...}}$ 、 $\mathcal{cal} = \text{\mathcal{...}}$ , 以此类推.

rm	bf	sf	$\mathbf{bm} + \mathsf{sf}$
	$\mathbf{bm}$	-	-
cal	$\mathbf{bm} + \mathcal{cal}$	bb	-
scr	-	frak	$\mathbf{bm} + \mathfrak{frak}$

D	D	D	D	x	x	x	x
D	D	D	D	x	x	x	x
$\mathcal{D}$	$\mathfrak{D}$	$\mathbb{D}$	$\mathbb{D}$	x	x	$\mathbb{x}$	$\mathbb{x}$
$\mathcal{D}$	$\mathfrak{D}$	$\mathfrak{D}$	$\mathfrak{D}$	x	x	$\mathfrak{x}$	$\mathfrak{x}$

## 多行公式与公式对齐 (一)

align 环境或 gather 环境. 在调用 amsmath 宏包的情形下, 也可以用 aligned 和 gathered. 同矩阵一样, & 分列, \\ 对齐, 用 \tag 和 \notag 控制标号.

## 多行公式与公式对齐 (一)

align 环境或 gather 环境. 在调用 amsmath 宏包的情形下, 也可以用 aligned 和 gathered. 同矩阵一样, & 分列, \\ 对齐, 用 \tag 和 \notag 控制标号.

```
\begin{align}
\hbar = c &= 1 && \\
\notag & & & \\
\tag{42} & & & \hat{\delta}.
\end{align}
```

## 多行公式与公式对齐 (一)

align 环境或 gather 环境. 在调用 amsmath 宏包的情形下, 也可以用 aligned 和 gathered. 同矩阵一样, & 分列, \\ 对齐, 用 \tag 和 \notag 控制标号.

$$\begin{aligned} \hbar &= c &= 1 && \\\notag && && \\ \tag{42} \hbar &= 137\alpha & & & \\ \end{aligned} \quad \begin{aligned} \hbar &= c = 1 & (1) \\ &= 137\alpha \\ &= \hat{\delta}. & (42) \end{aligned}$$

## 多行公式与公式对齐 (一)

align 环境或 gather 环境. 在调用 amsmath 宏包的情形下, 也可以用 aligned 和 gathered. 同矩阵一样, & 分列, \\ 对齐, 用 \tag 和 \notag 控制标号.

```
\begin{align}
\hbar = c &= 1 && \\
\notag && &\\
\tag{42} && \hbar = c = 1 & (1) \\
&& &= 137\alpha \\
\end{align}
```

=  $\hat{\delta}$ . (42)

想要去掉所有标号可以用共轭算符 \* 环境:

```
\begin{gather*}
p(E = n\hbar\omega) =
\frac{e^{-\beta n\hbar\omega}}{Z}; \\
Z = \frac{1}{1 - e^{-\beta\hbar\omega}}.
\end{gather*}
```

## 多行公式与公式对齐 (一)

align 环境或 gather 环境. 在调用 amsmath 宏包的情形下, 也可以用 aligned 和 gathered. 同矩阵一样, & 分列, \\ 对齐, 用 \tag 和 \notag 控制标号.

$$\begin{aligned} \hbar &= c &= 1 && \\\notag && && \\ \tag{42} \hbar &= 137\alpha & & & \\ \end{aligned} \quad \begin{aligned} \hbar &= c = 1 & (1) \\ &= 137\alpha & \\ &= \hat{\delta}. & (42) \end{aligned}$$

想要去掉所有标号可以用共轭算符 \* 环境:

$$\begin{aligned} \begin{gathered} p(E = n\hbar\omega) = \\ \frac{e^{-\beta n\hbar\omega}}{Z}; \\ Z = \frac{1}{1 - e^{-\beta\hbar\omega}}. \end{gathered} \quad \begin{aligned} p(E = n\hbar\omega) &= \frac{e^{-\beta n\hbar\omega}}{Z}; \\ Z &= \frac{1}{1 - e^{-\beta\hbar\omega}}. \end{aligned} \end{aligned}$$

## 多行公式与公式对齐 (二)

当多行公式只需一个标号时, 可以用 `equation` 套上 `aligned` 或 `gathered` 环境解决:

```
\begin{equation}
\begin{aligned}
\partial_t p &= -\partial_x H, \\
d_t x &= \partial_p H.
\end{aligned}
\end{equation}
```

## 多行公式与公式对齐 (二)

当多行公式只需一个标号时, 可以用 `equation` 套上 `aligned` 或 `gathered` 环境解决:

```
\begin{equation}
\begin{aligned}
\partial_t p &= -\partial_x H, \\
d_t x &= \partial_p H.
\end{aligned}
\end{equation}
```

$$\begin{aligned} \partial_t p &= -\partial_x H, \\ d_t x &= \partial_p H. \end{aligned} \quad (2)$$

## 多行公式与公式对齐 (二)

当多行公式只需一个标号时, 可以用 `equation` 套上 `aligned` 或 `gathered` 环境解决:

```
\begin{equation}
\begin{aligned}
\partial_t p &= -\partial_x H, \\
d_t x &= \partial_p H.
\end{aligned}
\end{equation}
```

(2)

请不要使用 `\[ ... \]` `\[ ... \]` `\[ ... \]`! 更多需求可以使用 `alignat` 和 `flalign` 等, 总有一款适合你.

## 多行公式与公式对齐 (三)

如果要对很多子公式进行主-子编号，可以用 `subequations` 环境  
搭配 `align` 和 `gather` 使用：

## 多行公式与公式对齐 (三)

如果要对很多子公式进行主-子编号, 可以用 `subequations` 环境  
搭配 `align` 和 `gather` 使用:

```
\begin{subequations}
  \begin{gather}
    \mathcal{H}_{\text{ac}} \colon= \\
    \{x \in \mathcal{H} \mid \mu_x \leq m\}, \\
    \dots \\
  \end{gather}
\end{subequations}
```

## 多行公式与公式对齐 (三)

如果要对很多子公式进行主-子编号, 可以用 `subequations` 环境搭配 `align` 和 `gather` 使用:

```
\begin{subequations}
  \begin{gather}
    \mathcal{H}_{\text{ac}} \colon= \{x \in \mathcal{H} \mid \mu_x \ll m\}, \\
    \mathcal{H}_{\text{sc}} \colon= \{x \in \mathcal{H} \mid \mu_{x,x} \perp m \text{ but } \mu_{x,x}(\{\text{pt}\}) = 0\}, \\
    \mathcal{H}_{\text{pp}} \colon= \{x \in \mathcal{H} \mid \lambda \mapsto \langle E_\lambda x, x \rangle \text{ is discrete}\},
  \end{gather}
\end{subequations}
```

$$\mathcal{H}_{\text{ac}} \colon= \{x \in \mathcal{H} \mid \mu_x \ll m\}, \tag{3a}$$

$$\mathcal{H}_{\text{sc}} \colon= \{x \in \mathcal{H} \mid \mu_{x,x} \perp m \text{ but } \mu_{x,x}(\{\text{pt}\}) = 0\}, \tag{3b}$$

$$\mathcal{H}_{\text{pp}} \colon= \{x \in \mathcal{H} \mid \lambda \mapsto \langle E_\lambda x, x \rangle \text{ is discrete}\}, \tag{3c}$$

$$\mathcal{H} = \mathcal{H}_{\text{ac}} \oplus \mathcal{H}_{\text{sc}} \oplus \mathcal{H}_{\text{pp}}. \tag{3d}$$

# 符号

- 明确一点，大多数（抽象的）符号都只是编码，也即：

# 符号

- 明确一点，大多数（抽象的）符号都只是编码，也即：
  - U+1D510: ℮;
  - U+02B32: Ⓜ;
  - U+060A8: 您;
  - .....

具体显示效果来源于**字体**，不同的字体对同样的符号显示效果可能是不一样的。

- 明确一点，大多数（抽象的）符号都只是编码，也即：
  - U+1D510:  $\mathfrak{M}$ ;
  - U+02B32:  $\leftrightarrow$ ;
  - U+060A8: 您;
  - .....

具体显示效果来源于**字体**，不同的字体对同样的符号显示效果可能是不一样的。

- 因此「符号」不可能只用一个按钮或者一个命令就可以从天而降：需要字体；需要 L<sup>A</sup>T<sub>E</sub>X 调用字体；需要语义化的命令而不是直接输入编码得到符号……

# 如何寻找符号?

- 由 T<sub>E</sub>X 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 

# 如何寻找符号？

- 由 TeX 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 
- unicode-math 宏包使用者: 万有 *Unimath* 

# 如何寻找符号?

- 由 T<sub>E</sub>X 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 
- unicode-math 宏包使用者: 万有 *Unimath* 
- 仅仅知道形状? *Detexify* 

# 如何寻找符号?

- 由 TeX 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 
- unicode-math 宏包使用者: 万有 *Unimath* 
- 仅仅知道形状? *Detexify* 
- 我要苹果商标 ! 这个字形没有收录在 Unicode 标准中, 换言之, 只有特殊的字体才会有 (亦即, 在字体的私有区中制作), 可见 fontawesome5 或者 simpleicons 宏包;

# 如何寻找符号?

- 由 TeX 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 
- unicode-math 宏包使用者: 万有 *Unimath* 
- 仅仅知道形状? *Detexify* 
- 我要苹果商标 ! 这个字形没有收录在 Unicode 标准中, 换言之, 只有特殊的字体才会有 (亦即, 在字体的私有区中制作), 可见 fontawesome5 或者 simpleicons 宏包;
- emoji ()? emoji 也有对应的 Unicode 规范, 但 X<sub>H</sub>TeX 对彩色字体支持性不高, 可以用 LuaTeX 或者 twemoji 宏包;

# 如何寻找符号?

- 由 TeX 宏包所定义的符号: *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List* 
- unicode-math 宏包使用者: 万有 *Unimath* 
- 仅仅知道形状? *Detexify* 
- 我要苹果商标 ! 这个字形没有收录在 Unicode 标准中, 换言之, 只有特殊的字体才会有 (亦即, 在字体的私有区中制作), 可见 fontawesome5 或者 simpleicons 宏包;
- emoji ()? emoji 也有对应的 Unicode 规范, 但 X<sub>H</sub>TeX 对彩色字体支持性不高, 可以用 LuaTeX 或者 twemoji 宏包;
- 用 \%、\\$、\&、\textbackslash 得到 %、\$、&、\;

# 文档结构

---

内容与表现分离 (*separation of content and presentation*) 原则的本意是「文档的实际内容、逻辑结构」与「文档呈现给读者的样式」是相互独立的。并且，「内容与样式分离」是一种通用的原则，而并非只属于 L<sup>A</sup>T<sub>E</sub>X.

# 一个例子

```
\documentclass{article} % 导言区. 设置文档样式.  
\usepackage{amsmath}  
\usepackage{fixdif} % 正体微分算子  $d$ .  
\newcommand{\key}[1]{\textbf{#1}} % 定义新命令.  
\begin{document}  
    \section{The Spectral Theorem} % 小节标题.  
    If  $\mu$  is a  $\text{projection-valued measure}$  on  $(X, \Omega)$  with  
    values in  $\mathsf{B}(\mathcal{H})$ ,  $\psi$  is an element of  $\mathcal{H}$ , then we can construct a positive real-valued  
    measure  $\mu_\psi$  from  $\mu$  by setting  $\mu_\psi(E) = \langle \psi, \mu(E)\psi \rangle$ , for each measurable set  $E$ .  
  
    To motivate the following definition, consider integration of a  
    \emph{bounded} measurable function  $f$  against a projection-  
    valued measure  $\mu$ . Since the integral is multiplicative and  
    complex-conjugation of a function corresponds to adjoint of the  
    operator, we have  
    
$$[\biggl| \Bigl( \int_X f d \mu \Bigr) \psi \biggr|^2 = \int_X |f|^2 d \mu_\psi.]$$
  
\end{document}
```

# The Spectral Theorem

If  $\mu$  is a **projection-valued measure** on  $(X, \Omega)$  with values in  $B(\mathcal{H})$ ,  $\psi$  is an element of  $\mathcal{H}$ , then we can construct a positive real-valued measure  $\mu_\psi$  from  $\mu$  by setting  $\mu_\psi(E) = \langle \psi, \mu(E)\psi \rangle$ , for each measurable set  $E$ .

To motivate the following definition, consider integration of a *bounded* measurable function  $f$  against a projection-valued measure  $\mu$ . Since the integral is multiplicative and complex-conjugation of a function corresponds to adjoint of the operator, we have

$$\left\| \left( \int_X f d\mu \right) \psi \right\|^2 = \int_X |f|^2 d\mu_\psi.$$

文本来源: BRIAN C. HALL. *Quantum Theory for Mathematicians*.

# 布局：从零 (一)

L<sup>A</sup>T<sub>E</sub>X 给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

## 布局：从零 (一)

LATEX给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

- 标题。事实上包括标题还有作者的名字，日期，所属机构等等。或者更广泛地，扉页和版权页：`\title`, `\author`, `\date`, `\maketitle`;

## 布局：从零 (一)

LATEX给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

- 标题。事实上包括标题还有作者的名字，日期，所属机构等等。或者更广泛地，扉页和版权页：`\title`, `\author`, `\date`, `\maketitle`;
- 摘要、致谢与声明。与这些辅文类似的，还有序、前言之类：`\begin{abstract} ... \end{abstract}`;

# 布局：从零 (一)

LATEX 给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

- 标题。事实上包括标题还有作者的名字，日期，所属机构等等。或者更广泛地，扉页和版权页：`\title`, `\author`, `\date`, `\maketitle`;
- 摘要、致谢与声明。与这些辅文类似的，还有序、前言之类：`\begin{abstract} ... \end{abstract}`;
- 目录。`\tableofcontents`;

# 布局：从零（一）

LATEX给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

- 标题。事实上包括标题还有作者的名字，日期，所属机构等等。或者更广泛地，扉页和版权页：`\title`, `\author`, `\date`, `\maketitle`;
- 摘要、致谢与声明。与这些辅文类似的，还有序、前言之类：`\begin{abstract} ... \end{abstract}`;
- 目录。`\tableofcontents`;
- 你的正文。大致分`\chapter`, `\section`, `\subsection`, `\paragraph`等；

# 布局：从零（一）

LATEX 给了我们这样一个机会去思考一篇文章里面到底有什么，事实上，简单的文章大概都会包含以下这些：

- 标题。事实上包括标题还有作者的名字，日期，所属机构等等。或者更广泛地，扉页和版权页：`\title`, `\author`, `\date`, `\maketitle`;
- 摘要、致谢与声明。与这些辅文类似的，还有序、前言之类：`\begin{abstract} ... \end{abstract}`;
- 目录。`\tableofcontents`;
- 你的正文。大致分 `\chapter`, `\section`, `\subsection`, `\paragraph` 等;
- 参考文献。`\bibliography` 与 `\cite`.

## 布局：从零 (二)

- 在真正写东西之前，了解自己需要写什么：article? book? 亦或是只用来做极小测试的 minimal? 在想清楚这点之后，我们就可以选取所需的**文档类**了。

## 布局：从零 (二)

- 在真正写东西之前，了解自己需要写什么：article? book? 亦或是只用来做极小测试的 minimal? 在想清楚这点之后，我们就可以选取所需的**文档类**了。
- 文档类用 `\documentclass[选项]{名称}` 在引言区调用，三大标准类：
  - article: 用于科技论文、报告、说明文档等；
  - report: 具有**章节结构**，用于综述、长篇论文、简单书籍等；
  - book: 包含章节结构和前言、正文、后记等结构。

## 布局：从零 (二)

- 在真正写东西之前，了解自己需要写什么：article? book? 亦或是只用来做极小测试的 minimal? 在想清楚这点之后，我们就可以选取所需的**文档类**了。
- 文档类用 \documentclass[选项]{名称} 在引言区调用，三大标准类：
  - article: 用于科技论文、报告、说明文档等；
  - report: 具有**章节结构**，用于综述、长篇论文、简单书籍等；
  - book: 包含章节结构和前言、正文、后记等结构。

因此，article 没有定义 \chapter 命令合情合理。在科技论文中，如果要将文章分割成多个超大板块，可以用 \part.

# 布局：文本标记

我要加粗！我要楷体！我要黑体 和  
居中中中中中

右对齐！我还要：

——甲方

# 布局：文本标记

我要加粗！我要楷体！我要黑体 和  
居中中中中中

右对齐！我还要：

——甲方

- 如果不管格式与内容分离的原则（或者懒），可以用
  - 粗体：`\textbf{...}`、`\bfseries ...`；
  - 黑体：`\textsf{...}`、`\sfseries ...`；
  - 意大利体（斜体）：`\textit{...}`、`\itshape ...`；
  - 控制大小：`\tiny`、`\small`、`\normalsize`、`\large`、`\Huge`。

# 布局：文本标记

我要加粗！我要楷体！我要黑体 和  
居中中中中中

右对齐！我还要：

——甲方

- 如果不管格式与内容分离的原则（或者懒），可以用
  - 粗体：`\textbf{...}`、`\bfseries ...`；
  - 黑体：`\textsf{...}`、`\sffamily ...`；
  - 意大利体（斜体）：`\textit{...}`、`\itshape ...`；
  - 控制大小：`\tiny`、`\small`、`\normalsize`、`\large`、`\Huge`.
- 但是为了避免别人诟病，你可以（新定义）`\keyword{...}`代替`\textbf{...}`、`\emph{...}`代替`\textit{...}`；

# 布局：文本标记

我要加粗！我要楷体！我要黑体 和  
居中中中中中

右对齐！我还要：

——甲方

- 如果不管格式与内容分离的原则（或者懒），可以用
  - 粗体：`\textbf{...}`、`\bfseries ...`；
  - 黑体：`\textsf{...}`、`\sffamily ...`；
  - 意大利体（斜体）：`\textit{...}`、`\itshape ...`；
  - 控制大小：`\tiny`、`\small`、`\normalsize`、`\large`、`\Huge`。
- 但是为了避免别人诟病，你可以（新定义）`\keyword{...}`代替`\textbf{...}`、`\emph{...}`代替`\textit{...}`；
- 折行是`\backslash`，但这只是折行，分段用`\par`或者直接空行；

# 布局：文本标记

我要加粗！我要楷体！我要黑体 和  
居中中中中中

右对齐！我还要：

——甲方

- 如果不管格式与内容分离的原则（或者懒），可以用
  - 粗体：`\textbf{...}`、`\bfseries ...`；
  - 黑体：`\textsf{...}`、`\sffamily ...`；
  - 意大利体（斜体）：`\textit{...}`、`\itshape ...`；
  - 控制大小：`\tiny`、`\small`、`\normalsize`、`\large`、`\Huge`.
- 但是为了避免别人诟病，你可以（新定义）`\keyword{...}` 代替 `\textbf{...}`、`\emph{...}` 代替 `\textit{...}`；
- 折行是 `\backslash`，但这只是折行，分段用 `\par` 或者直接空行；
- 居中可以用 `\centering` 和 `center` 环境。当然在一般的文献中，居中一般用于摘要，你也可以用 `abstract` 环境。

# 另一个例子

## 例子

公理 1 (自然单位). 上帝曾言,  $\hbar = c = 1$ .

- 粗体 → 「公理 1」 → 取消粗体 → 「(自然单位)」 → 粗体  
→ 句点 → 取消粗体 → 楷体 (斜体) → 「上帝曾言,」 →  
公式开始 → 「 $\hbar = c = 1$ 」 → 公式结束 → 句点;

# 另一个例子

## 例子

公理 1 (自然单位). 上帝曾言,  $\hbar = c = 1$ .

- 粗体 → 「公理 1」 → 取消粗体 → 「(自然单位)」 → 粗体  
→ 句点 → 取消粗体 → 楷体(斜体) → 「上帝曾言,」 →  
公式开始 → 「 $\hbar = c = 1$ 」 → 公式结束 → 句点;
- 不应该思考「这一小块内容应该是加粗、倾斜、字号几何」等  
问题;

# 另一个例子

## 例子

公理 1 (自然单位). 上帝曾言,  $\hbar = c = 1$ .

- 粗体 → 「公理 1」 → 取消粗体 → 「(自然单位)」 → 粗体  
→ 句点 → 取消粗体 → 楷体(斜体) → 「上帝曾言,」 →  
公式开始 → 「 $\hbar = c = 1$ 」 → 公式结束 → 句点;
- 不应该思考「这一小块内容应该是加粗、倾斜、字号几何」等  
问题;
- 这是「定理」 → 它的名字是「勾股定理」 → 它的内容是「设  
直角三角形……」 → 把它们分别放在正确环境中.

# 另一个例子

## 例子

公理 1 (自然单位). 上帝曾言,  $\hbar = c = 1$ .

- 粗体 → 「公理 1」 → 取消粗体 → 「(自然单位)」 → 粗体 → 句点 → 取消粗体 → 楷体(斜体) → 「上帝曾言,」 → 公式开始 → 「 $\hbar = c = 1$ 」 → 公式结束 → 句点;
- 不应该思考「这一小块内容应该是加粗、倾斜、字号几何」等问题;
- 这是「定理」 → 它的名字是「勾股定理」 → 它的内容是「设直角三角形……」 → 把它们分别放在正确环境中.
- 如果写作的过程中会需要你不停调整格式, 这意味着你喜欢调整格式你应该切换思考方式, 写东西只需要一个框架, 一切完工后再慢慢设计不迟.

# 布局：浮动体不是**不动体**

- 动了吗？如动。 ——古谚

# 布局：浮动体不是不动体

- 动了吗？如动。 ——古谚
- 主要的浮动体是 `figure` 和 `table` 环境：
  - 标题用 `\caption`；
  - 标签用 `\label`；
  - 引用用 `\ref` 或者 `\autoref`（需要 `hyperref` 宏包）；

# 布局：浮动体不是不动体

- 动了吗？如动。 ——古谚
- 主要的浮动体是 `figure` 和 `table` 环境：
  - 标题用 `\caption`；
  - 标签用 `\label`；
  - 引用用 `\ref` 或者 `\autoref`（需要 `hyperref` 宏包）；
- 浮动体是会动的，**故意的**。
  - 治本型：不要用「上图下表」；
  - 眼不见心不烦型：先写文章，全部写完后再插图片；
  - 摆烂型：所有图片全部放最后面；
  - 实用主义型：不就是要个编号吗？我手动加上去就是了：`caption` 宏包。（不是很推荐）

# 布局：浮动体不是不动体

- 动了吗？如动。 ——古谚
- 主要的浮动体是 `figure` 和 `table` 环境：
  - 标题用 `\caption`；
  - 标签用 `\label`；
  - 引用用 `\ref` 或者 `\autoref`（需要 `hyperref` 宏包）；
- 浮动体是会动的，**故意的**。
  - 治本型：不要用「上图下表」；
  - 眼不见心不烦型：先写文章，全部写完后再插图片；
  - 摆烂型：所有图片全部放最后面；
  - 实用主义型：不就是要个编号吗？我手动加上去就是了：`caption` 宏包。（不是很推荐）
- 归根结底就是，`LATEX` 是不会急的，所以请你也别急。

## 布局：参考文献

- 我们一开始给出了 BiBTeX 的大致使用方式，更现代的方法可以使用 biber 后端配合 biblatex 宏包；

```
% 引言区
\usepackage[backend=biber, style=gb7714-2015]{
    biblatex}
\addbibresource[location=local]{cite.bib}
% \begin{document} 之后
\printbibliography[heading=bibliography, title=参考
    文献]
```

然后分多步编译：

```
xelatex jobname.tex
biber jobname
xelatex jobname.tex
xelatex jobname.tex
```

## Debugging

---

先花几小看完 `lshort-zh-cn.pdf` (当睡前读物看也行, 附录也很重要), 初学者遇到的问题大多数是 `lshort-zh-cn.pdf` 可以搞定的, 即使读文档对不读文档的人来说可能是让他们的学习曲线变得陡峭的重要环节, 或者也可以加入 LaTeX 工作室的交流群.

# 命令行

---

小心编辑器帮你过滤信息（比如 VS Code 和 LaTeX Workshop）导致用户没看到完整的报错。

- 编辑器按钮要用，命令行操作也要用，这样才算得上健全。

```
xelatex -options "your_name.tex";
```

# 命令行

小心编辑器帮你过滤信息（比如 VS Code 和 LaTeX Workshop）导致用户没看到完整的报错。

- 编辑器按钮要用，命令行操作也要用，这样才算得上健全。  
`xelatex -options "your_name.tex";`
- 想要一次性编译多次？`latexmk`；

小心编辑器帮你过滤信息（比如 VS Code 和 LaTeX Workshop）导致用户没看到完整的报错。

- 编辑器按钮要用，命令行操作也要用，这样才算得上健全。  
`xelatex -options "your_name.tex";`
- 想要一次性编译多次？`latexmk`；
- 宏包的文档？`texdoc pkg_name`；
  - 我没有安装本地发行版：
  - 不想看英文文档？

小心编辑器帮你过滤信息（比如 VS Code 和 LaTeX Workshop）导致用户没看到完整的报错。

- 编辑器按钮要用，命令行操作也要用，这样才算得上健全。  
`xelatex -options "your_name.tex";`
- 想要一次性编译多次？`latexmk`；
- 宏包的文档？`texdoc pkg_name`；
  - 我没有安装本地发行版：`\o{}`；
  - 不想看英文文档？`\o{}`.
- 想要看命令是怎么被定义的？`(la)texdef`、`\show`、`\meaning` 以及 `show2e` 宏包；

# 命令行

小心编辑器帮你过滤信息（比如 VS Code 和 LaTeX Workshop）导致用户没看到完整的报错。

- 编辑器按钮要用，命令行操作也要用，这样才算得上健全。  
`xelatex -options "your_name.tex";`
- 想要一次性编译多次？`latexmk`；
- 宏包的文档？`texdoc pkg_name`；
  - 我没有安装本地发行版：
  - 不想看英文文档：
- 想要看命令是怎么被定义的？`(la)texdef`、`\show`、`\meaning` 以及 `show2e` 宏包；
- 包管理器：`tlmgr`；

# 遇到问题之后（一）

出现问题了——

- 我换了好几个模板都出现了问题，因此我把他们所有正常的代码全部放在一块，总共有 500 kB；
- 再按几次编译按钮，直到编译成功为止；
- 去论坛提问“`latex` 报错为什么？”.

# 遇到问题之后（一）

出现问题了——

- 我换了好几个模板都出现了问题，因此我把他们所有正常的代码全部放在一块，总共有 500 kB；
- 再按几次编译按钮，直到编译成功为止；
- 去论坛提问“`latex 报错为什么？`”.

更合理的办法：

- 先把所有辅助文件删掉，然后编译试试；
- 二分法排查代码，看看能不能用最小的文档复刻报错；
  - 花里胡哨的叫法：最小工作示例（MWE, minimal working example）；
  - 不仅自己可以看清楚些，如果走投无路了想要发论坛求救用更少的代码复现问题会更好.

## 遇到问题之后（二）

- 看看报错日志：

```
Runaway argument?  
\BIBentryALTinterwordspacing W.\~Commons,  
``File hyperspectralcube.jpg \ETC.  
! Paragraph ended before \BR@c@bibitem was complete.  
<to be read again>  
\par
```

Runaway argument? 是报错类型，  
\BIBentryALTinterwordspacing... \ETC. 是报错的上下文， ! Paragraph...complete. 是报错原因。

## 遇到问题之后（二）

- 看看报错日志：

```
Runaway argument?  
\BIBentryALTinterwordspacing W.\~Commons,  
``File hyperspectralcube.jpg \ETC.  
! Paragraph ended before \BR@c@bibitem was complete.  
<to be read again>  
\par
```

Runaway argument? 是报错类型，

\BIBentryALTinterwordspacing... \ETC. 是报错的上下文， ! Paragraph...complete. 是报错原因。

- 先在 `lshort-zh-cn.pdf` 附录里找有没有适合自己的解决方法，或者在 Google、论坛上搜索报错原因，发给 ChatGPT 也行。

## 遇到问题之后 (三)

- 如果 Google、ChatGPT 都不管用，那你就去尝试去论坛提问了！

## 遇到问题之后 (三)

- 如果 Google、ChatGPT 都不管用，那你就去尝试去论坛提问了！
- 常用论坛：CT<sub>E</sub>X 临时论坛 , Stack Exchange , 知乎 ;

## 遇到问题之后（三）

- 如果 Google、ChatGPT 都不管用，那你就去尝试去论坛提问了！
- 常用论坛：CT<sub>E</sub>X 临时论坛 , Stack Exchange , 知乎 ;
- 提问的姿势水平 ：
  - 提供最小工作示例；
  - 使用描述性语句，谨慎使用（自创的排版）术语，解释涉及具体学科的术语；
  - 除非是付费服务，否则不能要求得到及时的、满意的回应；
  - 提供准确而详细的信息：
    - 介绍自己在做什么、为什么要这么做；
    - 介绍自己是怎么做的，让问题能在他人电脑上复现.
  - **别急**，急也没用.

# 参考文献

- [1] DONALD E. KNUTH. **The TeXbook: Computers & Typesetting, volume C** [M], 1984.  
**Addison-Wesley Publishing Company, Boston**
- [2] 刘海洋. **LATeX 入门** [M], 2013.  
北京: 电子工业出版社
- [3] 曾祥东. 现代 **LATeX** 入门讲座 [EB/OL], 2022.  
<https://github.com/stone-zeng/latex-talk>
- [4] 张庭瑄. **LATeX 新手上路指南** [EB/OL], 2022.  
<https://github.com/AlphaZTX/LaTeX-tutorials>
- [5] 黄晨成. 一份其实很短的 **LATeX** 入门文档 [EB/OL], 2022.  
<https://liam.page/2014/09/08/latex-introduction>
- [6] OETIKER T, PARTL H, HYNA I and SCHLEGL E. **CTeX** 开发小组 译. 一份(不太)简短的 **LATeX 2 $\epsilon$**  介绍: 或 **111 分钟了解 LATeX 2 $\epsilon$**  [EB/OL], 2023.  
<https://ctan.org/pkg/lshort-zh-cn>
- [7] **LATeX project. LearnLATeX.org** [EB/OL].  
<https://www.learnlatex.org>
- [8] The Type. **孔雀计划: 中文字体排印的思路** [EB/OL].  
<https://www.thetype.com/kongque>

私货

---

- Typst：
  - Typst 很快；支持增量编译；
  - LATEX 用户惯性大（因为繁杂的缘故）；
  - 可以直接调用 CSV 文件——现代生产力水平；
  - 在据说要打败 LATEX 的产品中，生态最好的；
  - 现代的设计语言风格；
  - 用比较省力的方式排版出一个基本好看的文章；
  - 适合不满足于 Markdown 功能简陋、又无法接受 LATEX 过于冗杂的用户；
  - 上手容易不代表可以轻松用好它；
  - 静待期刊支持……

# Typst、M\$ Word 与 LATEX

- Typst：
  - Typst 很快；支持增量编译；
  - LATEX 用户惯性大（因为繁杂的缘故）；
  - 可以直接调用 CSV 文件——现代生产力水平；
  - 在据说要打败 LATEX 的产品中，生态最好的；
  - 现代的设计语言风格；
  - 用比较省力的方式排版出一个基本好看的文章；
  - 适合不满足于 Markdown 功能简陋、又无法接受 LATEX 过于冗杂的用户；
  - 上手容易不代表可以轻松用好它；
  - 静待期刊支持……
- M\$ Word：
  - Word 是字处理软件，而不是专门用来排版的；或者反过来，LATEX 不是字处理软件；
  - 至少上手容易，但同样，不代表可以轻松用好它；

- 为什么要有 LATEX3?
  - LATEX 2<sub>e</sub> 没有很好地实现内容与格式分离的原则；
  - 因为内核相对较小，当有格式修改需求时，经常遇到功能不足的问题，对格式进行更改需要大量搜索和编写复杂代码；
  - 就连新定义一些变长参数命令都很麻烦.

- 为什么要有 LATEX3?
  - LATEX 2<sub>e</sub> 没有很好地实现内容与格式分离的原则；
  - 因为内核相对较小，当有格式修改需求时，经常遇到功能不足的问题，对格式进行更改需要大量搜索和编写复杂代码；
  - 就连新定义一些变长参数命令都很麻烦.
- LATEX3 旨在——
  - 提供现代编程语言的语法与命名规范；
  - 简化 LATEX 的宏展开控制；
  - 标准化 LATEX 各个功能的接口；

- (目前的) 主要成果 expl3 宏包;
- 两个宏包：
  - 创建用户层命令: xparse 宏包 (默认载入);
  - 面向对象: xtemplate 宏包
- 一步到位: `texdoc interface3`;
- 懒得学? 用 Chat GPT!

- 当甲方的认为比写出良好的代码更紧急的时候……
- L<sup>A</sup>T<sub>E</sub>X Hooks!

## 现实案例

### Contents

1 Supersymmetry 1

2 ... 1

1 Supersymmetry

2 ...

### Contents

• 1 Supersymmetry 1

• 2 ... 1

1 Supersymmetry

2 ...

- 当甲方的认为比写出良好的代码更紧急的时候……
- L<sup>A</sup>T<sub>E</sub>X Hooks!

## 现实案例

### Contents

1	Supersymmetry	1
2	...	1

### 1 Supersymmetry

2 ...

### Contents

• 1	Supersymmetry	1
• 2	...	1

### 1 Supersymmetry

2 ...

- etoolbox 宏包. 偷偷修改 (补丁) 默认的定义 (?).

# 新闻：取所好者

- texdoc ltnews

# 新闻：取所好者

- `texdoc ltnews`
- LaTeX Companion 3 发售了；

# 新闻：取所好者

- `texdoc ltnews`
- LaTeX Companion 3 发售了；
- L<sup>A</sup>T<sub>E</sub>X Hooks. 拦截宏展开；

# 新闻: 取所好者

- `texdoc ltnews`
- LaTeX Companion 3 发售了;
- L<sup>A</sup>T<sub>E</sub>X Hooks. 拦截宏展开;
- 整数和浮点数基本计算默认调用: `\inteval`、`\fpeval`;

# 新闻：取所好者

- `texdoc ltnews`
- LaTeX Companion 3 发售了；
- L<sup>A</sup>T<sub>E</sub>X Hooks. 拦截宏展开；
- 整数和浮点数基本计算默认调用: `\inteval`、`\fpeval`；
- 命令和环境复制器. `\NewEnvironmentCopy` (暂未更新)、  
`\NewCommandCopy`.

# 彩色盒子

你想要

这样的盒子吗？

使用 `tcolorbox` 宏包！文档：`texdoc tcolorbox`；或者用  
`texdoc tcolorbox-example` 查看例子。

- 对中文使用者来说，一般使用 `fontspec` 宏包。
  - CTeX 宏集的策略大致是，将 Unicode 字符分成不同的类，每一类单独处理。比如：中文和英文用的就是不一样的字体。
  - 日本語。当作者意欲在中文内插入日文时，自然应该使用日文字体：

八坂神奈子 与 八坂神奈子

而不是使用中文字体，这大抵上是最容易出错的细节。
  - 繁体中文和韩语类似。大陆字体会做日文、韩文部分不代表可以用其来排日文、韩文。

- 对中文使用者来说，一般使用 `fontspec` 宏包。
  - CTeX 宏集的策略大致是，将 Unicode 字符分成不同的类，每一类单独处理。比如：中文和英文用的就是不一样的字体。
  - 日本語。当作者意欲在中文内插入日文时，自然应该使用日文字体：

八坂神奈子 与 八坂神奈子

而不是使用中文字体，这大抵上是最容易出错的细节。
  - 繁体中文和韩语类似。大陆字体会做日文、韩文部分不代表可以用其来排日文、韩文。
- 如果作者发给你的是 Word 文档，可以先好好沟通一下格式应当如何处理。

# 文字类排版: OpenType 特性

- 连字: ff → ff, fi → fi, fl → fl;

# 文字类排版: OpenType 特性

- 连字: ff → ff, fi → fi, fl → fl;
- 老式数字 (onum): 0123456789 → o123456789;

# 文字类排版: OpenType 特性

- 连字: ff → ff, fi → fi, fl → fl;
- 老式数字 (onum): 0123456789 → o123456789;
- 小写大写字母 (smcp): An Academic Fantasy For Minority → AN ACADEMIC FANTASY FOR MINORITY;

# 文字类排版: OpenType 特性

- 连字: ff → ff, fi → fi, fl → fl;
- 老式数字 (onum): 0123456789 → o123456789;
- 小写大写字母 (smcp): An Academic Fantasy For Minority → AN ACADEMIC FANTASY FOR MINORITY;
- case 特性: (西文括号配中文) → (西文括号配中文);

# 文字类排版: OpenType 特性

- 连字: ff → ff, fi → fi, fl → fl;
- 老式数字 (onum): 0123456789 → o123456789;
- 小写大写字母 (smcp): An Academic Fantasy For Minority → AN ACADEMIC FANTASY FOR MINORITY;
- case 特性: (西文括号配中文) → (西文括号配中文);
- 想要知道自己手上字体的 OpenType 特性? ⚡

## 文字类排版: 一些碎碎念

- 伪粗体和伪斜体. 正文中各种意义上都不推荐. 但需说明的是, 伪斜体可以用在排版代码中的注释.

## 文字类排版: 一些碎碎念

- 伪粗体和伪斜体. 正文中各种意义上都不推荐. 但需说明的是, 伪斜体可以用在排版代码中的注释.
- 罗马数字. I - x , U+2160-217F. Unicode 标准声称收容罗马数字乃是为了兼容性之举, 对大多数目的而言应用恰当的拉丁字母列替代之.

## 文字类排版：一些碎碎念

- 伪粗体和伪斜体. 正文中各种意义上都不推荐. 但需说明的是, 伪斜体可以用在排版代码中的注释.
- 罗马数字. I - x , U+2160–217F. Unicode 标准声称收容罗马数字乃是为了兼容性之举, 对大多数目的而言应用恰当的拉丁字母列替代之.
- 引号. 目前常用的蝌蚪形引号有四种 (不分中英):

U+201C    U+201D    U+2018    U+2019

“

”

‘

’

对 LATEX 和 CTEX 来说, 输入以上四种会调用中文字体的全宽引号. 欲得到调用西文字体的引号请输入 ` 和 '.

## 文字类排版：一些碎碎念

- 伪粗体和伪斜体. 正文中各种意义上都不推荐. 但需说明的是, 伪斜体可以用在排版代码中的注释.
- 罗马数字. I - x , U+2160-217F. Unicode 标准声称收容罗马数字乃是为了兼容性之举, 对大多数目的而言应用恰当的拉丁字母列替代之.
- 引号. 目前常用的蝌蚪形引号有四种 (不分中英):

U+201C    U+201D    U+2018    U+2019



对 LATEX 和 CTEX 来说, 输入以上四种会调用中文字体的全宽引号. 欲得到调用西文字体的引号请输入 ` 和 '.

- 破折号. 破折号可以说是最难以处理的标点, 目前没有完美的解决方案. ☺

# 喜欢的宏包

- 必备
  - amsmath
  - graphicx
  - hyperref
- 样式
  - caption
  - enumitem
  - fancyhdr
  - geometry
- 数学
  - bm
  - mathtools
- 表格
  - array
  - booktabs
  - longtable
  - tabularx
  - tabulararray
- 插图、绘图
  - subfig
  - tikz
  - pgfplots
- 字体
  - newtx
  - newpx
  - pifont
  - fontspec
- 杂项功能
  - beamer
  - biblatex
  - fancyhdr
  - listings
  - mhchem
  - siunitx
  - xcolor

\end{document}