

高速 \LaTeX 入门课程

曾梦辰

北京师范大学 OM 学社

最后编译: 2024 年 9 月 2 日

目录

- 1 前言
- 2 环境配置
- 3 编写文档
 - 我的 Hello, world!
 - 输入数学公式
- 4 一个完整的例子
- 5 结束语

1 前言

2 环境配置

3 编写文档

4 一个完整的例子

5 结束语

关于 \LaTeX

\LaTeX (读作 Lah-tek 或 Lay-tek, 不要读成 Lay-teks), 是一种基于 \TeX 的排版系统, 由美国计算机科学家 Leslie Lamport 在 20 世纪 80 年代初期开发. 利用这种格式系统的处理, 即使用户没有排版和程序设计的知识也可以充分发挥由 \TeX 所提供的强大功能, 不必一一亲自去设计或校对, 能在几天, 甚至几小时内生成很多具有书籍质量的印刷品生成复杂表格和数学公式, 这一点表现得尤为突出. 因此它非常适用于生成高印刷质量的科技和数学, 物理文档. 这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档.

\LaTeX 使用 \TeX 作为它的格式化引擎, 当前的版本是 $\text{\LaTeX} 2_{\epsilon}$.

关于 \LaTeX

\LaTeX (读作 Lah-tek 或 Lay-tek, 不要读成 Lay-teks), 是一种基于 \TeX 的排版系统, 由美国计算机科学家 Leslie Lamport 在 20 世纪 80 年代初期开发. 利用这种格式系统的处理, 即使用户没有排版和程序设计的知识也可以充分发挥由 \TeX 所提供的强大功能, 不必一一亲自去设计或校对, 能在几天, 甚至几小时内生成很多具有书籍质量的印刷品生成复杂表格和数学公式, 这一点表现得尤为突出. 因此它非常适用于生成高印刷质量的科技和数学, 物理文档. 这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档.

\LaTeX 使用 \TeX 作为它的格式化引擎, 当前的版本是 $\text{\LaTeX} 2_{\epsilon}$.

以上是抄的维基百科.

参考资料

我们推荐一些可以阅读的参考资料.

参考资料

我们推荐一些可以阅读的参考资料.

Overleaf. *Learn LaTeX in 30 minutes*. (本幻灯片的主要参考资料, 不过没有汉语翻译.)

参考资料

我们推荐一些可以阅读的参考资料.

Overleaf. *Learn LaTeX in 30 minutes*. (本幻灯片的主要参考资料, 不过没有汉语翻译.)

Tobias Oetiker, et. al. 一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍. (可能是最好的入门教材.)

参考资料

我们推荐一些可以阅读的参考资料.

Overleaf. *Learn LaTeX in 30 minutes*. (本幻灯片的主要参考资料, 不过没有汉语翻译.)

Tobias Oetiker, et. al. 一份 (不太) 简短的 $\text{\LaTeX}2_{\epsilon}$ 介绍. (可能是最好的入门教材.)

Frank Mittelbach, Ulrike Fischer. *The $\text{\LaTeX} Companion$* , 3rd Edition. (将近两千页的大肥书, 可以挑着看, 不要试图看完.)

参考资料

我们推荐一些可以阅读的参考资料.

Overleaf. *Learn LaTeX in 30 minutes*. (本幻灯片的主要参考资料, 不过没有汉语翻译.)

Tobias Oetiker, et. al. 一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍. (可能是最好的入门教材.)

Frank Mittelbach, Ulrike Fischer. *The \LaTeX Companion*, 3rd Edition. (将近两千页的大肥书, 可以挑着看, 不要试图看完.)

本幻灯片. 用法: 使用 Acrobat 或者你喜欢的 PDF 阅读器, 然后利用鼠标左键/鼠标滚轮/幻灯片右下角的按钮进行翻页.

听完本次课程后, 你将会...

- (0) 能够科学使用搜索引擎.
- (1) 学会如何使用`texdoc`命令或者在CTAN上查找宏包文档.
- (2) 学会如何使用 Overleaf 或者 T_EXPage 进行 L^AT_EX 写作.
- (3) 学会编写属于自己的 *Hello, World!* 文档.
- (4) 知道怎么输入数学公式.
- (5) 从我的群里薅走一些实用书籍.

听完本次课程后, 你将会...

- (0) 能够科学使用搜索引擎.
- (1) 学会如何使用`texdoc`命令或者在CTAN上查找宏包文档.
- (2) 学会如何使用 Overleaf 或者 T_EXPage 进行 L^AT_EX 写作.
- (3) 学会编写属于自己的 *Hello, World!* 文档.
- (4) 知道怎么输入数学公式.
- (5) 从我的群里薅走一些实用书籍.

我觉得这些就够了. 学习 L^AT_EX 不是听一次课就能学会的, 需要自己下来勤加练习. 有了上面这几点, 我认为已经有了一个良好的开端.

搜索引擎和texdoc

不要用百度 $\times \infty$. 在国内目前对科研, 学习目的最友好的搜索引擎是必应.¹ 在搜索时, 善用高级搜索选项:

高级搜索选项

在更少的时间找到要查找的内容。使用以下符号快速修改搜索词或搜索函数：

符号	函数
+	查找包含前面带有 + 符号的所有术语的网页。还允许包括通常被忽略的术语。
" "	查找短语中的确切字词。
()	查找或排除包含一组单词的网页。
AND 或 &	查找包含所有术语或短语的网页。
NOT 或 -	排除包含术语或短语的网页。
OR 或	查找包含任一术语或短语的网页。

¹微软快给我打钱.

搜索引擎和texdoc

如果你的电脑上安装了 $\text{T}_{\text{E}}\text{XLive}$, 那么你可以在命令行中输入 `texdoc Packagename` 来查看名为 `Packagename` 的宏包的文档. 如果没有安装的话, 可以在搜索引擎中搜索 `CTAN Packagename` 搜索宏包文档.

搜索引擎和texdoc

如果你的电脑上安装了 $\text{T}_{\text{E}}\text{XLive}$, 那么你可以在命令行中输入 `texdoc Packagename` 来查看名为 `Packagename` 的宏包的文档. 如果没有安装的话, 可以在搜索引擎中搜索 `CTAN Packagename` 搜索宏包文档.

练习 1: 使用 `texdoc` 或者在 CTAN 上找到文档 *lshort-chinese*.

练习 2: 搜索在 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 中使用 `bibtex` 插入参考文献的办法 (你不必现在学会), **排除** 所有来自 CSDN 的内容.

- 1 前言
- 2 环境配置**
- 3 编写文档
- 4 一个完整的例子
- 5 结束语

T_EXLive 的安装

这道题期末不考.

T_EXLive 的安装

这道题期末不考.

感兴趣的同学可以查看文档一份简短的关于 \LaTeX 安装的介绍
(`texdoc install-latex-guide-zh-cn`).

开始使用 Overleaf

首先打开网址<https://cn.overleaf.com>, 看到如下页面:



开始使用 Overleaf

首先打开网址<https://cn.overleaf.com>, 看到如下页面:



然后在这两个方框里填写你的邮箱和你想要的密码. Then enjoy \LaTeX ing!

- 1 前言
- 2 环境配置
- 3 编写文档
- 4 一个完整的例子
- 5 结束语

创建第一个项目

现在你创建好了账号，你应该会看到这样一个页面：



创建第一个项目

现在你创建好了账号，你应该会看到这样一个页面：



然后依次“创建一个新项目”，“空白项目”，并输入项目名称Hello, world!或者你喜欢的名字.

创建第一个项目

现在你创建好了账号，你应该会看到这样一个页面：



然后依次“创建一个新项目”，“空白项目”，并输入项目名称Hello, world!或者你喜欢的名字。

之后再创建项目的页面不长这样，但是也很容易操作。

编写 Hello, world! 的代码

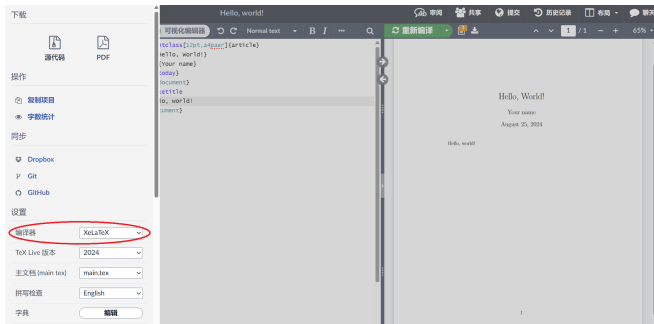
把 Overleaf 自带的代码全部删掉, 然后输入如下的代码: (请注意所有标点都是英文标点)

```
1 \documentclass[12pt]{article}
2 \usepackage[paper=a4paper]{geometry}
3 \title{Hello, World!}
4 \author{Your name\thanks{ID: Your ID}}
5 \date{\today}
6 \begin{document}
7     \maketitle
8     Hello, world! Hello, \LaTeX!
9 \end{document}
```

或者直接复制粘贴我群里发的代码.

一个好习惯

点击左上角“菜单”，然后如图将编译器调整为 XeLaTeX.



然后可以点“重新编译”查看效果.

使用 Overleaf 的好处是你不用操心编译，直接点“重新编译”就可以了.

控制序列

我们开始讨论 \LaTeX 的语法. 一个典型的**控制序列** (control sequence) 形如

1

```
\csname[option]{#1}
```

其中`\csname`是控制序列的名称, 花括号内的是控制序列作用的对象, 方括号内是参数. 有时候后两者是不必要的. 我们现在见到了控制序列`\documentclass[12pt]{article}`等, 接下来我们会仔细讨论它们的含义.

导言区

文档Hello, World!的导言区为

1
2

```
\documentclass[12pt]{article}  
\usepackage[paper=a4paper]{geometry}
```

导言区

文档Hello, World!的导言区为

```
1 \documentclass[12pt]{article}
2 \usepackage[paper=a4paper]{geometry}
```

在导言区中, 我们声明了**文档类**为article, 并引用了geometry包. 文档类article指的是文档的类型为“文章”, 文档类的参数12pt是全局字体大小, 默认为 10pt, 这里调成 12pt 是为了照顾大家的英语课. geometry包的参数paper指定了纸张大小为 A4.

导言区

文档Hello, World!的导言区为

```
1 \documentclass[12pt]{article}
2 \usepackage[paper=a4paper]{geometry}
```

在导言区中, 我们声明了**文档类**为article, 并引用了geometry包. 文档类article指的是文档的类型为“文章”, 文档类的参数12pt是全局字体大小, 默认为 10pt, 这里调成 12pt 是为了照顾大家的英语课. geometry包的参数paper指定了纸张大小为 A4.

练习: 将字体大小调节为 10pt 与 11pt, 并调节纸张大小为b5paper, 观察效果.

正文

正文由`\begin{document}`与`\end{document}`包裹. 在这之后的任何内容都会被编译器忽略.

正文

正文由`\begin{document}`与`\end{document}`包裹. 在这之后的任何内容都会被编译器忽略.

正文中的`\maketitle`将产生标题 (页). 在 \LaTeX 中, 在源代码里断行将会导致文章中断句, 而分段则通过在源代码中给出一个空行实现. 建议无论使用什么语言写作, 都使用英文标点符号.

练习: 在正文中输入以下代码, 并观察效果.

```
1      \maketitle
2      Hello, world!
3      Hello, \LaTeX!
4
5      This is my document.
```


标题, 作者与日期

观察以下五条控制序列:

```
1 \title{Hello, World!}  
2 \author{Your name\thanks{ID: Your ID}}  
3 \date{\today}
```

其中`\title{}`指定文章的标题, `\author`指定作者, `\thanks{}`提供作者的信息, `\date{}`指定日期, 而`\today`则输出当前的日期.

输入中文

直接在 \LaTeX 里输入中文是做不到的, 我们需要使用一些宏包或者文档类. 我们可以在导言区引用 `ctex` 包

```
1 \usepackage{ctex}
```

或者直接使用 `ctexart` 文档类 (中文文章文档类.)

```
1 \documentclass{ctexart}
```

输入中文

直接在 \LaTeX 里输入中文是做不到的, 我们需要使用一些宏包或者文档类. 我们可以在导言区引用 `ctex` 包

```
1 \usepackage{ctex}
```

或者直接使用 `ctexart` 文档类 (中文文章文档类.)

```
1 \documentclass{ctexart}
```

提示 1: 一定要使用 XeLaTeX 编译.

输入中文

直接在 \LaTeX 里输入中文是做不到的, 我们需要使用一些宏包或者文档类. 我们可以在导言区引用 `ctex` 包

```
1 \usepackage{ctex}
```

或者直接使用 `ctexart` 文档类 (中文文章文档类.)

```
1 \documentclass{ctexart}
```

提示 1: 一定要使用 XeLaTeX 编译.

提示 2: 课下你可能会找到某些告诉你使用 CJK 包或者 `cctart` 文档类输入中文的资料, 这是**过时的**, 一定不要学.

输入数学公式

现在流行的数学公式输入宏包是来自美国数学会的`amsmath`宏包. 新建一个项目 (这个应该都会了吧), 然后在导言区输入

```
1 \usepackage{amsmath}
2 \usepackage{amssymb}
3 \usepackage{amsthm}
```

(请记得补充完整文档类, 标题和`\maketitle`) 这三个宏包依次提供 AMS 数学环境, 更多的符号和定理环境. 然后在正文输入

```
1 We have Pythagorean theorem  $a^2+b^2=c^2$  and
   fundamental theorem of calculus
2  $\int_a^b f'(x)dx=f(b)-f(a)$ 
3 provided  $f'(x)$  is Riemann integrable.
```

数学公式: 语法

数学公式分为两种, 一种是行内公式, 另一种是行间公式, 分别使用 \dots 和 $\left[\dots\right]$ 包裹. 也有使用 $\left(\dots\right)$ 包裹行内公式和使用 $\$ \dots \$$ 包裹行间公式的, 但这两种写法我个人不推荐. 在上一页的公式中, 我们见到了 a^b 表示上标. 而下标则通过 a_b 表示. \int 则是算符, 它会作用在一些公式上. `amsmath`宏包提供了很多常见函数, 例如下面的例子:

1

傅里叶分析中的狄利克雷核为

2

$$\left[\sum_{n=-N}^N e^{inx} = \frac{\sin(N + \frac{1}{2})x}{\sin(x/2)}\right]$$

(记得引`ctex`包哦 ~)

数学公式：语法

在上一页的公式中，要注意花括号和算符的作用。请尝试编译以下代码并观察效果。

```
1 \[e^{ix}=\cos(x)+i\sin(x)\]  
2 \[e^{\mathrm{i}x}=\cos(x)+i\sin(x)\]
```

事实上， $\text{T}_{\text{E}}\text{X}$ 引擎编译时的实际过程是逐字符展开，一个控制序列实际上是一系列字符的组合，而花括号形成一个“组”，并被视为一个字符。

数学公式：语法

在上一页的公式中，要注意花括号和算符的作用。请尝试编译以下代码并观察效果。

```
1 \[e^{ix}=\cos(x)+isin(x)\]
2 \[e^{\{ix\}}=\cos(x)+i\sin(x)\]
```

事实上， $\text{T}_\text{E}\text{X}$ 引擎编译时的实际过程是逐字符展开，一个控制序列实际上是一系列字符的组合，而花括号形成一个“组”，并被视为一个字符。

回到上一页的公式，我们见到了表示分式的 $\backslash\text{frac}\{\}\{\}$ 。与分式类似的有根式 $\backslash\text{sqrt}[\]\{\}$ 。请编译以下代码并观察效果。

```
1 方程 $x^n=a$  ( $a>0$ ) 的 $n$ 个复根为
2 \[x_k=\sqrt[n]{a}(\cos\frac{2k\pi}{n}+i\sin\frac{2k\pi}{n})\]
```


学习数学公式输入

受到篇幅和课时限制, 本幻灯片不可能面面俱到地讲授 \LaTeX 的公式输入. 所以我们应当阅读一些书籍, 例如我们提到的一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍. 接下来我们通过阅读学习一些 \LaTeX 公式输入技巧.

学习数学公式输入

受到篇幅和课时限制, 本幻灯片不可能面面俱到地讲授 \LaTeX 的公式输入. 所以我们应当阅读一些书籍, 例如我们提到的一份 (不太) 简短的 \LaTeX 2 ϵ 介绍. 接下来我们通过阅读学习一些 \LaTeX 公式输入技巧.

现在请阅读 4.3.9 小节, 并重新输入上一页的公式:

$$x_k = \sqrt[n]{a} \left(\cos \frac{2k\pi}{n} + i \sin \frac{2k\pi}{n} \right)$$

提示: 我们这里使用了定界符来控制括号的大小.

学习数学公式输入

现在请阅读 4.5 节, 并输入下面的矩阵公式:

$$\frac{d}{dx} \left(\det \begin{bmatrix} f_{11}(x) & \cdots & f_{1n}(x) \\ \vdots & \ddots & \vdots \\ f_{n1}(x) & \cdots & f_{nn}(x) \end{bmatrix} \right) \\ = \sum_{i=1}^n \det \begin{bmatrix} f_{11}(x) & \cdots & f'_{1i}(x) & \cdots & f_{1n}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{n1}(x) & \cdots & f'_{ni}(x) & \cdots & f_{nn}(x) \end{bmatrix}$$

输入的时候不需要考虑换行, 直接输入即可.

可能会用到的符号: 行列式`\det`, 横省略号`\cdots`, 竖省略号`\vdots`, 斜省略号`\ddots`.

数学符号表

在lshort-chinese中, 第 4.9 节为数学符号表. 请对照数学符号表输入以下公式:

$$\Gamma_{\beta\gamma}^{\alpha} = \frac{1}{2}g^{\alpha\nu}(\partial_{\beta}g_{\nu\gamma} + \partial_{\gamma}g_{\beta\nu} - \partial_{\nu}g_{\beta\gamma}).$$

以及

$$\text{Vol}(B_r(p)) = \omega_n r^n \left(1 - \frac{\text{Scal}(p)}{6(n+2)} r^2 + O(r^3) \right) \quad (r \rightarrow 0).$$

提示: Vol 与 Scal 是算符, 请用`\operatorname{}`包裹它们. 可以用`\quad`制造一个较长的空格.

数学符号表

在lshort-chinese中, 第 4.9 节为数学符号表. 请对照数学符号表输入以下公式:

$$\Gamma_{\beta\gamma}^{\alpha} = \frac{1}{2} g^{\alpha\nu} (\partial_{\beta} g_{\nu\gamma} + \partial_{\gamma} g_{\beta\nu} - \partial_{\nu} g_{\beta\gamma}).$$

以及

$$\text{Vol}(B_r(p)) = \omega_n r^n \left(1 - \frac{\text{Scal}(p)}{6(n+2)} r^2 + O(r^3) \right) \quad (r \rightarrow 0).$$

提示: Vol 与 Scal 是算符, 请用`\operatorname{}`包裹它们. 可以用`\quad`制造一个较长的空格.

lshort-chinese中的符号表是一个比较小 (但是够用) 的符号表, 完整的符号表请使用texdoc或者 CTAN 查找文档symbols (完整名称为 *The Comprehensive L^AT_EX Symbol List*).

- 1 前言
- 2 环境配置
- 3 编写文档
- 4 一个完整的例子**
- 5 结束语

一个完整的例子

请再次新建一个项目 (这次你可以不删代码), 我们的目标是尽可能地输出如图的文档.

Scalar Curvature

Your name

September 2, 2024

1 Introduction

It is known that scalar curvature controls the volume of geodesic ball.

2 Formula

Theorem 1. *The asymptotic expression of volume of geodesic ball is:*

$$\text{Vol}(B_r(p)) = \omega_n r^n \left(1 - \frac{\text{Scal}(p)}{6(n+2)} r^2 + O(r^3) \right) \quad (r \rightarrow 0).$$

Proof. See Gallot, et. al., *Riemannian Geometry*.

□

文档结构

article文档类提供了从高到低`\section`, `\subsection`, `\subsubsection`三个层次的章节. 如果你之后可能会用到book文档类的话, 最高还有一个`\chapter`章节层级. 在你需要的位置可以放置`\tableofcontents`生成目录.
练习: 随便打开一个项目, 然后设置小节并生成目录.

字体控制

LaTeX 中提供了控制字体类型的命令. 具体使用如下表:

表 5.1: 字体命令

<code>\rmfamily</code>	<code>\textrm{...}</code>	roman	衬线字体 (罗马体)
<code>\sffamily</code>	<code>\textsf{...}</code>	sans serif	无衬线字体
<code>\ttfamily</code>	<code>\texttt{...}</code>	typewriter	等宽字体
<code>\mdseries</code>	<code>\textmd{...}</code>	medium	正常粗细 (中等)
<code>\bfseries</code>	<code>\textbf{...}</code>	bold face	粗体
<code>\upshape</code>	<code>\textup{...}</code>	upright	直立体
<code>\itshape</code>	<code>\textit{...}</code>	<i>italic</i>	意大利斜体
<code>\slshape</code>	<code>\textsl{...}</code>	<i>slanted</i>	倾斜体
<code>\scshape</code>	<code>\textsc{...}</code>	SMALL CAPS	小型大写字母
<code>\em</code>	<code>\emph{...}</code>	<i>emphasized</i>	强调, 默认斜体
<code>\normalfont</code>	<code>\textnormal{...}</code>	normal font	默认字体

字体控制

有些同学可能有使用 Times 类字体的需求, 这里提供两种解决方案:

(1) 使用fontspec宏包, 一个最小工作示例是

```
1 \usepackage{fontspec}
2 \setmainfont{Times New Roman}
```

注意使用fontspec包的话必须使用 XeLaTeX 编译.

(2) 使用newtxtext与newtxmath宏包. 这将同时改变你的正文字体与数学字体.

定理环境

amsthm包中提供了声明定理环境的方法. 一个例子如下:

```
1 \theoremstyle{plain}
2 \newtheorem{theorem}{Theorem}
```

其中`\theoremstyle{}`决定了接下来声明的定理环境的风格, 有 `plain`, `definition` 和 `remark` 三种选择. `\newtheorem{ }{ }`则声明了定理环境, 前一个参数表示控制序列的名称, 后一个参数则表示显示的名称. 例如, 在以上代码声明了`Theorem`环境之后, 可以通过代码

```
1 \begin{theorem}
2   Scalar curvature\dots
3 \end{theorem}
```

来使用`Theorem`环境.

证明环境

amsmath包中也提供了证明环境, 使用`\begin{proof}`与`\end{proof}`包裹即可. 例如

```
1 \begin{proof}
2   We have Ricci identity
3   
$$\nabla_{Y,X}T - \nabla_{X,Y}T = R(X,Y)T$$

4 \end{proof}
```

一个完整的例子

现在你应该可以完成这个练习了.

Scalar Curvature

Your name

September 2, 2024

1 Introduction

It is known that scalar curvature controls the volume of geodesic ball.

2 Formula

Theorem 1. *The asymptotic expression of volume of geodesic ball is:*

$$\text{Vol}(B_r(p)) = \omega_n r^n \left(1 - \frac{\text{Scal}(p)}{6(n+2)} r^2 + O(r^3) \right) \quad (r \rightarrow 0).$$

Proof. See Gallot, et. al., *Riemannian Geometry*. □

- 1 前言
- 2 环境配置
- 3 编写文档
- 4 一个完整的例子
- 5 结束语**

最后的话

最后我想说

最后的话

最后我想说

首先不要害怕使用 \LaTeX , 虽然 \LaTeX 上手确实不容易, 但是对着书籍和搜索引擎边用边改也是能够入门的.

最后的话

最后我想说

首先不要害怕使用 \LaTeX , 虽然 \LaTeX 上手确实不容易, 但是对着书籍和搜索引擎边用边改也是能够入门的.

请一定要去阅读一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍, 即使没有时间全部读完也要挑着重点读.

最后的话

最后我想说

首先不要害怕使用 \LaTeX , 虽然 \LaTeX 上手确实不容易, 但是对着书籍和搜索引擎边用边改也是能够入门的.

请一定要去阅读一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍, 即使没有时间全部读完也要挑着重点读.

善用搜索引擎, 可以试试 AI...?

最后的话

最后我想说

首先不要害怕使用 \LaTeX , 虽然 \LaTeX 上手确实不容易, 但是对着书籍和搜索引擎边用边改也是能够入门的.

请一定要去阅读一份 (不太) 简短的 $\text{\LaTeX} 2_{\epsilon}$ 介绍, 即使没有时间全部读完也要挑着重点读.

善用搜索引擎, 可以试试 AI...?

实在不行了就多问师兄师姐. 直接找我装 \TeX Live 都是可以的.

感谢聆听!