



L^AT_EX 随笔

ElegantL^AT_EX 入门之作

作者：SummerSong

时间：2024 年 11 月 25 日



目录

版本更新历史	1
第一章 L^AT_EX 基础	3
1.1 L ^A T _E X 家族	3
1.2 L ^A T _E X 安装	4
1.2.1 用户名为中文安装 Texlive 失败	4
1.2.2 报错: ‘vars’ expected but ‘powershell’	4
1.3 L ^A T _E X 命令和代码结构	5
1.3.1 最短的 L ^A T _E X 代码	5
1.3.2 L ^A T _E X 命令和环境	5
1.3.3 L ^A T _E X 源代码结构	6
1.4 L ^A T _E X 文档类和宏包	6
1.4.1 L ^A T _E X 文档类	6
1.4.2 L ^A T _E X 宏包	6
1.5 L ^A T _E X 使用到的文件类型	7
1.6 L ^A T _E X 文件组织方式	7
第二章 版面和格式	9
2.1 字体设置	9
2.1.1 字体族	9
2.1.2 使用中文	9
2.1.3 全局字体设置	10
2.1.3.1 英文字体设置	10
2.1.3.2 中文字体设置	10
2.1.4 局部字体设置	10
2.1.4.1 英文字体设置	10
2.1.4.2 中文字体设置	10
2.2 分栏	10
2.3 断行、断页、断词	11
2.4 空格和分段	11
2.5 注释	11
2.6 特殊字符	12
2.7 问题	12
2.7.1 双面空白页	12
第三章 正文工具	13
3.1 章节	13
3.2 目录	13
3.3 页眉页脚	13
3.3.1 基本的页眉页脚样式	13
3.3.2 手动更改页眉页脚的内容	14
3.3.3 fancyhdr 宏包	14

3.3.4 pifont宏包触发页脚中文不显示	15
3.4 脚注	15
3.5 尾注	16
3.6 交叉引用	16
3.7 列表	16
3.7.1 无序列表	16
3.7.2 有序列表	16
3.7.3 paralist 宏包	16
3.8 代码展示	17
3.8.1 lstlisting 宏包	17
3.9 标点	18
3.9.1 引号	18
3.9.2 破折, 短横	18
3.9.3 下划线与删除线	18
第四章 图片	19
4.1 简述	19
4.2 浮动体	19
第五章 表格	20
5.1 基本表格	20
5.1.1 修改表格线	20
5.1.2 列格式	21
5.1.3 列宽	21
5.1.4 行距	22
5.1.5 表格标题	22
5.2 复杂表格	22
5.2.1 合并单元格	22
5.2.2 单元格旋转	23
5.3 问题	23
5.3.1 longtable 与 arydshln 宏包冲突	23
5.3.2 cline 命令 undefined control sequence 报错	23
5.3.3 multirow 宏包居中	24
第六章 数学公式	25
6.1 行内公式	25
6.2 行间公式	25
6.3 数学符号	26
6.3.1 指数、上下标和导数	26
6.3.2 分式和根式	26
6.4 符号表	26
6.4.1 L ^A T _E X 普通符号	27
6.4.1.1 文本/数学模式通用符号	27
6.4.1.2 希腊字母	27
6.4.1.3 二元关系符	28
6.4.1.4 二元运算符	28

6.4.1.5 巨算符	29
6.4.1.6 数学重音符号	29
6.4.1.7 箭头	29
6.4.1.8 作为重音的箭头符号	30
6.4.1.9 定界符	30
6.4.1.10 用于行间公式的大定界符	30
6.4.1.11 其他符号	30
6.4.2 \mathcal{AMS} 符号	30
6.4.2.1 希腊字母和希伯来字母	31
6.4.2.2 \mathcal{AMS} 二元关系符	31
6.4.2.3 \mathcal{AMS} 二元运算符	31
6.4.2.4 \mathcal{AMS} 箭头	32
6.4.2.5 \mathcal{AMS} 反义二元关系符和箭头	32
6.4.2.6 \mathcal{AMS} 定界符	33
6.4.2.7 \mathcal{AMS} 其它符号	33
6.5 多行公式	33
6.5.1 公用编号的多行公式	34
6.6	34
第七章 参考文献设置	35
7.1 问题	35
7.1.1 不按出现顺序引用	35
7.1.2 作者年份引用设置仅年份超链接	35
7.1.3 作者年份引用设置 et al 为斜体	35
7.1.4 beamer 参考文献断行	35
7.1.5 Template for the Computer Journal (COMJNL) 使用 bibtex 编译报错	36
第八章 类的编写	38
8.1 文档类和宏包的结构	38
8.1.1 标识	38
8.1.2 使用类和宏包	38
8.1.3 声明选项	38
8.1.3.1 简单文本方式	38
8.1.3.2 键-值系统	39
8.2 一个最小的文档类文件	39
8.2.1 示例：一个新闻简报类	40
8.3 用于类和宏包编写者的命令	41
8.3.1 识别	41
8.3.1.1 NeedsTeXFormat	41
8.3.1.2 ProvidesClass 和 ProvidesPackage	41
8.3.2 加载文件	41
8.3.2.1 LoadClass	41
8.3.2.2 RequirePackage	41
8.3.3 延迟代码	42
8.3.4 创建和使用键值选项	42

参考资料

43

版本更新历史

- 2024/04/10 更新：动笔
- ① 开始记录
 - ② 添加第 7.1.1 条 不按出现顺序引用
 - ③ 添加第 1 章 \LaTeX 基础 内容

- 2024/04/17 更新：完成第 2 章 版面和格式
- ① 添加第 2 章 版面和格式 内容

- 2024/04/18 更新：完成第 3 章 正文工具、第 4 章 图片
- ① 添加第 3 章 正文工具 内容
 - ② 添加第 4 章 图片 内容

- 2024/04/23 更新：完成第 5 章 表格、第 6 章 数学公式
- ① 添加第 5 章 表格 内容
 - ② 调整章节顺序
 - ③ 添加第 6 章 数学公式 内容

- 2024/05/08 更新：添加第 8.3.4 条 参考资料 部分
- ① 添加了三个参考资料

- 2024/05/09 更新：添加第 7 章 参考文献设置 部分问题
- ① 添加第 7.1.2 条 作者年份引用设置仅年份超链接
 - ② 添加第 7.1.3 条 作者年份引用设置 `et al` 为斜体

- 2024/05/15 更新：添加第 7.1.4 条 beamer 参考文献断行 部分问题
- ① 添加第 7.1.4 条 beamer 参考文献断行

- 2024/05/21 更新：添加第 5.3.1 条 longtable 与 `arydshln` 宏包冲突 部分问题
- ① 添加第 5.3.1 条 longtable 与 `arydshln` 宏包冲突

- 2024/06/02 更新：添加第 1.2.1 条 用户名为中文安装 Texlive 失败 部分问题
- ① 添加第 1.2.1 条 用户名为中文安装 Texlive 失败

- 2024/06/03 更新：添加第 3.3.4 条 `pifont`宏包触发页脚中文不显示 部分问题
- ① 添加第 3.3.4 条 `pifont`宏包触发页脚中文不显示

- 2024/06/08 更新：添加第 3.8.1 条 `lstlisting` 宏包 内容
- ① 添加第 3.8.1 条 `lstlisting` 宏包

- 2024/06/11 更新：添加第 5.3.2 条 `cline` 命令 `undefined control sequence` 报错 及第 5.3.3 条 `multirow` 宏包居中 内容
- ① 添加第 5.3.2 条 `cline` 命令 `undefined control sequence` 报错

-
- ② 添加第 5.3.3 条 `multirow` 宏包居中
-

2024/07/19 更新：添加问题及部分内容

- ① 添加第 1.2.2 条 报错：‘vars’ expected but ‘powershell’
 - ② 补充第 6.2 节 行间公式
 - ③ 添加第 6.3 节 数学符号
 - ④ 添加第 6.4 节 符号表
 - ⑤ 添加第 6.5 节 多行公式
-

2024/10/21 更新：添加第 8 章 类的编写

- ① 添加第 8 章 类的编写
 - ② 添加第 8.1 节 文档类和宏包的结构
 - ③ 添加第 8.2 节 一个最小的文档类文件
-

2024/10/23 更新：添加第 8.2.1 条 示例：一个新闻简报类 部分

- ① 添加第 8.2.1 条 示例：一个新闻简报类
 - ② 修改第 2.1 节 字体设置 内容
-

2024/11/04 更新：添加部分内容

- ① 添加第 5.2.2 条 单元格旋转
-

2024/11/05 更新：添加问题

- ① 添加第 7.1.5 条 Template for the Computer Journal (COMJNL) 使用 bibtex 编译报错
-

2024/11/06 更新：添加内容

- ① 添加第 8.3 节 用于类和宏包编写者的命令
-

2024/11/25 更新：添加内容

- ① 添加第 3.9 节 标点

第一章 L^AT_EX 基础

1.1 L^AT_EX 家族

本节介绍一下各种名词，这里主要引用一篇知乎文章：[TeX 家族（TeX, XeTeX, LuaTeX, XeLaTeX …看完这篇就懂了](#)），加之个人理解。

(引擎)

引擎是真正干活的程序。引擎的基本功能就是解释 TeX 语法，把字排成行，把行排成页，涉及到断字、断行、分页等算法。最原始的引擎是 TeX。

- TeX: 1978 年由 Donald Erwin Knuth（高德纳）开发。是后来大部分 TeX 相关的基础。其生成 dvi 文件，然后经由其他程序转换为 pdf 文件。
- pdfTeX: TeX 语言的又一个实现，将 TeX 代码直接编译成 PDF 文件。
- XeTeX: TeX 语言的新的实现，支持 Unicode 编码和直接访问操作系统字体。
- LuaTeX: TeX 语言的一个完整的有扩展的实现。LuaTeX 支持 Unicode、系统字体和内嵌语言扩展，能直接输出 PDF 格式文件，也可以仍然输出 DVI 格式。

我的理解：TeX 相当于汇编语言。

(格式)

TeX 语言本身只有 300 个命令，晦涩难懂，只适合非正常的人类。一个简单的符号可能就需要多个命令来实现，可以将这些最基本的命令封装起来做个简写（宏）以实现特殊的目的。一堆简写的合集就构成了格式。格式可以与不同的引擎相结合。

- Plain TeX: 由 Don Knuth 提供的最小的宏集合。
- LaTeX: 更易于使用的宏集，最常见的一种格式。
- ConTeXt: 另一种常见的格式。

我的理解：格式就是 C 语言等高级语言

(编译命令)

是实际调用的、结合了引擎和格式的命令。如 xelatex 命令是结合 XeTeX 引擎和 LaTeX 格式的一个编译命令。

(宏包)

一些辅助文件，在 LaTeX 中叫做 packages，在 ConTeXt 中叫做 modules。在 LaTeX 格式中，导言区的 usepackage 的作用就是引入各种宏包。宏包其实也是一堆基本的 TeX 命令的集合，只是其不够全，所以称之为宏包而不是格式。

我的理解：宏包类似 python 的库，里面有封装好的函数

(发行版)

一个完整的 TeX 需要最基本的 TeX 引擎、格式支持、各种辅助宏包、一些转换程序、GUI、编辑器、文档查看器等等。通过选择不同的组合就构成了不同的发行版。

- TeX Live: 支持 Linux, Windows, Mac OS
- MiKTeX: 只支持 Windows
- CTeX: CTeX 基于 MiKTeX，并加入了中文的支持，只支持 Windows。同时 CTEX 是一个网站，ctex

是可以很好支持中文的宏包。
我的理解：IDE?



1.2 L^AT_EX 安装

参考[TexLive+VScode+SumatraPDF 配置 LaTeX 编辑环境](#)

1.2.1 用户名为中文安装 Texlive 失败

问题：在使用 GUI 界面安装时, 出现的错误形如图1.1

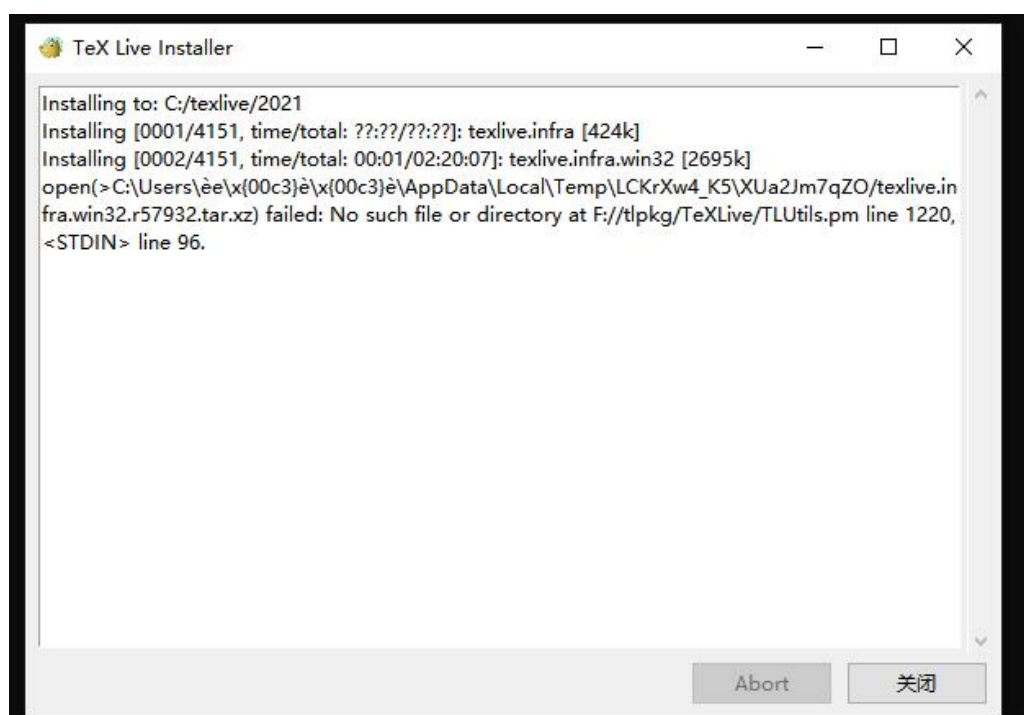


图 1.1: Texlive 安装错误

解决方法：参考[Windows 不合法的缓存路径导致 TeX Live 安装失败](#), 建议临时修改TEMP与TMP环境变量的值。在 cmd 中输入以下命令：

```
mkdir C:\temp
set TEMP=C:\temp
set TMP=C:\temp
```

然后运行安装脚本。

1.2.2 报错：‘vars’ expected but ‘powershell’

问题：安装 texlive 过程报错, 提示 'var' expected but 'powershell' 不是内部或外部的命令, 如图1.2所示。

解决方法：参考[texlive 安装过程中报错 vars expected but powershell](#), 添加如下内容到系统 PATH 路径中：

```
%SystemRoot%\System32
%SystemRoot%
```

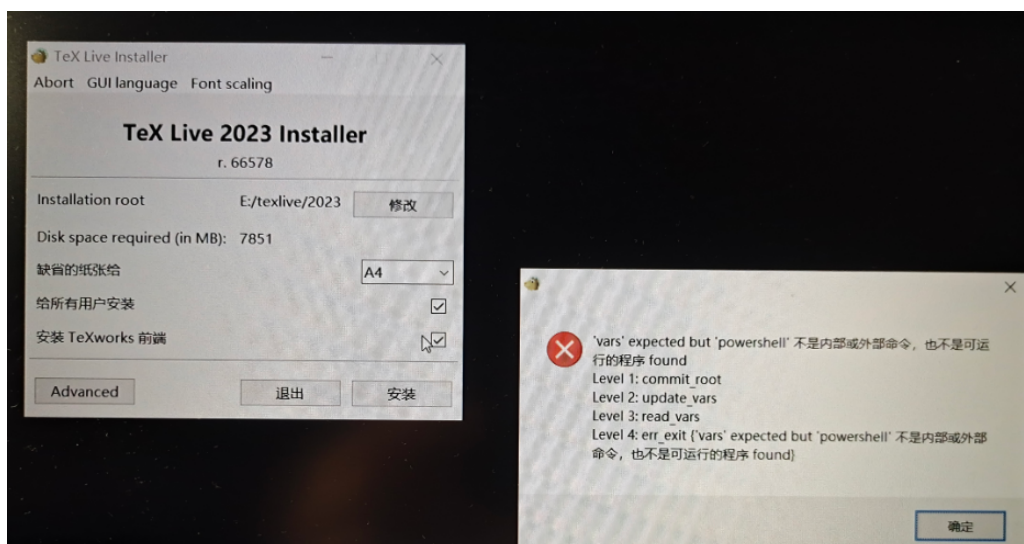


图 1.2: Texlive 安装错误

```
%SystemRoot%\System32\Wben
%SystemRoot%\System32\WindowsPowerShell\v1.0
```

1.3 L^AT_EX 命令和代码结构

1.3.1 最短的 L^AT_EX 代码

```
\documentclass{article}
\begin{document}
Hello, \LaTeX{}!
\end{document}
```

上述代码是 L^AT_EX 排版的最短代码。下面简单介绍 L^AT_EX 命令和代码的结构。

1.3.2 L^AT_EX 命令和环境

L^AT_EX 命令以反斜杠 \ 开头，后面跟一串字母，如 \LaTeX。它们以任意非字母符号为界限。

要注意 L^AT_EX 命令是对大小写敏感的，比如输入 \LaTeX 命令可以生成错落有致的 L^AT_EX 字母组合，但输入 \Latex 或者 \LaTex 什么都得不到，还会报错；它们与 \LaTeX 是不同的命令。

一些 L^AT_EX 命令可以接收一些参数，参数的内容会影响命令的效果。L^AT_EX 的参数分为可选参数和必选参数。可选参数以方括号 [] 包裹；必选参数一般以花括号 {} 包裹。还有些命令可以带一个星号 *，带星号和不带星号的命令效果有一定差异，一般是带不带编号，比如章节标题。

L^AT_EX 中还包括环境，用以令一些效果在局部生效，或是生成特殊的文档元素。L^AT_EX 环境的用法为一对命令 \begin 和 \end：

```
\begin{环境名称}[可选参数]{必选参数}
  内容
\end{环境名称}
```

有些命令（如 \bfseries）会对其后所有内容产生作用。若要限制其作用范围，则需要使用分组。L^AT_EX 使用一对花括号 {} 作为分组，在分组中使用的命令被限制在分组内，不会影响到分组外的内容。

1.3.3 L^AT_EX 源代码结构

L^AT_EX 源代码以一个 `\documentclass` 命令作为开头，它指定了文档使用的文档类。`document` 环境当中的内容是文档正文。

在 `\documentclass` 和 `\begin{document}` 之间的位置称为**导言区**。在导言区中常会使用 `\usepackage` 命令调用**宏包**，还会进行文档的全局设置。

```
\documentclass{...} % ...为某文档类，比如article、book、report等
% 导言区
\usepackage{ctex} %加载 ctex 宏包
\begin{document}
这里是正文。
\end{document}
%此后内容被忽略。
```

在正文区中，一般会包含一些文本、公式、图表、表格等内容。

1.4 L^AT_EX 文档类和宏包

1.4.1 L^AT_EX 文档类

文档类规定了 L^AT_EX 源代码所要生成的文档的性质——普通文章、书籍、演示文稿、个人简历等等。L^AT_EX 源代码的开头须用 `\documentclass` 指定文档类：

```
\documentclass[可选参数]{文档类名称}
```

文档类如 L^AT_EX 提供的 `article`、`report`、`book`，在其基础上派生的一些文档类，如支持中文排版的 `ctexart`、`ctexrep`、`ctexbook`，或者有其它功能的一些文档类，如 `moderncv`、`beamer` 等。L^AT_EX 提供的基础文档类见表 1.1，其中前三个习惯上称为“标准文档类”。

表 1.1: L^AT_EX 文档类

类别	说明
<code>article</code>	普通文章
<code>book</code>	书籍
<code>report</code>	报告
<code>letter</code>	信件
<code>slides</code>	幻灯片
<code>beamer</code>	幻灯片
<code>memoir</code>	书籍类
<code>acmart</code>	科技论文

可选参数为文档类指定选项，以全局地规定一些排版的参数，如字号、纸张大小、单双面等等。比如调用 `article` 文档类排版文章，指定纸张为 A4 大小，基本字号为 11pt，双面排版：

```
\documentclass[11pt,a4paper,twoside]{article}
```

1.4.2 L^AT_EX 宏包

在使用 L^AT_EX 时，时常需要依赖一些扩展来增强或补充 L^AT_EX 的功能，比如排版复杂的表格、插入图片、增加颜色甚至超链接等等。这些扩展称为**宏包**。调用宏包的方法非常类似调用文档类的方法：

```
\usepackage[可选参数]{宏包名称}
```

`\usepackage` 可以一次性调用多个宏包：

```
% 一次性调用三个排版表格常用的宏包
\usepackage{tabularx, makecell, multirow}
```

1.5 L^AT_EX 使用到的文件类型

除了源代码文件 `.tex` 以外，我们在使用 L^AT_EX 时还可能接触到各种格式的文件。本节简单介绍一下在使用 L^AT_EX 时能够经常见到的文件。

每个宏包和文档类都是带特定扩展名的文件，除此之外也有一些文件出现于 L^AT_EX 模板中：

- `.sty` 宏包文件。宏包的名称与文件名一致。
- `.cls` 文档类文件。文档类名称与文件名一致。
- `.bib` BibTeX 参考文献数据库文件。
- `.bst` BibTeX 用到的参考文献格式模板。

L^AT_EX 在编译过程中除了生成 `.dvi` 或 `.pdf` 格式的文档外，还可能会生成相当多的辅助文件和日志。一些功能如交叉引用、参考文献、目录、索引等，需要先通过编译生成辅助文件，然后再次编译时读入辅助文件得到正确的结果，所以复杂的 L^AT_EX 源代码可能要编译多次：

- `.log` 排版引擎生成的日志文件，供排查错误使用。
- `.aux` L^AT_EX 生成的主辅助文件，记录交叉引用、目录、参考文献的引用等。
- `.toc` L^AT_EX 生成的目录记录文件。
- `.lof` L^AT_EX 生成的图片目录记录文件。
- `.lot` L^AT_EX 生成的表格目录记录文件。
- `.bbl` BibTeX 生成的参考文献记录文件。
- `.blg` BibTeX 生成的日志文件。
- `.idx` L^AT_EX 生成的供 `makeindex` 处理的索引记录文件。
- `.ind` `makeindex` 处理 `.idx` 生成的用于排版的格式化索引文件。
- `.ilg` `makeindex` 生成的日志文件。
- `.out` `hyperref` 宏包生成的 PDF 书签记录文件。

1.6 L^AT_EX 文件组织方式

编写长篇文档时，例如当编写书籍、毕业论文时，单个源文件会使修改、校对变得十分困难。将源文件分割成若干个文件，例如将每章内容单独写在一个文件中，会大大简化修改和校对的工作。

L^AT_EX 提供了命令 `\include` 用来在源代码里插入文件：

```
\include{chapter1}
\include{chapter2}
\include{chapter3}
```

需要注意的是，`\include` 命令在读入新的文件之前会另起一页。有的时候我们不需要这样，则用 `\input` 命令，它仅仅只是将文件里面的内容插入到当前位置。

当导言区内容较多时，常常将其单独放置在一个 `.tex` 文件中，再用 `\input` 命令插入。复杂的图、表、代码等也会用类似的手段处理。

L^AT_EX 还提供了一个 `\includeonly` 命令来组织文件，用于导言区，指定只载入某些文件。导言区使用了 `\includeonly` 后，正文中不在其列表范围的 `\include` 命令不会起效：

```
\includeonly{chapter1,chapter3}
```

第二章 版面和格式

2.1 字体设置

2.1.1 字体族

LaTeX 默认预定义的三种字体族为：罗马字体族，无衬线字体族，打字机字体族。有两种命令可以局部使用这些字体族：

字体族	带参数命令	声明命令
罗马	<code>\textrm{ }</code>	<code>\rmfamily</code>
无衬线	<code>\textsf{ }</code>	<code>\sffamily</code>
打字机	<code>\texttt{ }</code>	<code>\ttfamily</code>

但是对于中文，没有太多的变体，因此，我们一般使用字体族来区分（宋体，隶书等）。

因此，中文字体的选择与西文字体是分离的，这个要注意。

`ctex` 宏包及其文档类（如 `ctexart`）另外新定义了一些组合字体，可以让中文拥有如同西文一样使用粗体（`\bfseries`）和意大利体（`\itshape`）的功能，并且重新定义了使他同时对中文起作用。这样就默认了中文的字体组为 `rm`，正常字体是宋体，粗体是黑体，意大利体是楷体，符合我们平时使用的习惯。

我是一个人(宋体)，

`\textbf{你呢（黑体）？}`

`\textit{不是吧，是汪（楷体）！}`

我是一个人(宋体)，

你呢（黑体）？

不是吧，是汪（楷体）！

中文的字体族，在 `ctex` 宏包及其文档类下进行了一部分预定义，在 `win` 下配置了四种字体族，并提供了如下的简化命令来进行使用

`{\songti 我是宋体}`，

`{\heiti 我是黑体}`，

`{\fangsong 我是仿宋}`，

`{\kaishu 我是楷体}`。

我是宋体，

我是黑体，

我是仿宋，

我是楷体。

2.1.2 使用中文

使用中文需要导入 `ctex` 宏包。

```
\usepackage{ctex}
```


2.1.3 全局字体设置

2.1.3.1 英文字体设置

设置正文罗马字体族，无衬线字体族和打字机字体族

```
\setmainfont[可选选项]{字体名}
```

```
\setsansfont[可选选项]{字体名}
```

```
\setmonofont[可选选项]{字体名}
```

设置好之后，`fontspec`会自动找到并匹配相应的粗体，斜体等，令我们使用`\bfseries`和`\itshape`也有效，若没有，那需要进行如下的设置。

```
\setmainfont[
  BoldFont      = texgyrepagella-bold.otf ,
  ItalicFont    = texgyrepagella-italic.otf ,
  BoldItalicFont = texgyrepagella-bolditalic.otf ]{texgyrepagella-regular.otf}
```

2.1.3.2 中文字体设置

```
\setCJKmainfont[可选选项]{字体名}
```

```
\setCJKsansfont[可选选项]{字体名}
```

```
\setCJKmonofont[可选选项]{字体名}
```

```
\setCJKfamilyfont{中文字体族}{字体名}
```

2.1.4 局部字体设置

2.1.4.1 英文字体设置

```
\usepackage{fontspec}
```

```
\newfontfamily\fugu{Luminari-Regular}
```

```
\newfontfamily\ptmr{PTMono-Regular}
```

2.1.4.2 中文字体设置

```
\newCJKfontfamily\qingsong{FZQKBYSJW--GB1-0}
```

或者

```
\setCJKfamilyfont{zh-song}{FZShuSong-Z01}
```

```
\newcommand*\songti{\CJKfamily{zh-song}}
```

可以使用`\qingsong`或者`\songti`命令来调用宋体字体。

2.2 分栏

\LaTeX 支持简单的单栏或双栏排版。标准文档类的全局选项 `onecolumn`、`twocolumn` 可控制全文分单栏或双栏排版。 \LaTeX 也提供了切换单/双栏排版的命令：

```
\onecolumn
```

```
\twocolumn
```

切换单/双栏排版时总是会另起一页(`\clearpage`)。在双栏模式下使用 `\newpage` 会换栏而不是换页;`\clearpage` 则能够换页。

使用 `multicol` 宏包可以实现多栏排版。并且切换不会另起一页。

```
\usepackage{multicol}
\begin{multicols}{2}
...
\end{multicols}
```

2.3 断行、断页、断词

`\\[length]` % 可以带可选参数 `length`, 用于在断行处添加垂直间距, 可以在表格公式等地方使用

`\newline` % 不带可选参数, 只能用于文本段落中

`\newpage` % 双栏模式下另起一栏, 单栏模式下另起一页

`\clearpage` % 单栏、双栏模式下都另起一页

如果 \LaTeX 遇到了很长的英文单词, 仅在单词之间的“空格”处断行无法生成疏密程度匀称的段落时, 就会考虑从单词中间断开。对于绝大多数单词, \LaTeX 能够找到合适的断词位置, 在断开的行尾加上连字符-。如果一些单词没能自动断词, 我们可以在单词内手动使用 `\-` 命令指定断词的位置:

```
I think this is: su\per\cal\-%
i\frag\i\lis\tic\ex\pi\-%
al\i\do\ciousu\per\cal\-%
i\frag\i\lis\tic\.
```

```
I think this is: supercalifragilisticexpialido-
ciousupercalifragilistic
```

2.4 空格和分段

\LaTeX 源代码中, 空格键和 `Tab` 键输入的空白字符视为“空格”。连续的若干个空白字符视为一个空格。一行开头的空格忽略不计。

行末的换行符视为一个空格; 但连续两个换行符, 也就是空行, 会将文字分段。多个空行被视为一个空行。也可以在行末使用 `\par` 命令分段。

```
This is a new paragraph. Text text text
text text text text text text.
```

```
This is a new paragraph.\par
This is a new paragraph.
```

```
This is a new paragraph. Text text text text
text text text text text.
```

```
This is a new paragraph.
This is a new paragraph.
```

2.5 注释

\LaTeX 用 `%` 字符作为注释。在这个字符之后直到行末, 所有的字符都被忽略, 行末的换行符也不引入空格。

```
This is an % short comment
% ---
% Long and organized
% comments
% ---
example: Comments do not bre%
ak a word.
```

This is an example: Comments do not break a word.

2.6 特殊字符

以下字符在 LATEX 里有特殊用途，如% 表示注释，\$、^、_ 等用于排版数学公式，& 用于排版表格，等等。直接输入这些字符得不到对应的符号，还往往会出错：

```
# $ % & { } _ ^ ~ \
```

如果想要输入以上符号，需要使用以下带反斜线的方式输入，类似编程语言里的“转义”符号：

```
\# \$ \% \& \{ \} \_ \^ \~ \textbackslash
```

2.7 问题

2.7.1 双面空白页

问题：采用twoside文档类，使用\cleardoublepage设置双面空白页，如何去除空白页？

解决方法：在所有\cleardoublepage之前添加\let\cleardoublepage\clearpage。

第三章 正文工具

3.1 章节

一篇结构化的、条理清晰文档一定是层次分明的，通过不同的命令分割为章、节、小节。三个标准文档类 `article`、`report` 和 `book`¹ 提供了划分章节的命令：

```
\chapter{章名}
\section{节名}
\subsection{小节名}
\subsubsection{}
\paragraph{}
\subparagraph{}
```

带星号版本不进行编号，也不生成目录项和页眉页脚。

3.2 目录

在合适部分添加：

```
\tableofcontents
```

上述命令会生成目录，标题默认是“Contents”，可通过下列命令更改：

```
\renewcommand{\contentsname}{目录}
```

要正确生成目录项，一般需要编译两次源代码。

如果要使用 `\chapter*` 命令的章节加入目录中，需要使用：

```
\addcontentsline{toc}{level}{title} %level为chapter, section等, title为章节名
```

3.3 页眉页脚

3.3.1 基本的页眉页脚样式

\LaTeX 中提供了命令 `pagestyle` 来修改页眉页脚的样式：

```
\pagestyle{page-style}
```

命令 `thispagestyle` 只影响当页的页眉页脚样式：

```
\thispagestyle{page-style}
```

`page-style` 参数为样式的名称，在 \LaTeX 里预定义了四类样式，见表 3.1。

其中 `headings` 的情况较为复杂：

- `article` 文档类，`twoside` 选项：偶数页为页码和节标题，奇数页为小节标题和页码；
- `article` 文档类，`oneside` 选项：页眉为节标题和页码；
- `report` 和 `book` 文档类，`twoside` 选项：偶数页为页码和章标题，奇数页为节标题和页码；

¹千万注意是标准文档类，其它文档类，如果不是从标准文档类衍生而来，很可能没有定义或只定义了一部分命令

表 3.1: L^AT_EX 预定义的页眉页脚样式

empty	页眉页脚为空
plain	页眉为空，页脚为页码。（article 和 report 文档类默认；book 文档类的每章第一页也为 plain 格式）
headings	页眉为章节标题和页码，页脚为空。（book 文档类默认）
myheadings	页眉为页码及 \markboth 和 \markright 命令手动指定的内容，页脚为空。

- report 和 book 文档类，oneside 选项：页眉为章标题和页码。

pagenumbering 命令令我们能够改变页眉页脚中的页码样式：

```
\pagenumbering{style}
```

style 为页码样式，默认为 arabic（阿拉伯数字），还可修改为 roman（小写罗马数字）、Roman（大写罗马数字）等。注意使用 pagenumbering 命令后会将页码重置为 1。book 文档类的 \frontmatter 和 \mainmatter 内部就使用了 \pagenumbering 命令切换页码样式。

3.3.2 手动更改页眉页脚的内容

对于 headings 或者 myheadings 样式，L^AT_EX 允许用户使用命令手动修改页眉上面的内容，特别是因为使用了 \chapter* 等命令而无法自动生成页眉页脚的情况：

```
\markright{right-mark}
\markboth{left-mark}{right-mark}
```

在双面排版、headings 或 myheadings 页眉页脚样式下，left-mark 和 right-mark 的内容分别预期出现在左页（偶数页）和右页（奇数页）。事实上 \chapter 和 \section 等章节命令内部也使用 \markboth 或者 \markright 生成页眉。

L^AT_EX 默认将页眉的内容都转为大写字母。如果保持字母的大小写，可以尝试以下代码²：

```
\renewcommand\chaptermark[1]{%
  \markboth{Chapter \thechapter\quad #1}{}}
\renewcommand\sectionmark[1]{%
  \markright{\thesection\quad #1}}
```

其中 \thechapter、\thesection 等命令为章节计数器的数值。注意以上代码适用于 report 和 book 文档类；对于 article 文档类，与两个页眉相关的命令分别为 \sectionmark 和 \subsectionmark。

3.3.3 fancyhdr 宏包

fancyhdr 宏包改善了页眉页脚样式的定义方式，允许我们将内容自由安置在页眉和页脚的左、中、右三个位置，还为页眉和页脚各加了一条横线。

fancyhdr 自定义了样式名称 fancy。使用 fancyhdr 宏包定义页眉页脚之前，通常先用 \pagestyle{fancy} 调用这个样式。在 fancyhdr 中定义页眉页脚的命令为：

```
\fancyhf[position]{...}
\fancyhead[position]{...}
\fancyfoot[position]{...}
```

²但是这不能改变页眉的斜体样式（slshape），斜体是定义在 headings 样式里的。如果不喜欢斜体，可在 markboth 等命令的参数里先使用 normalfont，再使用想要的字体样式命令，或直接尝试使用 fancyhdr 宏包。

其中 `position` 为 L（左）/C（中）/R（右）以及与 O（奇数页）/E（偶数页）字母的组合。`\fancyhf` 用于同时定义页眉和页脚，习惯上使用 `\fancyhf{}` 来清空页眉页脚的设置。

对于单页文档也可以使用 `\lhead`、`\lfoot` 等命令分别定义页眉和页脚。

```
\lhead{...}
\cfoot{...}
% l、c、r，左中右
```

下方代码给出了 `fancyhdr` 基础用法的一个示例，效果为将章节标题放在和 `headings` 一致的位置，但使用加粗格式；页码都放在页脚正中；修改横线宽度，“去掉”页脚的横线。

```
% 在导言区使用此代码
\usepackage{fancyhdr}
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}
\fancyfoot[C]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.4pt} % 注意不用 \setlength
\renewcommand{\footrulewidth}{0pt}
```

`fancyhdr` 还支持用 `\fancypagestyle` 为自定义的页眉页脚样式命名，或者重新定义已有的样式如 `plain` 等：

```
% 自定义 myfancy 样式
\fancypagestyle{myfancy}{%
  \fancyhf{}
  \fancyhead{...}
  \fancyfoot{...}
}
% 使用样式
\pagestyle{myfancy}
```

3.3.4 pifont宏包触发页脚中文不显示

问题：使用 `pifont` 宏包中的 `\ding` 命令，页脚中文不显示。

① $y = x^2$ ；② $S = ah$ (a 为常数)；③ $g = 200h + 98$ ；④ $y = 12t - 9$ 。
活动二：用函数表达式表示下列过程中两个变量间的关系，并指出一次函数、正

21 35

第 22 页 共 35 页

图 3.1: 页脚中文不显示

解决方法：参考 `pifont` 宏包可能会触发中文不显示，把 `\makeexeCJKactive` 加到页眉页脚的开头。

```
\fancyfoot[C]{ \songti 第\thepage 页 \hspace{0.5em} 共\pageref{LastPage}页}
```

3.4 脚注

脚注命令：


```
\footnote[手动指定序号, 可忽略]{脚注内容}
```

3.5 尾注

\LaTeX 目前没有提供直接插入尾注的命令, 但可以调用 `endnotes` 宏包实现。不常用, 暂时略过。

3.6 交叉引用

可以使用 `\ref` 命令和 `\label` 命令进行交叉引用。在正文中, 使用 `\ref{label}` 命令引用, 在相应位置使用 `\label{label}` 命令进行标记。

3.7 列表

3.7.1 无序列表

无序列表环境:

```
\begin{itemize}
  \item 第一项
  \item [-] 第二项
  \item [*] 第三项
\end{itemize}
```

- 第一项
- 第二项
- * 第三项

条目之间间距较大, 可以使用长度赋值命令将条目环境额外的垂直空白设置为 `0pt`, 达到与正文间距一致:

```
\itemsep=0pt
\parskip=0pt
```

3.7.2 有序列表

有序列表环境:

```
\begin{enumerate}
  \item 第一项
  \item 第二项
  \item 第三项
\end{enumerate}
```

1. 第一项
2. 第二项
3. 第三项

有序列表可以嵌套, 可对其序号、标号和前缀进行重定义, 但是比较麻烦, 可以使用 `paralist` 宏包。

3.7.3 `paralist` 宏包

暂未接触使用, 略过。

3.8 代码展示

因为 \LaTeX 排版会自动忽略空白字符等，如果需要按照原格式排版，可以使用 `verbatim` 环境。

不使用 `verbatim` 环境：

```
int main() {
    printf("Hello, world!");
    return 0;
}
```

使用 `verbatim` 环境：

```
\begin{verbatim}
int main() {
    printf("Hello, world!");
    return 0;
}
\end{verbatim}
```

不使用 `verbatim` 环境：int main() printf("Hello, world!"); return 0;

使用 `verbatim` 环境：

```
int main() {
    printf("Hello, world!");
    return 0;
}
```

行内可以使用 `\verb|内容|`。

3.8.1 `lstlisting` 宏包

也可以使用 `lstlisting` 宏包。可以实现复杂的高亮效果，但需要额外的配置。

添加标题：

```
\begin{lstlisting}[caption=代码标题]
```

代码内容

```
\end{lstlisting}
```

设置标题位置，在导言区配置

```
\usepackage{listings}
\lstset{captionpos=b} % 标题位置在下方,t为上方
```

可以设置标签：

```
\renewcommand\lstlistingname{\figurename}
```

设置代码环境计数器与图计数器保持一致：

```
\makeatletter
\AtBeginDocument{%
    \let\c@figure\c@lstlisting
    \let\thefigure\thelstlisting
    \let\ftype@lstlisting\ftype@figure % give the floats the same precedence
}
\makeatother
```

3.9 标点

3.9.1 引号

英文左单引号是```，右单引号是`'`，左双引号是````，右双引号是`''`。

英文下的引号嵌套需要借助`\thinspace`命令分隔，比如：

```
``\thinspace`Max' is here.``
```

```
```Max' is here.
```

“‘Max’ is here.”

“‘Max’ is here.

中文下的单引号和双引号可以用中文输入法直接输入。

### 3.9.2 破折，短横

英文的短横分为三种：

- 连字符：输入一个短横：`-`，效果如 daughter-in-law
- 数字起止符：输入两个短横：`--`，效果如：page 1–2
- 破折号：输入三个短横：`---`，效果如：Listen—I’ m serious.

中文的破折号可以直接使用日常的输入方式。

### 3.9.3 下划线与删除线

$\text{\LaTeX}$  提供`\underline`命令设置下划线。

```
\underline{underlined text}
```

underlined text

`\underline`命令提供的下划线样式不够灵活，`ulem` 或 `CJKfntef` 宏包提供了更灵活的下划线命令，`CJKfntef` 宏包对中文支持更好。

```
\CJKunderdot{important 非常重要}\\
\CJKunderline{notice 注意}\\
\CJKunderdblline{urgent 必须}\\
\CJKunderwave{prompt 提示}\\
\CJKsout{wrong 错误}\\
\CJKxout{removed 删除}
```

important 非常重要  
notice 注意  
urgent 必须  
prompt 提示  
~~wrong 错误~~  
~~removed 删除~~

## 第四章 图片

### 4.1 简述

L<sup>A</sup>T<sub>E</sub>X 本身不支持插图功能，需要借助 graphicx 宏包。

使用 latex + dvipdfmx 编译命令时，调用 graphicx 宏包时要指定 dvipdfmx 选项<sup>1</sup>；而使用 pdflatex 或 xelatex 命令编译时不需要。

下表给出了不同编译命令支持的图片格式：

表 4.1: 各种编译方式支持的主流图片格式

格式	矢量图	位图
latex + dvipdfmx	.eps	N/A
（调用 bmpsize 宏包）	.eps .pdf	.jpg .png .bmp
pdflatex	.pdf	.jpg .png
（调用 epstopdf 宏包）	.pdf .eps	.jpg .png
xelatex	.pdf .eps	.jpg .png .bmp

注：在较新的 T<sub>E</sub>X 发行版中，latex + dvipdfmx 和 pdflatex 命令可不依赖宏包，支持原来需要宏包扩展的图片格式（但 pdflatex 命令仍不支持.bmp 格式的位图）。

引入 graphicx 宏包后，可使用 `\includegraphics` 命令插入图片，其语法为：

```
\includegraphics[options]{file} % options可指定图片属性，如width=
```

其中，options 可选参数有：

表 4.2: `\includegraphics` 命令的可选参数

参数	含义
width=width	将图片缩放到宽度为 width
height=height	将图片缩放到高度为 height
scale=scale	将图片相对于原尺寸缩放 scale 倍
angle=angle	将图片逆时针旋转 angle 度

### 4.2 浮动体

浮动体将图、表与其标题定义为整体，然后动态排版，以解决图、表卡在换页处造成的过长的垂直空白的问题。图片的浮动体是 figure。

下面是一个例子：

```
\begin{figure}[htbp]
 \centering
 \includegraphics[width=0.8\textwidth]{example-image}
 \caption{This is an example image.}
 \label{fig:example-image}
\end{figure}
```

参数 htpb 表示浮动体的位置，h 表示插入此处、t 表示在页面上端，b 表示在页面下端、p 表示允许浮动体单开一页。

<sup>1</sup>早期常使用 latex + dvips 组合命令，后者将.dvi 文件转为.ps 文件（PostScript），可进一步通过 ps2pdf 工具生成 PDF。dvips 和 dvipdfmx 在图形、颜色、超链接等功能的实现上有差别，而 L<sup>A</sup>T<sub>E</sub>X 无法识别用户是用 dvips 还是 dvipdfmx，所以要指定选项（缺省为 dvips）。

## 第五章 表格

### 5.1 基本表格

表格和图片类似，也有一个浮动体环境为 `table`。

表格最基本的环境为 `tabular`，用法如下：

```
\begin{tabular}[可选参数]{|c|c|c|}
\hline
第一列 & 第二列 & 第三列 \\
\hline
1 & 2 & 3 \\
4 & 5 & 6 \\
\hline
\end{tabular}
```

其中 `|c|c|c|` 是列格式标记，详细见 5.1.2，`c` 表示列居中，`|` 表示列之间有竖线。使用 `&` 用来分割列，使用 `\\` 表示换行。`\hline` 用来绘制行之间的横线。

#### 5.1.1 修改表格线

可在 `tabular` 环境外修改全部表格线的粗细，如 `\setlength{\arrayrulewidth}{2pt}` 或 `\arrayrulewidth=2pt`。

如果需要单独修改表格线，可采用如下方法：

1) 修改垂直表格线，使用 `array` 宏包提供的新列格式选项定义命令：

```
\newcolumntype{新选项名称}[参数数量]{列格式}
\newcolumntype{I}{!{\vrule width 2pt}}
```

```
\centering
\newcolumntype{I}{!{\vrule width 2pt}}
\begin{tabular}{|c|c|c|}
 \hline
 \multicolumn{3}{|c|}{垂直线粗细更改} \\
 \hline
 7 & 5 & 3 \\
 \hline
 6 & 1 & 8 \\
 \hline
\end{tabular}
```

垂直线粗细更改		
7	5	3
6	1	8

2) 修改水平表格线，可使用 `booktabs` 宏包，该宏包可以任意修改水平线粗细，还可以在上下附加一段垂直空白。

```

\centering
\begin{tabular}{|c|c|c|}
\hline
\multicolumn{3}{|c|}{水平表格线粗细更改} \\
\specialrule{2pt}{0pt}{0pt}
7 & 5 & 3 \\
\hline
6 & 1 & 8 \\
\hline
\end{tabular}

```

水平表格线粗细更改		
7	5	3
6	1	8

## 5.1.2 列格式

基本列格式如下表所示：

表 5.1: L<sup>A</sup>T<sub>E</sub>X 表格列格式

列格式	说明
l/c/r	单元格内容左对齐/居中/右对齐，不折行
p{width}	单元格宽度固定为 width，可自动折行
	绘制竖线
@{string}	自定义内容 string

格中每行的单元格数目不能多于列格式里 l/c/r/p 的总数（可以少于这个总数），否则出错。

@ 格式可在单元格前后插入任意的文本，但同时它也消除了单元格前后额外添加的间距。@ 格式可以适当使用以充当“竖线”。特别地，@{} 可直接用来消除单元格前后的间距。

另外 L<sup>A</sup>T<sub>E</sub>X 还提供了简便的将格式参数重复的写法 `*{ n }{ column-spec }`，比如以下两种写法是等效的：

```

\begin{tabular}{|c|c|c|c|c|p{4em}|p{4em}|}
\begin{tabular}{|*{5}{c}|*{2}{p{4em}}|}

```

## 5.1.3 列宽

L<sup>A</sup>T<sub>E</sub>X 本身提供了 tabular\* 环境用来排版定宽表格，但是不太方便使用，比如要用到 @ 格式插入额外命令，令单元格之间的间距为 \fill，但即使这样仍然有瑕疵：

```

\begin{tabular*}{14em}{%
{@{\extracolsep{\fill}}|c|c|c|c|}
\hline
A & B & C & D \\ \hline
a & b & c & d \\ \hline
\end{tabular*}

```

A	B	C	D
a	b	c	d

tabularx 宏包为我们提供了方便的解决方案。它引入了一个 X 列格式，类似 p 列格式，不过会根据表格宽度自动计算列宽，多个 X 列格式平均分配列宽。X 列格式也可以用 array 里的辅助格式修饰对齐方式：



```
% \usepackage{array,tabularx}
\begin{tabularx}{14em}%
 {|*{4}{>{\centering\arraybackslash}X|}}
 \hline
 A & B & C & D \\ \hline
 a & b & c & d \\ \hline
\end{tabularx}
```

A	B	C	D
a	b	c	d

### 5.1.4 行距

修改参数 `\arraystretch` 可以得到行距更加宽松的表格：

```
\renewcommand\arraystretch{1.8}
```

另一种增加间距的办法是给换行命令 `\\` 添加可选参数，在这一行下面加额外的间距，适合用于在行间不加横线的表格：

### 5.1.5 表格标题

表格标题可以用 `\caption` 命令设置，默认只能在浮动体环境内部使用。

也可以在导言区添加如下命令，在浮动体外使用 `\figcaption` 和 `\tabcaption` 为图表添加标题。为了防止标题和图表不在一页，可以使用 `minipage` 环境将它们包起来。

```
\makeatletter
\newcommand\figcaption{def\@capttype{figure}\caption}
\newcommand\tabcaption{def\@capttype{table}\caption}
\makeatother
```

## 5.2 复杂表格

### 5.2.1 合并单元格

1) 跨列

使用 `\multicolumn` 命令可以合并列：

```
\multicolumn{合并列数目}{列格式}{内容}
```

2) 跨行

跨行需要引入 `multirow` 宏包，使用 `\multirow` 命令：

```
\multirow{合并行数目}{宽度}{内容} %宽度可以填*以使用自然宽度
```

既跨行又跨列时，需要把 `\multirow` 命令放在 `\multicolumn` 命令内部。

```

\centering
\begin{center}
 \begin{tabular}{|c|c|c|}
 \hline
 \multirow{2}{2cm}{A Text!}
 & ABC & DEF \\
 \cline{2-3}
 & abc & def \\
 \hline
 \multicolumn{2}{|c|}
 {\multirow{2}{*{Nothing}}} & XYZ \\
 \multicolumn{2}{|c|}{} & xyz \\
 \hline
 \end{tabular}
\end{center}

```

A Text!	ABC	DEF
	abc	def
Nothing		XYZ
		xyz

## 5.2.2 单元格旋转

使用`\rotatebox`命令可以旋转单元格内容。

```

\begin{tabular}{|c|c|c|}
 \hline
 \rotatebox{90}{A} & B & C \\
 \hline
 D & \rotatebox{45}{E} & F \\
 \hline
 \rotatebox[origin=l]{90}{G} & H & I \\
 \hline
\end{tabular}

```

⤴	B	C
D	↷	F
⤵	H	I

## 5.3 问题

### 5.3.1 longtable 与 arydshln 宏包冲突

**问题：**最近帮人 debug，使用 `longtable` 排版报错。

**解决方法：**参考 [LaTeX: arydshln 与 longtable 的冲突及教训 - 知乎 \(zhihu.com\)](https://zhuanlan.zhihu.com/p/100000000)，发现 `longtable` 与 `arydshln` 宏包存在冲突，`arydshln` 宏包重定义了 `\hline`，可以采用如下解决方法：

1. 注释掉 `\usepackage{arydshln}`。
2. 将 `\usepackage{arydshln}` 放于 `\usepackage{longtable}` 之后

### 5.3.2 cline 命令 undefined control sequence 报错

**问题：**使用 Springer 模板制作表格，使用 `\cline` 命令报错，提示 `undefined control sequence`。

**解决方法：**Springer 模板重新定义了 `\cline` 命令，找到 Springer 模板的 `sn-jnl.cls` 文件，将下面一行注释掉：

```
\let\cline\cmidrule
```

```

96 \onecolumn
97
98 {\tiny
99 \begin{longtable}{|c|c|c|c|c|}
100 \caption{caption }\label{tab:mytable} \\
101 \hline
102 1 & 2 & 3 & 4 & 5 \\
103 \hline
104 6 & 7 & 8 & 9 & 10 \\
105 \hline
106 11 & 12 & 13 & 14 & 15 \\
107 \end{longtable}
108 }
109
110 \end{document}
111

```

问题 12 输出 调试控制台 终端 端口

TeX 111.tex 8

- ⊗ Undefined control sequence. LaTeX [行 101, 列 1] ^  
`\adl@@cr ...tempdima \xdef \adl@rowsL {\adl@rowsL`
- ⊗ Undefined control sequence. LaTeX [行 101, 列 1] ^  
`\adl@@cr ...ef \adl@rowsL {\adl@rowsL {\adl@colsL`
- ⊗ Undefined control sequence. LaTeX [行 101, 列 1] ^  
`\adl@@cr ...a )\fi }\xdef \adl@rowsR {\adl@rowsR`
- ⊗ Undefined control sequence. LaTeX [行 101, 列 1] ^

图 5.1: 报错信息

### 5.3.3 multirow 宏包居中

**问题:** 列格式设置为居中, 但是使用 `multirow` 多行合并内容不会居中

**解决方法:** 添加如下命令:

```
\renewcommand{\multirowsetup}{\centering}
```

## 第六章 数学公式

### 6.1 行内公式

行内公式使用`\$`和`\$`包裹，如：

```
行内公式测试$a=b$
```

行内公式测试  $a = b$

### 6.2 行间公式

行间公式使用`\begin{equation}`和`\end{equation}`包裹，如：

```
\begin{equation}
 a+b=c
\end{equation}
```

$$a + b = c \quad (6.1)$$

`equation` 环境为公式自动生成一个编号，这个编号可以用 `label` 和 `ref` 生成交叉引用，`amsmath` 的 `eqref` 命令甚至为引用自动加上圆括号；还可以用 `tag` 命令手动修改公式的编号，或者用 `notag` 命令取消为公式编号（与之基本等效的命令是 `nonumber`）。

```
The Pythagorean theorem is:
\begin{equation}
 a^2 + b^2 = c^2 \label{pythagorean}
\end{equation}
Equation \eqref{pythagorean} is
called 'Gougu theorem' in Chinese.

It's wrong to say
\begin{equation}
 1 + 1 = 3 \tag{dumb}
\end{equation}
or
\begin{equation}
 1 + 1 = 4 \notag
\end{equation}
```

The Pythagorean theorem is:

$$a^2 + b^2 = c^2 \quad (6.2)$$

Equation (6.2) is called 'Gougu theorem' in Chinese.

It's wrong to say

$$1 + 1 = 3 \quad (\text{dumb})$$

or

$$1 + 1 = 4$$

使用`\begin{equation*}`和`\end{equation*}`包裹的公式不带编号，如：

```
\begin{equation*}
 a+b=c
\end{equation*}
```

$$a + b = c$$



2. 蓝色的命令依赖 `amsmath` 宏包（非 `amssymb` 宏包）；
3. 带有角标 <sup>$\ell$</sup> 的符号命令依赖 `latexsym` 宏包。

## 6.4.1 L<sup>A</sup>T<sub>E</sub>X 普通符号

### 6.4.1.1 文本/数学模式通用符号

表 6.1: 文本/数学模式通用符号

这些符号可用于文本和数学模式。

{	\{	}	\}	\$	\\$	%	\%
†	\dag	§	\S	©	\copyright	...	\dots
‡	\ddag	¶	\P	£	\pounds		

### 6.4.1.2 希腊字母

表 6.2: 希腊字母

$\alpha$	\alpha	$\theta$	\theta	$o$	o	$\upsilon$	\upsilon
$\beta$	\beta	$\vartheta$	\vartheta	$\pi$	\pi	$\phi$	\phi
$\gamma$	\gamma	$\iota$	\iota	$\varpi$	\varpi	$\varphi$	\varphi
$\delta$	\delta	$\kappa$	\kappa	$\rho$	\rho	$\chi$	\chi
$\epsilon$	\epsilon	$\lambda$	\lambda	$\varrho$	\varrho	$\psi$	\psi
$\varepsilon$	\varepsilon	$\mu$	\mu	$\sigma$	\sigma	$\omega$	\omega
$\zeta$	\zeta	$\nu$	\nu	$\varsigma$	\varsigma		
$\eta$	\eta	$\xi$	\xi	$\tau$	\tau		
$\Gamma$	\Gamma	$\Lambda$	\Lambda	$\Sigma$	\Sigma	$\Psi$	\Psi
$\Delta$	\Delta	$\Xi$	\Xi	$\Upsilon$	\Upsilon	$\Omega$	\Omega
$\Theta$	\Theta	$\Pi$	\Pi	$\Phi$	\Phi		
$\varGamma$	\varGamma	$\varLambda$	\varLambda	$\varSigma$	\varSigma	$\varPsi$	\varPsi
$\varDelta$	\varDelta	$\varXi$	\varXi	$\varUpsilon$	\varUpsilon	$\varOmega$	\varOmega
$\varTheta$	\varTheta	$\varPi$	\varPi	$\varPhi$	\varPhi		



## 6.4.1.3 二元关系符

表 6.3: 二元关系符

所有的二元关系符都可以加 `not` 前缀得到相反意义的关系符，例如 `not=` 就得到不等号（同 `ne`）。

$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$=$	<code>=</code>
$\leq$	<code>\leq</code> or <code>\le</code>	$\geq$	<code>\geq</code> or <code>\ge</code>	$\equiv$	<code>\equiv</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\doteq$	<code>\doteq</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset</code> <sup>ℓ</sup>	$\sqsupset$	<code>\sqsupset</code> <sup>ℓ</sup>	$\Join$	<code>\Join</code> <sup>ℓ</sup>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$ , $\owns$	<code>\ni</code> , <code>\owns</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$ $	<code>\mid</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>
$\smile$	<code>\smile</code>	$\frown$	<code>\frown</code>	$\asymp$	<code>\asymp</code>
$:$	<code>:</code>	$\notin$	<code>\notin</code>	$\neq$	<code>\neq</code> or <code>\ne</code>

## 6.4.1.4 二元运算符

表 6.4: 二元运算符

+	+	-	-	
±	\pm	∓	\mp	◀ \triangleleft
⋅	\cdot	÷	\div	▶ \triangleright
×	\times	\	\setminus	★ \star
∪	\cup	∩	\cap	* \ast
⊔	\sqcup	⊓	\sqcap	◦ \circ
∨	\vee, \lor	∧	\wedge, \land	● \bullet
⊕	\oplus	⊖	\ominus	◊ \diamond
⊙	\odot	⊘	\oslash	⊕ \uplus
⊗	\otimes	◯	\bigcirc	⧿ \amalg
Δ	\bigtriangleup	▽	\bigtriangledown	† \dagger
◁	\lhd^ℓ	▷	\rhd^ℓ	‡ \ddagger
⊲	\unlhd^ℓ	⊳	\unrhd^ℓ	⌵ \wr

## 6.4.1.5 巨算符

表 6.5: 巨算符

$\Sigma$	$\sum$	<code>\sum</code>	$\cup$	$\bigcup$	<code>\bigcup</code>	$\vee$	$\bigvee$	<code>\bigvee</code>
$\prod$	$\prod$	<code>\prod</code>	$\cap$	$\bigcap$	<code>\bigcap</code>	$\wedge$	$\bigwedge$	<code>\bigwedge</code>
$\coprod$	$\coprod$	<code>\coprod</code>	$\sqcup$	$\bigsqcup$	<code>\bigsqcup</code>	$\oplus$	$\bigoplus$	<code>\bigoplus</code>
$\oint$	$\oint$	<code>\oint</code>	$\oint$	$\oint$	<code>\oint</code>	$\odot$	$\bigodot$	<code>\bigodot</code>
$\bigoplus$	$\bigoplus$	<code>\bigoplus</code>	$\bigotimes$	$\bigotimes$	<code>\bigotimes</code>			
$\iiint$	$\iiint$	<code>\iiint</code>	$\iiint$	$\iiint$	<code>\iiint</code>	$\iiint$	$\iiint$	<code>\iiint</code>
$\int \cdots \int$	$\int \cdots \int$	<code>\int \cdots \int</code>						

## 6.4.1.6 数学重音符号

表 6.6: 数学重音符号

最后一个 `\wideparen` 依赖 `yhmath` 宏包。

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
$\acute{a}$	<code>\acute{a}</code>	$\grave{a}$	<code>\grave{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\mathring{a}$	<code>\mathring{a}</code>
$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\overset{\cdot}{a}$	<code>\overset{\cdot}{a}</code>
$\overset{\cdot}{a}$	<code>\overset{\cdot}{a}</code>				
$\widehat{AAA}$	<code>\widehat{AAA}</code>	$\widetilde{AAA}$	<code>\widetilde{AAA}</code>	$\wideparen{AAA}$	<code>\wideparen{AAA}</code>

## 6.4.1.7 箭头

表 6.7: 箭头

$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>
$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Longleftarrow$	<code>\Longleftarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$	<code>\iff</code>
$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\Downarrow$	<code>\Downarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\nearrow$	<code>\nearrow</code>	$\searrow$	<code>\searrow</code>
$\swarrow$	<code>\swarrow</code>	$\nwarrow$	<code>\nwarrow</code>
$\leadsto$	<code>\leadsto</code>		

## 6.4.1.8 作为重音的箭头符号

表 6.8: 作为重音的箭头符号

$\overrightarrow{AB}$	<code>\overrightarrow{AB}</code>	$\overleftarrow{AB}$	<code>\overleftarrow{AB}</code>
$\overleftarrow{AB}$	<code>\overleftarrow{AB}</code>	$\overleftrightarrow{AB}$	<code>\overleftrightarrow{AB}</code>
$\overleftrightarrow{AB}$	<code>\overleftrightarrow{AB}</code>	$\underrightarrow{AB}$	<code>\underrightarrow{AB}</code>
$\underrightarrow{AB}$	<code>\underrightarrow{AB}</code>	$\underleftarrow{AB}$	<code>\underleftarrow{AB}</code>
$\underleftarrow{AB}$	<code>\underleftarrow{AB}</code>	$\underleftrightarrow{AB}$	<code>\underleftrightarrow{AB}</code>
$\underleftrightarrow{AB}$	<code>\underleftrightarrow{AB}</code>		

## 6.4.1.9 定界符

表 6.9: 定界符

`amsmath` 还定义了 `lvert`、`rvert` 和 `lVert`、`rVert`，分别作为 `vert` 和 `Vert` 对应的开符号（左侧）和闭符号（右侧）的命令。

<code>(</code>	<code>(</code>	<code>)</code>	<code>)</code>	$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
<code>[</code>	<code>[</code> or <code>\lbrack</code>	<code>]</code>	<code>]</code> or <code>\rbrack</code>	$\Uparrow$	<code>\Uparrow</code>	$\Downarrow$	<code>\Downarrow</code>
<code>{</code>	<code>\{</code> or <code>\lbrace</code>	<code>}</code>	<code>\}</code> or <code>\rbrace</code>	$\Updownarrow$	<code>\Updownarrow</code>	$\updownarrow$	<code>\updownarrow</code>
<code> </code>	<code> </code> or <code>\vert</code>	<code>  </code>	<code>\ </code> or <code>\Vert</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>	$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>
<code>/</code>	<code>/</code>	<code>\</code>	<code>\backslash</code>				

## 6.4.1.10 用于行间公式的大定界符

表 6.10: 用于行间公式的大定界符

$\left(\left(\right.\right)$	<code>\lggroup</code>	$\right.\right)$	<code>\rgroup</code>	$\left.\left.\right.\right)$	<code>\lmoustache</code>
$\left \right.$	<code>\arrowvert</code>	$\left.\right $	<code>\Arrowvert</code>	$\left \right $	<code>\bracevert</code>
$\left.\right.\right)$	<code>\rmoustache</code>				

## 6.4.1.11 其他符号

表 6.11: 其他符号

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\mho$	<code>\mho</code>	$\partial$	<code>\partial</code>
$'$	<code>'</code>	$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>
$\nabla$	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\Box$	<code>\Box</code>	$\Diamond$	<code>\Diamond</code>
$\bot$	<code>\bot</code>	$\top$	<code>\top</code>	$\angle$	<code>\angle</code>	$\surd$	<code>\surd</code>
$\diamond$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\neg$	<code>\neg</code> or <code>\lnot</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>

6.4.2  $\mathcal{AMS}$  符号

本小节所有符号依赖 `amssymb` 宏包。

## 6.4.2.1 希腊字母和希伯来字母

表 6.12:  $\mathcal{AMS}$  希腊字母和希伯来字母

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>	$\gimel$	<code>\gimel</code>	$\daleth$	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

6.4.2.2  $\mathcal{AMS}$  二元关系符表 6.13:  $\mathcal{AMS}$  二元关系符

$\lessdot$	<code>\lessdot</code>	$\gtrdot$	<code>\gtrdot</code>	$\doteqdot$	<code>\doteqdot</code>
$\leqslant$	<code>\leqslant</code>	$\geqslant$	<code>\geqslant</code>	$\risingdotseq$	<code>\risingdotseq</code>
$\eqslantless$	<code>\eqslantless</code>	$\eqslantgtr$	<code>\eqslantgtr</code>	$\fallingdotseq$	<code>\fallingdotseq</code>
$\leqq$	<code>\leqq</code>	$\geqq$	<code>\geqq</code>	$\eqcirc$	<code>\eqcirc</code>
$\lll$ or $\llless$	<code>\lll</code> or <code>\llless</code>	$\ggg$	<code>\ggg</code>	$\circeq$	<code>\circeq</code>
$\lesssim$	<code>\lesssim</code>	$\gtrsim$	<code>\gtrsim</code>	$\triangleq$	<code>\triangleq</code>
$\lessapprox$	<code>\lessapprox</code>	$\gtrapprox$	<code>\gtrapprox</code>	$\bumpeq$	<code>\bumpeq</code>
$\lessgtr$	<code>\lessgtr</code>	$\gtrless$	<code>\gtrless</code>	$\Bumpeq$	<code>\Bumpeq</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\gtreqless$	<code>\gtreqless</code>	$\thicksim$	<code>\thicksim</code>
$\lesseqqgtr$	<code>\lesseqqgtr</code>	$\gtreqqless$	<code>\gtreqqless</code>	$\thickapprox$	<code>\thickapprox</code>
$\preccurlyeq$	<code>\preccurlyeq</code>	$\succcurlyeq$	<code>\succcurlyeq</code>	$\approxeq$	<code>\approxeq</code>
$\curlyeqprec$	<code>\curlyeqprec</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>	$\backsim$	<code>\backsim</code>
$\precsim$	<code>\precsim</code>	$\succsim$	<code>\succsim</code>	$\backsimeq$	<code>\backsimeq</code>
$\precapprox$	<code>\precapprox</code>	$\succapprox$	<code>\succapprox</code>	$\vDash$	<code>\vDash</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\Vdash$	<code>\Vdash</code>
$\shortparallel$	<code>\shortparallel</code>	$\Supset$	<code>\Supset</code>	$\Vvdash$	<code>\Vvdash</code>
$\blacktriangleleft$	<code>\blacktriangleleft</code>	$\sqsupset$	<code>\sqsupset</code>	$\backepsilon$	<code>\backepsilon</code>
$\vartriangleright$	<code>\vartriangleright</code>	$\because$	<code>\because</code>	$\varpropto$	<code>\varpropto</code>
$\blacktriangleright$	<code>\blacktriangleright</code>	$\Subset$	<code>\Subset</code>	$\between$	<code>\between</code>
$\trianglerighteq$	<code>\trianglerighteq</code>	$\smallfrown$	<code>\smallfrown</code>	$\pitchfork$	<code>\pitchfork</code>
$\vartriangleleft$	<code>\vartriangleleft</code>	$\shortmid$	<code>\shortmid</code>	$\smallsmile$	<code>\smallsmile</code>
$\trianglelefteq$	<code>\trianglelefteq</code>	$\therefore$	<code>\therefore</code>	$\sqsubset$	<code>\sqsubset</code>

6.4.2.3  $\mathcal{AMS}$  二元运算符表 6.14:  $\mathcal{AMS}$  二元运算符

$\dot{+}$	<code>\dotplus</code>	$\cdot$	<code>\centerdot</code>		
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>	$\div$	<code>\divideontimes</code>
$\doublecup$	<code>\doublecup</code>	$\doublecap$	<code>\doublecap</code>	$\smallsetminus$	<code>\smallsetminus</code>
$\veebar$	<code>\veebar</code>	$\barwedge$	<code>\barwedge</code>	$\doublebarwedge$	<code>\doublebarwedge</code>
$\boxplus$	<code>\boxplus</code>	$\boxminus$	<code>\boxminus</code>	$\circleddash$	<code>\circleddash</code>
$\boxtimes$	<code>\boxtimes</code>	$\boxdot$	<code>\boxdot</code>	$\circledcirc$	<code>\circledcirc</code>
$\intercal$	<code>\intercal</code>	$\circledast$	<code>\circledast</code>	$\times$	<code>\rightthreetimes</code>
$\curlyvee$	<code>\curlyvee</code>	$\curlywedge$	<code>\curlywedge</code>	$\times$	<code>\leftthreetimes</code>

6.4.2.4  $\mathcal{AMS}$  箭头表 6.15:  $\mathcal{AMS}$  箭头

$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\rightrightarrows$	<code>\rightrightarrows</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\rightleftarrows$	<code>\rightleftarrows</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>
$\twoheadleftarrow$	<code>\twoheadleftarrow</code>	$\twoheadrightarrow$	<code>\twoheadrightarrow</code>
$\leftarrowtail$	<code>\leftarrowtail</code>	$\rightarrowtail$	<code>\rightarrowtail</code>
$\leftrightharpoons$	<code>\leftrightharpoons</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>
$\Lsh$	<code>\Lsh</code>	$\Rsh$	<code>\Rsh</code>
$\looparrowleft$	<code>\looparrowleft</code>	$\looparrowright$	<code>\looparrowright</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>
$\circlearrowleft$	<code>\circlearrowleft</code>	$\circlearrowright$	<code>\circlearrowright</code>
$\multimap$	<code>\multimap</code>	$\upuparrows$	<code>\upuparrows</code>
$\downdownarrows$	<code>\downdownarrows</code>	$\upharpoonleft$	<code>\upharpoonleft</code>
$\upharpoonright$	<code>\upharpoonright</code>	$\downharpoonright$	<code>\downharpoonright</code>
$\rightsquigarrow$	<code>\rightsquigarrow</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>

6.4.2.5  $\mathcal{AMS}$  反义二元关系符和箭头表 6.16:  $\mathcal{AMS}$  反义二元关系符和箭头

$\nless$	<code>\nless</code>	$\ngtr$	<code>\ngtr</code>	$\varsubsetneqq$	<code>\varsubsetneqq</code>
$\lneq$	<code>\lneq</code>	$\gneq$	<code>\gneq</code>	$\varsupsetneqq$	<code>\varsupsetneqq</code>
$\nleq$	<code>\nleq</code>	$\ngeq$	<code>\ngeq</code>	$\nsubseteq$	<code>\nsubseteq</code>
$\nleqslant$	<code>\nleqslant</code>	$\ngeqslant$	<code>\ngeqslant</code>	$\nsupseteq$	<code>\nsupseteq</code>
$\lneqq$	<code>\lneqq</code>	$\gneqq$	<code>\gneqq</code>	$\nmid$	<code>\nmid</code>
$\lvertneqq$	<code>\lvertneqq</code>	$\gvertneqq$	<code>\gvertneqq</code>	$\nparallel$	<code>\nparallel</code>
$\nleqq$	<code>\nleqq</code>	$\ngeqq$	<code>\ngeqq</code>	$\nshortmid$	<code>\nshortmid</code>
$\lnsim$	<code>\lnsim</code>	$\gnsim$	<code>\gnsim</code>	$\nshortparallel$	<code>\nshortparallel</code>
$\lnapprox$	<code>\lnapprox</code>	$\gnapprox$	<code>\gnapprox</code>	$\nsim$	<code>\nsim</code>
$\nprec$	<code>\nprec</code>	$\nsucc$	<code>\nsucc</code>	$\ncong$	<code>\ncong</code>
$\npreceq$	<code>\npreceq</code>	$\nsucceq$	<code>\nsucceq</code>	$\nvdash$	<code>\nvdash</code>
$\precneqq$	<code>\precneqq</code>	$\succneqq$	<code>\succneqq</code>	$\nvDash$	<code>\nvDash</code>
$\precnsim$	<code>\precnsim</code>	$\succnsim$	<code>\succnsim</code>	$\nVdash$	<code>\nVdash</code>
$\precnapprox$	<code>\precnapprox</code>	$\succnapprox$	<code>\succnapprox</code>	$\nVDash$	<code>\nVDash</code>
$\subsetneq$	<code>\subsetneq</code>	$\supsetneq$	<code>\supsetneq</code>	$\ntriangleleft$	<code>\ntriangleleft</code>
$\varsubsetneq$	<code>\varsubsetneq</code>	$\varsupsetneq$	<code>\varsupsetneq</code>	$\ntriangleright$	<code>\ntriangleright</code>
$\nsubseteq$	<code>\nsubseteq</code>	$\nsupseteq$	<code>\nsupseteq</code>	$\ntrianglelefteq$	<code>\ntrianglelefteq</code>
$\subsetneqq$	<code>\subsetneqq</code>	$\supsetneqq$	<code>\supsetneqq</code>	$\ntrianglerighteq$	<code>\ntrianglerighteq</code>
$\nleftarrow$	<code>\nleftarrow</code>	$\nrightarrow$	<code>\nrightarrow</code>	$\nleftrightarrow$	<code>\nleftrightarrow</code>
$\nLeftarrow$	<code>\nLeftarrow</code>	$\nRightarrow$	<code>\nRightarrow</code>	$\nLeftrightarrow$	<code>\nLeftrightarrow</code>

### 6.4.2.6 $\mathcal{AMS}$ 定界符

表 6.17:  $\mathcal{AMS}$  定界符

$\lrcorner$	<code>\ulcorner</code>	$\urcorner$	<code>\urcorner</code>	$\llcorner$	<code>\llcorner</code>	$\lrcorner$	<code>\lrcorner</code>
-------------	------------------------	-------------	------------------------	-------------	------------------------	-------------	------------------------

### 6.4.2.7 $\mathcal{AMS}$ 其它符号

表 6.18:  $\mathcal{AMS}$  其它符号

$\hbar$	<code>\hbar</code>	$\hslash$	<code>\hslash</code>		
$\square$	<code>\square</code>	$\blacksquare$	<code>\blacksquare</code>	$\textcircled{S}$	<code>\circledS</code>
$\triangle$	<code>\vartriangle</code>	$\blacktriangle$	<code>\blacktriangle</code>	$\complement$	<code>\complement</code>
$\nabla$	<code>\triangledown</code>	$\blacktriangledown$	<code>\blacktriangledown</code>	$\Game$	<code>\Game</code>
$\lozenge$	<code>\lozenge</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\bigstar$	<code>\bigstar</code>
$\angle$	<code>\angle</code>	$\measuredangle$	<code>\measuredangle</code>		
$\diagup$	<code>\diagup</code>	$\diagdown$	<code>\diagdown</code>	$\backprime$	<code>\backprime</code>
$\nexists$	<code>\nexists</code>	$\Finv$	<code>\Finv</code>	$\varnothing$	<code>\varnothing</code>
$\eth$	<code>\eth</code>	$\sphericalangle$	<code>\sphericalangle</code>	$\mho$	<code>\mho</code>

## 6.5 多行公式

目前最常用的是 `align` 环境，它将公式用 `&` 隔为两部分并对齐。分隔符通常放在等号左边：

```
\begin{align}
a &= b + c \\
&= d + e
\end{align}
```

$$a = b + c \quad (6.3)$$

$$= d + e \quad (6.4)$$

`align` 环境会给每行公式都编号。我们仍然可以用 `notag` 去掉某行的编号。在以下的例子，为了对齐等号，我们将分隔符放在右侧，并且此时需要在等号后添加一对括号 `{}` 以产生正常的间距：

```
\begin{align}
a &= {} & b + c & \\
&= {} & d + e + f + g + h + i & \\
+ j + k + l &\notag & & \\
&& + m + n + o & \\
&= {} & p + q + r + s &
\end{align}
```

$$a = b + c \quad (6.5)$$

$$= d + e + f + g + h + i + j + k + l + m + n + o \quad (6.6)$$

$$= p + q + r + s \quad (6.7)$$

`align` 还能够对齐多组公式，除等号前的 `&` 之外，公式之间也用 `&` 分隔：

```
\begin{align}
a &= 1 & b &= 2 & c &= 3 \\
d &= -1 & e &= -2 & f &= -5
\end{align}
```

$$a = 1 \quad b = 2 \quad c = 3 \quad (6.8)$$

$$d = -1 \quad e = -2 \quad f = -5 \quad (6.9)$$

如果我们不需要按等号对齐，只需罗列数个公式，`gather` 将是一个很好用的环境：

```
\begin{gather}
 a = b + c \\
 d = e + f + g \\
 h + i = j + k \notag \\
 l + m = n
\end{gather}
```

$$a = b + c \quad (6.10)$$

$$d = e + f + g \quad (6.11)$$

$$h + i = j + k$$

$$l + m = n \quad (6.12)$$

`align` 和 `gather` 有对应的不带编号的版本 `align*` 和 `gather*`。

### 6.5.1 公用编号的多行公式

另一个常见的需求是将多个公式组在一起公用一个编号，编号位于公式的居中位置。为此，`amsmath` 宏包提供了诸如 `aligned`、`gathered` 等环境，与 `equation` 环境套用。以 `-ed` 结尾的环境用法与前一节不以 `-ed` 结尾的环境用法一一对应。我们仅以 `aligned` 举例：

```
\begin{equation}
 \begin{aligned}
 a &= b + c \\
 d &= e + f + g \\
 h + i &= j + k \\
 l + m &= n
 \end{aligned}
\end{equation}
```

$$a = b + c$$

$$d = e + f + g \quad (6.13)$$

$$h + i = j + k$$

$$l + m = n$$

`split` 环境和 `aligned` 环境用法类似，也用于和 `equation` 环境套用，区别是 `split` 只能将每行的一个公式分两栏，`aligned` 允许每行多个公式多栏。

## 6.6

## 第七章 参考文献设置

### 7.1 问题

#### 7.1.1 不按出现顺序引用

**问题：**Latex 中 ACM-Reference-Format 顺序与论文引用顺序不一致

**解决方法：**修改ACM-Reference-Format.bst文件，将大写 SORT 注释掉，一共两处。

#### 7.1.2 作者年份引用设置仅年份超链接

**问题：**Latex 中采用作者年份引用，默认作者和年份都出现超链接，设置仅年份出现超链接

**解决方法：**在\begin{document}前加入如下代码：

```
\makeatletter
% Patch case where name and year are separated by aysep
\patchcmd{\NAT@citex}
 {\@citea\NAT@hyper@{%
 \NAT@nmfmt{\NAT@nm}%
 \hyper@natlinkbreak{\NAT@aysep\NAT@spacechar}{\@citeb\@extra@b@citeb}%
 \NAT@date}}
 {\@citea\NAT@nmfmt{\NAT@nm}%
 \NAT@aysep\NAT@spacechar\NAT@hyper@{\NAT@date}}{-}{-}

% Patch case where name and year are separated by opening bracket
\patchcmd{\NAT@citex}
 {\@citea\NAT@hyper@{%
 \NAT@nmfmt{\NAT@nm}%
 \hyper@natlinkbreak{\NAT@spacechar\NAT@@open\if*#1*\else#1\NAT@spacechar\fi}%
 {\@citeb\@extra@b@citeb}%
 \NAT@date}}
 {\@citea\NAT@nmfmt{\NAT@nm}%
 \NAT@spacechar\NAT@@open\if*#1*\else#1\NAT@spacechar\fi\NAT@hyper@{\NAT@date}}
 {}{}

\makeatother
```

#### 7.1.3 作者年份引用设置 et al 为斜体

**问题：**Latex 中采用作者年份引用，设置 et al 为斜体

**解决方法：**在相应.bst文件中将et~al.修改为\textit{et~al.}，不要修改et al

#### 7.1.4 beamer 参考文献断行

**问题：**beamer 参考文献断行显示，如图7.1所示。

**解决方法：**在导言区加入如下代码：



## References

- [1] Bin Li, J Friedman, R Olshen, and C Stone.  
Classification and regression trees (cart).  
*Biometrics*, 40(3):358–361, 1984.
- [2] World Health Organization.  
Global Status Report on Road.  
Technical report, W.H.O, 2018.

图 7.1: beamer 参考文献断行显示

```
\setbeamertemplate{bibliography entry title}{}
\setbeamertemplate{bibliography entry location}{}
\setbeamertemplate{bibliography entry note}{}

```

## References

- [1] Bin Li, J Friedman, R Olshen, and C Stone. Classification and regression trees (cart). *Biometrics*, 40(3):358–361, 1984.
- [2] World Health Organization. Global Status Report on Road. Technical report, W.H.O, 2018.

图 7.2: beamer 参考文献正常显示

### 7.1.5 Template for the Computer Journal (COMJNL)使用 bibtex 编译报错

**问题：**使用Template for the Computer Journal (COMJNL)模板时，使用bibtex编译报错，报错信息如下：

```
You can't pop an empty literal stack for entry CAN01
```

**解决方法** compj.bst文件有错误，将

```
FUNCTION {article}
{ output.bibitem
 format.authors "author" output.check
 http://format.date "year" output.check
 format.title "title" output.check
 new.block
 crossref missing$
 { journal emphasize "journal" output.check
 format.vol.num.pages output
 }
 { format.article.crossref output.nonnull
 format.pages output
 }
 if$
 format.note output
 fin.entry
}
```

中 `http://format.date "year" output.check` 改为 `format.date "year" output.check` 即可。其余文献类型类似。

## 第八章 类的编写

### 8.1 文档类和宏包的结构

文档类或宏包文件的大纲如下：

- **标识**文件声明自己是一个 LATEX2 $\epsilon$  宏包或文档类，并简要描述自身。
- **初步声明**在这里，文件声明一些命令，并可以加载其他文件。通常，这些命令将只是声明所用选项中需要的代码。
- **选项**文件声明并处理其选项。
- **更多声明**这是文件执行大部分工作的地方：声明新变量、命令和字体；以及加载其他文件。

#### 8.1.1 标识

文档类或宏包文件的第一件事是标识自己。文档类文件的标识如下：

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{myclass}[2019/01/01 v1.0 My Class]
```

#### 8.1.2 使用类和宏包

一个 LaTeX 宏包或文档类可以加载另一个宏包，方法如下：

```
\RequirePackage[options]{otherpackage}[date]
```

一个 LATEX 文档类可以加载另一个类，方法如下：

```
\LoadClass[options]{otherclass}[date]
```

以下命令可以在常见情况下使用，即您希望简单地加载一个具有当前类所使用的确切选项的类或宏包文件。

```
\LoadClassWithOptions{<class-name>}[<date>]
\RequirePackageWithOptions{<package>}[<date>]
```

#### 8.1.3 声明选项

宏包和文档类可以声明选项，作者可以指定这些选项；例如，twocolumn 选项由 article 类声明。注意，选项的名称应仅包含“LATEX 名称”中允许的字符；特别地，不能包含任何控制序列。

LATEX 支持两种创建选项的方法：键-值系统和“简单文本”方式。键-值系统推荐用于新的类和宏包，并且在处理选项类方面比简单文本方式更灵活。这两种选项方法在 LATEX 源文件中使用相同的基本结构：首先声明选项，然后在第二步处理选项。两者都允许将选项传递给其他宏包或底层类。

##### 8.1.3.1 简单文本方式

声明一个选项如下：

```
\DeclareOption{<option>}{<code>}
```

例如，graphics 宏包中的 dvips 选项（略作简化）的实现如下：

```
\DeclareOption{dvips}{\input{dvips.def}}
```

这意味着当作者写 `\usepackage[dvips]{graphics}` 时，文件 `dvips.def` 会被加载。

再举个例子，`article` 类中声明了 `a4paper` 选项以设置 `\paperheight` 和 `\paperwidth` 长度：

```
\DeclareOption{a4paper}{%
 \setlength{\paperheight}{297mm}%
 \setlength{\paperwidth}{210mm}%
}
```

有时用户会请求类或宏包未明确声明的选项。默认情况下，这将产生警告（对于类）或错误（对于宏包）；此行为可以通过以下方式更改：

```
\DeclareOption*{<code>}
```

例如，为使宏包 `fred` 对未知选项产生警告而不是错误，您可以指定：

```
\DeclareOption*{%
 \PackageWarning{fred}{Unknown option `'\CurrentOption'}%
}
```

这样，如果作者写了 `\usepackage[foo]{fred}`，他们会收到警告 `Package fred Warning: Unknown option 'foo'`。

\* 表示所有字符匹配？

可以使用 `\PassOptionsToPackage` 或 `\PassOptionsToClass` 命令将选项传递给另一个宏包或类（请注意，这是一种专门的操作，仅适用于选项名称），例如，要将每个未知选项传递给 `article` 类，可以使用：

```
\DeclareOption*{%
 \PassOptionsToClass{\CurrentOption}{article}%
}
```

如果这样做，您应该确保在稍后某个时刻加载该类，否则选项将永远不会被处理！

到目前为止，我们只解释了如何声明选项，而不是如何执行它们。要处理文件调用时使用的选项，应使用：

```
\ProcessOptions\relax
```

### 8.1.3.2 键-值系统

## 8.2 一个最小的文档类文件

一个文档类或宏包的大部分工作在于定义新命令或更改文档的外观。这是在文档类的主体中完成的，使用诸如 `\newcommand` 或 `\setlength` 等命令。

每个文档类文件必须包含四个内容：定义 `\normalsize`、设置 `\textwidth` 和 `\textheight` 的值，以及指定页码。因此，一个最小的文档类文件<sup>1</sup> 看起来像这样：

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{minimal}[2022-06-01 Standard LaTeX minimal class]
\renewcommand{\normalsize}{\fontsize{10pt}{12pt}\selectfont}
\setlength{\textwidth}{6.5in}
\setlength{\textheight}{8in}
\pagenumbering{arabic} % 即使这个类不显示页码也需要
```

<sup>1</sup>这个类现在已经在标准发行版中，名为 `minimal.cls`。

然而，这个类文件不支持脚注、边注、浮动对象等，也不提供像 `\rm` 这样的两个字母的字体命令；因此，大多数文档类会包含比这更多的内容！

### 8.2.1 示例：一个新闻简报类

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{simplnews}[2024-10-23 The eimple Newsletter class]

\newcommand{\headlinecolor}{\normalcolor}

% 将大多数指定的选项传递给 article 类：除了 onecolumn 选项被关闭，green选项将标题设置为绿色
\DeclareOption{oncolumn}{\OptionNotUsed}
\DeclareOption{green}{\renewcommand{\headlinecolor}{\color{green}}}

\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}

\ProcessOptions\relax

% 然后加载 article 类，使用 twocolumn 选项。
\LoadClass[twocolumn]{article}

% 由于新闻简报将以彩色打印，它现在加载 color 宏包。该类不指定设备驱动程序选项，因为这应由 simplnews 类的使
% 用者指定。
\RequirePackage{color}

% 重新定义 \maketitle，以 72 pt Helvetica 粗斜体显示标题，使用合适的颜色。
\renewcommand{\maketitle}{%
 \twocolumn[
 \fontsize{72}{80}\fontfamily{phv}\fontseries{b}
 \fontshape{sl}\selectfont\headlinecolor
 \@title
]
}

% 重新定义 \section 并关闭章节编号
\renewcommand{\section}{%
 \@startsection{section}{1}{0pt}{-1.5ex plus -1ex minus -.2ex}
 {1ex plus .2ex}{\large\sffamily\slshape\headlinecolor}
}

\setcounter{secnumdepth}{0}

% 设置了三个基本要素
\renewcommand{\normalsize}{\fontsize{9}{10}\selectfont}
\setlength{\textwidth}{17.5cm}
\setlength{\textheight}{20cm}
```

## 8.3 用于类和宏包编写者的命令

### 8.3.1 识别

#### 8.3.1.1 NeedsTeXFormat

```
\NeedsTeXFormat { format-name } [release-date]
```

该命令告诉 TEX 应使用名称为 `format-name` 的格式处理此文件。您可以使用可选参数 `release-date` 进一步指定所需格式的最早发布日期。当格式的发布日期早于指定的日期时，将生成警告。标准的 `format-name` 是 LaTeX2e。如果存在日期，则必须采用 `yyyy-mm-dd` 的形式。

#### 8.3.1.2 ProvidesClass 和 ProvidesPackage

```
\ProvidesClass { class-name } [release-info]
\ProvidesPackage { package-name } [release-info]
```

此声明当前文件包含文档类 `class-name` 或宏包 `package-name` 的定义。

可选的 `release-info`，如果使用，必须包含：

- 此版本文件的发布日期，形式为 `yyyy-mm-dd`；
- 可选地跟着一个空格和一个简短描述，可能包括版本号。

上述语法必须严格遵循，以便 `\LoadClass` 或 `\documentclass`（对于类）或 `\RequirePackage` 或 `\usepackage`（对于宏包）测试发布是否太旧。

### 8.3.2 加载文件

#### 8.3.2.1 LoadClass

```
\LoadClass [options-list] { class-name } [release-info]
\LoadClassWithOptions { class-name } [release-info]
```

这些命令仅用于类文件中，不能用于宏包文件；在类文件中最多只能使用一次。

使用 `\LoadClass` 与使用 `\documentclass` 加载类文件相同，示例：

```
\LoadClass{article}[2022-06-01]
\LoadClassWithOptions{article}[2022-06-01]
```

#### 8.3.2.2 RequirePackage

```
\RequirePackage [options-list] { package-name } [release-info]
\RequirePackageWithOptions { package-name } [release-info]
```

使用 `\RequirePackage` 与作者命令 `\usepackage` 相同。示例：

```
\RequirePackage{ifthen}[2022-06-01]
\RequirePackageWithOptions{graphics}[2022-06-01]
```

两个带 `WithOptions` 的版本仅加载具有当前文件（类或宏包）使用的选项的类（或宏包）文件。

### 8.3.3 延迟代码

### 8.3.4 创建和使用键值选项

```
\DeclareKeys [family] {declarations}
```

此命令从逗号分隔的 `declarations` 列表中创建一系列选项。列表中的每个条目都是键-值对，其中 `key` 具有一个或多个 `properties`。下面介绍了一小部分“基本”的 `properties`。

这里提供的基本属性包括：

- `.code` — 执行任意代码
  - `.if` — 设置一个 TEX `\if...` 开关
  - `.ifnot` — 设置一个反转的 TEX `\if...` 开关
  - `.store` — 将值存储到宏中
  - `.usage` — 定义选项是否仅能在加载时使用 (`load`)、在导言区使用 (`preamble`) 或在范围上没有限制 (`general`)
- 在 `property` 之前的 `key` 部分是 `name`，`value` 与 `property` 一起定义了选项的行为。

```
\DeclareKeys[mypkg]
{
 draft.if = @mypkg@draft,
 draft.usage = preamble,
 name.store = \@mypkg@name,
 name.usage = load,
 second-name.store = \@mypkg@other@name
}
```

上述代码将创建三个选项。选项 `draft` 可以在导言区的任何位置给出，并将设置一个名为 `\if@mypkg@draft` 的开关。选项 `name` 只能在加载宏包时给出，并将保存给定的任何值到 `\@mypkg@name` 中。最后，选项 `second-name` 可以在任何位置给出，并将其值保存到 `\@mypkg@other@name` 中。

## 参考资料

1. [TeX 家族 \(TeX, XeTeX, LuaTeX, XeLaTeX ...看完这篇就懂了\)](#)
2. [一份 \(不太\) 简短的 L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 介绍](#)
3. [邹思宇的 L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub> 学习笔记](#)
4. [参考文献全超链接 + 仅年份超链接 | latex 参考文献仅年份超链接-CSDN 博客](#)
5. [BibTeX style with "et al" in italic - TeX - LaTeX Stack Exchange](#)
6. [LaTeX: arydshln 与 longtable 的冲突及教训 - 知乎 \(zhihu.com\)](#)
7. [Windows 不合法的缓存路径导致 TeX Live 安装失败](#)
8. [pifont 宏包可能会触发中文不显示](#)
9. [texlive 安装过程中报错 vars expected but powershell](#)
10. [字体族 | L<sup>A</sup>T<sub>E</sub>X 入门与进阶](#)