



boook 的简单使用

LATEX 的简单入门教程 | 凉凉

BOOK

Preface

抹眼泪, L^AT_EX¹ 好麻烦. 虽然好用是真的好用, 但是麻烦是真的麻烦. 如果你愿意献祭自己的时间, 那么你一定能得到结果.

¹L^AT_EX 是一种用来排版的软件.

Contents

Part I	L^AT_EX for Different People	1
1	Only Math Needs: TeXmacs	3
1.1	About TeXmacs and Versions	3
1.2	So, How to Use?	4
1.2.1	Document Structure	4
1.2.2	Math Env	4
1.3	Further?	5
2	No, I need L^AT_EX	7
2.1	Installation	7
2.2	Basic Useage	7
2.2.1	Document Structure	9
2.2.2	Math and Symbols	10
2.2.3	Image and Tables	10
2.3	Speed Up L ^A T _E X Typing	11
2.4	Debugging LaTeX	12
A	测试	13
B	Boook 使用方法	15
B.1	使用 Boook	15
B.1.1	文档类的 options	15

B.1.2	一些填充信息	16
B.1.3	环境	16
B.2	开发 Booook	17

Part I

LAT_EX for Different People

Chapter 1

Only Math Needs: TeX-macs

如果你没有很多的排版上的需求, 没有很多自定义的脚本设置的要求, 那么我推荐使用一个叫做 TeXmacs ([link](#)) 的开源软件. 这个软件在公式输入的快捷键的设置上, 在 L^AT_EX 和 所见即所得之间做了很好的取舍.

1.1 About TeXmacs and Versions

我不想抄官网上的介绍, 如果你觉得英文看不惯的话, 可以参考王垠的一篇介绍性的博客 ([link](#)).

这里我就按自己的使用感受来介绍 TeXmacs:

- TeXmacs 名字中的 **macs** 不是针对 macOS, 而应该是对 Emacs¹ 的一个借鉴.
- TeXmacs 和 Emacs 类似, 建立在 Lisp 的一个方言上. 并且原则上, 你可以为其编写插件来无限地拓展其功能.

¹Emacs (官网, 中文社区) 是一个非常有趣的文本编辑器. 也是我目前使用的一个文本编辑器.

不过它已经自带了非常多的功能和插件, 所以我认为在很大程度上, 你可以做到开箱即用.

- 并且作为开源软件, 有很多人去维护它. 比如在 Windows 上, 原版的 TeXmacs 表现并不理想, 所以有人为其编写了更加流畅的版本 (mogan)²

1.2 So, How to Use?

因为我是双非数理专科学校的学生, 所以我只会着重介绍其公式输入的部分. 至于其他的一些东西, 我只会简单地跳过. 如果你需要更多的帮助, 豆瓣的中文社区 ([link](#)), 以及一个论坛 ([link](#))³ 都是一些可用的资源.

1.2.1 Document Structure

关于文档结构, 可以参考后面的 L^AT_EX 部分2.2.1的使用方法介绍.

像 L^AT_EX 一样, 可以通过 `\maketitle{}`, `\chapter{}`, `\section{}` 等命令展开环境.

不过更加方便纯新人使用的一个方法就是直接从菜单中选择插入 (Insert), 然后选择要添加的东西就好了.

1.2.2 Math Env

如果想要输入公式的话, 最快捷的方式就是按下 `\$`, 一般也就是按下 Shift-4: 进入 inline Math. 最方便的方法从菜单中选择插入 (Insert), 数学 (Mathematics), 然后选择是行内公式还是其他类型.⁴

²不过虽然这个版本在 Windows 下表现很好, 但是因为作者精力有限, 所以放弃了对 *unix 版本的维护, 所以如果是 macOS 或 Linux 的用户, 我的建议是使用原版的 TeXmacs.

³因为最近 (2022.11) 校园网不太行, 没法验证网站是否可用.

⁴虽然但是, 如果你想要学会不同的公式的快捷键的话, 在菜单的下拉栏里面就自带有快捷键提示了.

那么就是一些数学符号的输入了, 尽管可以像 L^AT_EX 一样, 通过命令来输入符号, 但是这样效率实在是有点低. 而在 TeXmacs 中, 通过输入一个字符 (或者字符组合), 然后按下 Tab 键或者 Shift-Tab 键, 就可以在预设的输入里面快速切换, 比如:

- j Tab → θ
- f Tab → φ
- a Tab → α
- b Tab → β
- g Tab → γ
- ...

并且还有很多的自动的替换, 比如输入 \rightarrow , 会被自动替换成 \rightarrow , 输入 \leq , 会变成 \leq 等等.

以及一些快捷键: 比如输入分数 C-f, 输入根号的 C-s.⁵

1.3 Further?

差点忘了这是介绍 L^AT_EX 的文档了, 所以就不多介绍了. 这里建议善用搜索引擎, 善用自带的 F1 文档提示.

如果想要更进一步成为 TeXmacs 的 contributor 的话, 我在此表示衷心的感谢.

⁵因为我经常改快捷键, 所以我也不清楚具体的快捷键映射是什么. 所以就不多介绍了. 直接上手会比较快.

Chapter 2

No, I need L^AT_EX

好吧, 看来你是一个 L^AT_EX 的严谨的狂热粉丝.

2.1 Installation

因为 L^AT_EX 有很多的发行版, 并且安装工作对我来说是很久之前的事了. 所以我可能并不能把这个部分写得很好, 这里建议是随便选择一个发行版, 然后把安装的事情都忘掉:

- TeXLive ([link](#))
- MikTeX ([link](#)), 这个发行版的好处就是可以不必一口气安装好几个 G 的软件包, 可以随用随下载.
- MacTeX ([link](#)), 在 macOS 上的, 有部分安装的方式.¹

2.2 Basic Useage

如果你英语足够好的话, 我的建议是去看 Overleaf 的一个教程: Learn L^AT_EX in 30 minutes. 或者这个文档 (一份不太简短的 L^AT_EX 2_c 介绍).

¹通过 brew 安装: `brew install basictex`.

一般来说, 如果你啥也不想干, 一般直接套模版就好了. 一些好用的模版比如 ElegentTeX ([link](#)) 的一系列的模版. 或者稍微难用一点的我的这个模版 boook ([github](#)).

```
\documentclass{boook}
\title{文档标题}
\author{凉凉}

\begin{document}
\maketitle
\tableofcontents

\chapter{章}
一些文字.
\section{节}
另外一些文字.
\subsection{小节}
以及, 其他的更多的文字.
\subsubsection{最小子节}
很好, 更多的其他的一些文字.
\end{document}
```

一个最简单的文档长这个样子, 你可以自行替换掉使用的 `\documentclass{}` 中使用的文档类. 对于 LATEX 自带的文档类, 主要有: beamer, article, book. 或者你可以在 CTAN ([link](#)) 上找到相应的列表.

观察其中的大部分的东西, 你会发现, 在形式上, LATEX 的文档有如下三种东西:

- 命令, 用斜杠 `\` 开始的一堆东西, 可能会有用花括号 `{}` 来传递参数.
- 环境, 用 `\begin{env}\end{env}` 包括一段文本, 里面就像是一个独立空间, 受到 `env` 的环境名称的修饰. 表现出和外部环境稍微不同的状态.
- 纯文本, 顾名思义, 大概就是各种各样的文字.

2.2.1 Document Structure

关于文档结构的命令和环境. 其实大部分就是在上面的那段代码里面提到了. 这里就只是介绍该如何使用这些命令来帮助文档写作. 或者说, 如何利用 LATEX 来结构化地写作.

一般来说, 如果你知道自己要写什么的话, 可以先写文档的结构, 然后再写文档的内容.

以物理基础实验报告为例, 一个简单的结构可以是:

```
\documentclass{report}
\title{基础物理实验报告}
\author{凉凉}
\begin{document}
\section{实验目的}
\section{实验仪器}
\section{实验原理}
\section{实验内容}
\subsection{实验一：测量}
\subsubsection{步骤}
\begin{enumerate}
\item 做了什么
\item 做了什么
\end{enumerate}
\end{document}
```

有了文章结构之后, 基本上要做的事情就是像填空一样填充结构了. 不过如果你不知道自己要写什么的话, 那么事情可能就会变得挺难办的了. 因为 LATEX 可能会让你非常痛苦. 因为这个时候, 你不仅要操心写什么, 甚至还要操心在 LATEX 里面怎么组织写什么.

2.2.2 Math and Symbols

当然, 作为双非数理专科学校, 数学公式的输入肯定是少不了的.²

一个小小的建议: 在自己的电脑上常备一份用来速查的手册,³ 方便自己忘记的时候去查阅.

在 LATEX 里面输入公式的话, 最简单的方式就是直接在代码中输入 `$ $` 也就是美元符号, 在其中插入公式代码, 就能够输入行内公式了. 比如: $F(X) = \int_0^1 f(x)dx$.

而通过 `\[]` 的方式可以插入行公式.⁴ 比如这样:

$$\oint_l f(z)dz = 2\pi i \sum_j \text{Res}f(z_j)$$

或者你也可以使用 `\begin{equation}\end{equation}` 的方式来输入行公式, 这样的形式的好处就是可以生成带编号的公式, 比如:

$$\text{Res}f(z) = a_{-1} = \lim_{z \rightarrow z_0} \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z - z_0)^{m-1} f(z)] \quad (2.1)$$

并且你还能够通过 `\label{}` 的方式来设置公式标签, 在文档中引用这个公式, 比如: 公式2.1.

2.2.3 Image and Tables

这里只是介绍一些一般性的操作, 如果使用的是 boook 文档类的话, 我已经将一些一般性的操作封装成了简单的命令了, 可以参考B.1.3.

² 虽然最近我发现, 只要能够折腾, 就能够让输入的速度变快. 前提是能够折腾, 具体可以参考2.3.

³ 比如这份 symbols.pdf ([link](#)), 或者在 CTAN 上的一份更加完整的 symbols-a4.pdf ([link](#)).

⁴ 虽然你也可以用 `$$ $$` 的方式来, 但是这样的方式在我这里会被语法检查 diss.

一些简单的操作:⁵

2.3 Speed Up LATEX Typing

如果你使用的是 TeXwork 来编辑你的 LATEX 文档的话, 确实, 很大程度上, 你会觉得 LATEX 真是一个复杂的东西. 你竟然要记住那么多的命令, 并且只要输错一个命令, 整个文档就会嘎嘎报错.⁶ 然而事实并不是这样的.

大部分时候, 至少我所见到的, 写 LATEX 的时候, 很少有完全靠人力来输入的, 自动化啊, 自动化! 比如在 VSCode 里面, 就会有自动化的公式, 命令补全插件. 并且也会有自动保存编译和链接编译文件的插件. 而一般来说:

首先, 不要重复造轮子: 对于公式的输入, 一个非常方便的方法可能是从 TeXmacs 或者 Mathematica 中已经有了的现成的公式里面复制出来, 直接粘贴到 LATEX 文档里面.⁷

然后, 对于重复的输入: 如果你的编辑器支持 Snippet 功能的话, 那么这就再好不过了. 你可以通过为编辑器添加 Snippet 的方法来保证自己可以快速输入各种模版.

以我的 Emacs 为例, 我就把我经常使用的一些符号按照 TeXmacs 里面的使用习惯添加到了我的 Yasnippet 里面了. 或者, 如果你是 Vim 用户, 可以参考 How I'm able to take notes in mathematics lectures using LaTeX and Vim.

并且如果你的 Snippet 系统支持代码运行的话, 那么事情可能就会变得更加有趣了. 甚至你可以写程序来生成 LATEX 的代码. 总之, 怎么简单, 怎么省事, 就怎么来.

比如这个是我写的一个用于生成签到表的 Rakefile 的一部分:

```
tables_raw = <<EOF
\\begin{tabularx}{\\textwidth}{|X|X|X|X|}
```

⁵ 虽然我觉得我肯定写得不如 Overleaf 上面的教程好. 所以放上教程链接: Inserting Images (图片插入), 以及 Tables(表格插入).

⁶ 关于如何处理报错的事情, 将会在之后的一节2.4 里面介绍.

⁷ 比如在 TeXmacs 里面, 编辑 (Edit), 复制为 LaTeX (Copy to LaTeX). 或者是在 Mathematica 里面, 使用复制为 LaTeX 的方法.

```
\hline
\multicolumn{4}{|>{\hspace=4\hspace}X|}{\bfseries
→ \ies{\%{tablename}}}\hline
\hline
\tablerows
\end{tabularx}
\vspace{5mm}
EOF
```

当然, 除了自动生成, 错误提示也挺重要的. 这里就不多讲了.

2.4 Debugging LaTeX

多亏了 boook 的编写, 我认为我现在对于 LATEX 的报错信息有了一些简单的理解.⁸

每当你被 xelatex 打断的时候, 这个时候, 命令行里面会显示打断你的命令所在的位置以及打断你的命令的提示和原因. (虽然有一个比较坑爹的地方就是, 常常提示你打断的原因的输入会在停止的位置的几行之前. 所以常常需要往上翻一下才能看到.)

对于一个普通的错误 (Error), 常常会见到这样的报错信息:

```
! Undefined control sequence.
1.319 \afakecommand
```

一般来说, 会用行号来告诉你问题出在那个地方. 这个时候一般就可以直接去那个位置去找了. 不过如果遇到一些稀奇古怪的问题, 这个时候, 除了怀疑自己是不是有问题, 可以去思考一下, 是不是自己用的包 (比如如果你用的是我的 boook 类的话, 那么强烈建议你去看看我的包里面有没写错了.) 有毛病.

⁸尽管这些都是一些非常简单, 甚至是非常肤浅的结论.

Appendix A

测试

凡人歌

你我皆凡人生在人世間
終日奔波苦一刻不得閒
既然不是仙難免有雜念
把道義就放兩旁把利字擺中間
多少男子漢一怒為紅顏
多少同林鳥已成了分飛燕
人生何其短何必苦苦戀
愛人不見了向誰去喊冤
問你何時曾看見這世界為了人們改變
有了夢寐以求的容顏是否就算是擁有春天
你我皆凡人生在人世間
終日奔波苦一刻不得閒
你既然不是仙難免有雜念
把道義就放兩旁把利字擺中間
多少男子漢一怒為紅顏
多少同林鳥已成了分飛燕
人生何其短何必苦苦戀
愛人不見了向誰去喊冤
問你何時曾看見這世界為了人們改變
有了夢寐以求的容顏是否就算是擁有春天

Appendix B

Boook 使用方法

Boook 的定位是一个用于学习如何编写 L^AT_EX 文档模版的一个小玩具. 所以其中并不会有很多高级的地方.



Figure B.1: Boook 标志

B.1 使用 Boook

B.1.1 文档类的 options

文档的属性通过如下方式传递:

```
\documentclass[lang=mix, colorsets=plain,  
→ geomode=default, code=minted]{boook}
```

目前 (2022,11,5) 有的属性有:

- `lang`, 有 `en`, `cn`, `mix` 三个选项.
默认是 `mix`, 即使用英文关键词, 支持中文. 另外两个
顾名思义.
- `colorsets`, 目前没有什么可以设置的, (目前只有默认的 `plain`) 建议有兴趣的同学帮我写.
- `geomode`, 目前有默认的 `default` 选项, 适用于内宽外
窄的印刷用. 可以选择 `digital`, 适用于居中的电子设
备阅读.
- `code`, 目前默认关闭 `none`, 可以使用 `minted` 打开.

B.1.2 一些填充信息

我提供了一些填充用的信息:

- `\title{}`
- `\subtitle{}`
- `\author{}`
- `\cover{}`, 建议使用 1476×1870 , 300 dpi 的图片.
- `\logo{}`, 建议不要改.

B.1.3 环境

图片插入

模仿 TeXmacs 里面的命令, 但是只留下了最简单的操作:

```
\image[带标题图片]{图片地址}
\image{不带标题图片地址}
```

代码

目前还没想好该怎么修改 `\mintinline`, 只有块代码的快速插入:

```
\begin{code}{ruby}
tree = Tree.new
moon = Moon.new
5.times do |i|
  i.stand_under(tree)
  .looking_at_the(moon)
  .feeling_sad_at("Why I have to coding so
    ↳ boring? ")
end
\end{code}
```

B.2 开发 Boook

请来到 boook 项目主页 ([boook github](#)), 提交你的 pull request 吧.

我是菜狗, 我需要你的帮助.