

# L<sup>A</sup>T<sub>E</sub>X 学习记录

潘重宇

兴业数金

2022/4/26

## 1 L<sup>A</sup>T<sub>E</sub>X 基础介绍

## 2 使用 L<sup>A</sup>T<sub>E</sub>X 进行编辑

# L<sup>A</sup>T<sub>E</sub>X 是什么

L<sup>A</sup>T<sub>E</sub>X (发音 LAY-tek 或 LAH-tek) 是一个编辑专业文档的工具。它基于 WYSIWYM (what you see is what you mean) 这个想法而出现, 这意味着你只要专注于文档内容, 编译器会完成排版、样式方面的工作。相比于 MS-word 这种需要自己在页面上进行繁琐排版的文档编辑工具, L<sup>A</sup>T<sub>E</sub>X 只需要使用者输入单纯的字符就能自己完成各种工作。

# 为什么学习 $\text{\LaTeX}$

$\text{\LaTeX}$  在全世界范围内被用于科学文章、书籍以及其它形式的出版物编辑。这不仅因为它可以用来优雅地排版，还允许使用者进行复杂的样式内容编辑，比如插入图片，创建表格，添加引用参考和配置统一的文本样式。得益于庞大的开源包资源， $\text{\LaTeX}$  有着无限的拓展可能。

使用  $\text{\LaTeX}$  的一个重要的理由是  $\text{\LaTeX}$  把文档的内容和样式分割开来。这意味着我们在完成了内容的编辑后可以非常轻易的改变文档的样式。这样就允许我们制作一个专门的样式模板，在日后的文档编辑工作中可以直接导入。

## 1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 基础介绍

## 2 使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 进行编辑

# 创建第一个 L<sup>A</sup>T<sub>E</sub>X 文档

让我们以一个简单的例子作为开始。输入文件是一个以.tex 为结尾的纯文本文件。它包含了计算机能够转译成 PDF 文档的代码。代码的第一行声明了文档类型，在这儿类型是 article。然后在 `\begin{document}` 与 `\end{document}` 之间我们写上了文档的内容。

```
1 \documentclass{article}
2
3 \begin{document}
4 First document. This is a simple example,
5 with no extra parameters or packages included.
6 \end{document}
```

# 创建第一个 L<sup>A</sup>T<sub>E</sub>X 文档

只有一段文字的文档显然是不够的。在这一步我们来加上作者、标题和日期并创建一个封面，然后对文档做些许排版。

```
1 \documentclass[12pt, letterpaper]{article}
2 \usepackage[utf8]{inputenc}
3
4 \title{First document}
5 \author{Pan \thanks{supported by Wang, Pang}}
6 \date{4/26/2022}
7 \begin{document}
8
9 \begin{titlepage}
10 \maketitle
11 \end{titlepage}
12
13 In this document some extra packages and parameters
14 were added. There is an encoding package
15 and pagesize and fontsize parameters.
16
17 \end{document}
```

# 创建第一个 L<sup>A</sup>T<sub>E</sub>X 文档

我们在 TeXstudio 里观察一下实现的效果（样例 2）



# 选择一个 L<sup>A</sup>T<sub>E</sub>X 编译器

我们把编译.tex 文件的程序称为 TeX 发行版，目前有以下一些保持更新的发行版可供选择。同时还有许多公司和社区提供在线的 TeX 编译工具，比如 Overleaf。

- MiKTeX for Windows

# 选择一个 L<sup>A</sup>T<sub>E</sub>X 编译器

我们把编译.tex 文件的程序称为 TeX 发行版，目前有以下一些保持更新的发行版可供选择。同时还有许多公司和社区提供在线的 TeX 编译工具，比如 Overleaf。

- MiKTeX for Windows
- TeX Live for Linux and other UNIX-like systems

# 选择一个 L<sup>A</sup>T<sub>E</sub>X 编译器

我们把编译.tex 文件的程序称为 TeX 发行版，目前有以下一些保持更新的发行版可供选择。同时还有许多公司和社区提供在线的 TeX 编译工具，比如 Overleaf。

- MiKTeX for Windows
- TeX Live for Linux and other UNIX-like systems
- MacTeX redistribution of TeX Live for macOS

# 选择一个 L<sup>A</sup>T<sub>E</sub>X 编辑工具

就像各种集成开发环境，L<sup>A</sup>T<sub>E</sub>X 也在各类操作系统上有着丰富的高级文本编辑工具，能提供拼写检查、代码补全以及直观的 PDF 生成预览等功能。同时 Emacs 和 vim 也有 L<sup>A</sup>T<sub>E</sub>X 支持插件。

- 开源编辑器：AUCTEX, GNU TeXmacs, Gummi, Kile, LaTeXila, MeWa, TeXShop, TeXnicCenter, Texmaker, TeXstudio, TeXworks

L<sup>A</sup>T<sub>E</sub>X 有以下几种编译方式

- latex 输出.DVI 文件

L<sup>A</sup>T<sub>E</sub>X 有以下几种编译方式

- latex 输出.DVI 文件
- pdflatex 输出.PDF 文件

L<sup>A</sup>T<sub>E</sub>X 有以下几种编译方式

- latex 输出.DVI 文件
- pdflatex 输出.PDF 文件
- XeLaTeX 支持 UTF-8 的编译方式

# L<sup>A</sup>T<sub>E</sub>X 的文档结构

L<sup>A</sup>T<sub>E</sub>X 里提供的文章结构由大至小为 Part, Chapter (仅在文档类型为 report 和 book 中存在), Section 和 Subsection。具体演示见样例 3。

除了默认的文档结构样式, 通过引入 titlesec 包 L<sup>A</sup>T<sub>E</sub>X 可以实现个性化的配置。如以下代码所示, 我们能修改 Part 和 Section 的表现

```
1 \titleformat
2 {\chapter} % command
3 [display] % shape
4 {\bfseries\Large\itshape} % format
5 {Story No. \ \thechapter} % label
6 {0.5ex} % sep
7 {
8 \rule{\textwidth}{1pt}
9 \vspace{1ex}
10 \centering
11 } % before-code
12 [
13 \vspace{-0.5ex}%
14 \rule{\textwidth}{0.3pt}
15 ] % after-code
```



# L<sup>A</sup>T<sub>E</sub>X 的文档结构

```
1 \titleformat
2 \titleformat{\section}[wrap]
3 {\normalfont\bfseries}
4 {\thesection.}{0.5em}{}
5
6 \titlespacing{\section}{12pc}{1.5ex plus .1ex minus .2ex}{1pc}
```

## 关于 L<sup>A</sup>T<sub>E</sub>X 的 class 与 package

在进行个性化样式的学习中我们不免会想到，如果在之后的文档编辑中需要再次使用这种样式，能否避免在新的文档代码里嵌入大块复杂的布局参数。答案是可以的。L<sup>A</sup>T<sub>E</sub>X 提供了名为 LaTeX2e 的模板工具让使用者能够编写自己的 class 与 package（这两者的区别在于 class 只能作用于特定的文档类型，而 package 能在任何地方引用），从而覆盖默认的样式参数或者实现自己的 begin-end 内容模块样式。

# 一些操纵演示

字体，表格，公式，定理，化学结构等。