LaTeX 快速入门

刘海洋

2020 年 3 月 15 日

谈谈历史

高教授和蓝博士

图: 高德纳（Donald Knuth），Stanford 大学计算机程序设计艺术荣誉教授，Turing 奖得主。为了写他的七卷本著作《The Art of Computer Programming》而编制了 TEX 排版系统。但或许因为在 TEX 上花的十年时间太长，这部著作至今才写到第四卷。

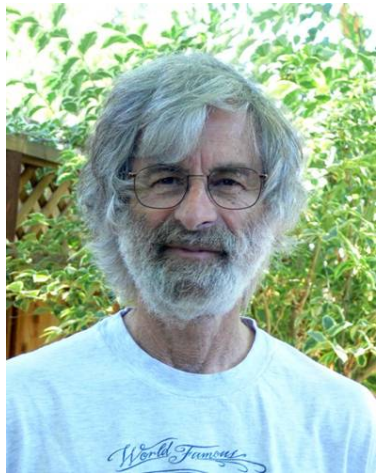


图: Leslie Lamport，微软研究院资深研究员，Turing 奖得主。为了准备他的著作《The Great American Concurrency Book》而编写了一组基于 TEX 的宏，即 LATEX，后交给 LATEX3 小组，逐渐发展演变为现在的样子。但是，那部著作一直没有动笔。

印象里 TEX/LATEX 大概是什么？

- 写毕业论文，据说很方便

印象里 TEX/LATEX 大概是什么？

- 写毕业论文，据说很方便
- 论文投稿要用，别的格式不要

印象里 TEX/LATEX 大概是什么？

- 写毕业论文，据说很方便
- 论文投稿要用，别的格式不要
- 写书的工具，有的老师用它

印象里 TEX/LATEX 大概是什么？

- 写毕业论文，据说很方便
- 论文投稿要用，别的格式不要
- 写书的工具，有的老师用它
- 可以写作业、记笔记，输出 PDF

人们说 TeX/LaTeX 是什么？

TeX 来自 technology 的希腊词根 $\tau\epsilon\chi$，读音 [tɛx]

LaTeX = Lamport TeX，读音 ['lɑːtɛx; 'leitɛx] 或者随便

人们说 TeX/LaTeX 是什么？

TeX 来自 technology 的希腊词根 $\tau\epsilon\chi$，读音 [tɛx]

LaTeX = Lamport TeX，读音 ['lɑːtɛx; 'leitɛx] 或者随便

- TeX 是一种专业排版软件。与它在各方面最为类似的是方正的书版；功能相近而用法不大相同的有方正飞腾创意，Adobe 的 PageMaker、InDesign 等。
- TeX 是一种计算机宏语言。同为宏语言的有 C 语言预处理宏、Linux 下的 M4；但功能和形式更相近的是 HTML+PHP。
- LaTeX 是定义在 TeX 语言上的一大组宏命令，一种格式。它提供了结构化的方式使得书籍文章可以方便地按内容的逻辑结构进行排版。LaTeX 之于 TeX 类似 HTML+CSS 之于基本的 HTML。

人们说 TEX/LATEX 是什么？

- 来自 technology 的希腊词根 $\tau\epsilon\chi$，读音 [tɛx]
- LATEX = Lamport TEX，读音 ['lɑːtɛx; 'leitɛx] 或者随便

- TEX 是一种[illegible]排版软件。与它在各方面最为类[illegible]是方正的书版；功能相近而用法[illegible]同的有方正飞腾创[illegible]Adobe 的 PageMaker、InDesign 等。
- TEX 是一种计算机宏语言。[illegible]言的有 C 语言预处理宏、Linux 下的 M4；但功能和形[illegible]相近的是[illegible]L+PHP。
- LATEX 是定义在 T[illegible]言上的一大组宏命令，[illegible]格式。它提供了结构化的方式[illegible]书籍文章可以方便地按内容的逻辑[illegible]进行排版。LATEX[illegible]TEX 类似 HTML+CSS 之于基本的 HTML。

LaTeX 到底是什么？——从左到右的转换

```
\documentclass{ctexart}
\usepackage{amsmath}
\usepackage{graphicx}
\title{再论商高之勾股定理}
\author{赵爽}

\begin{document}
\maketitle
句股各自乘，併之為弦實，開方除之即弦。
\cite{zhou}
\begin{gather}\label{eq:gougu}
  c = \sqrt{a^2 + b^2}
\end{gather}
% 其中省略若干行
\bibliographystyle{plain}
\bibliography{chinabib}
\end{document}
```

LaTeX 到底是什么？——从左到右的转换

```
\documentclass{ctexart}
\usepackage{amsmath}
\usepackage{graphicx}
\title{再论商高之勾股定理}
\author{赵爽}

\begin{document}
\maketitle
句股各自乘，併之為弦實，開方除之即弦。
\cite{zhou}
\begin{gather}\label{eq:gougu}
  c = \sqrt{a^2 + b^2}
\end{gather}
% 其中省略若干行
\bibliographystyle{plain}
\bibliography{chinabib}
\end{document}
```

再论商高之勾股定理

赵爽

2009 年 4 月 22 日

句股各自乘，併之為弦實，開方除之即弦。[1]

$$c = \sqrt{a^2 + b^2} \tag{1}$$

案: 弦圖又可以句股相乘為朱實二，倍之為朱實四，以句股之差自相乘為中黃實，加差實亦成弦實。

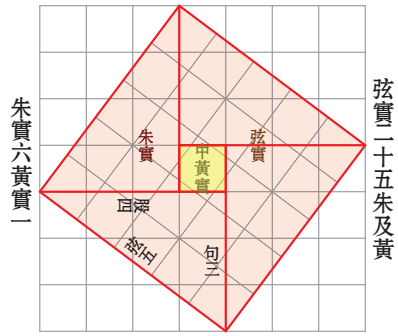


图 1: 弦圖

参考文献

[1] 古之贤人. 周髀算经. 古贤人居所, 古代.

LaTeX 到底是什么？——从左到右的转换

```
\documentclass{ctexart}
\usepackage{amsmath}
\usepackage{graphicx}
\title{再论商高之勾股定理}
\author{赵爽}

\begin{document}
\maketitle
句股各自乘，併之為弦實，開方除之即弦。
\cite{zhou}
\begin{gather}\label{eq:gougu}
  c = \sqrt{a^2 + b^2}
\end{gather}
% 其中省略若干行
\bibliographystyle{plain}
\bibliography{chinabib}
\end{document}
```

格式化的代码 → 排好的文档

再论商高之勾股定理

赵爽

2009 年 4 月 22 日

句股各自乘，併之為弦實，開方除之即弦。[1]

$$c = \sqrt{a^2 + b^2} \tag{1}$$

案: 弦圖又可以句股相乘為朱實二，倍之為朱實四，以句股之差自相乘為中黃實，加差實亦成弦實。

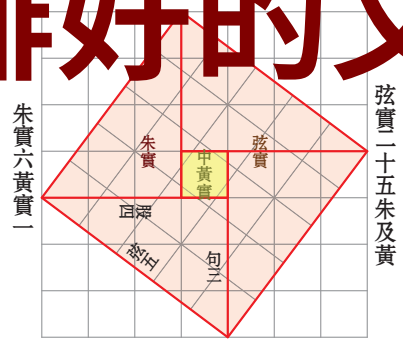


图 1: 弦圖

参考文献

[1] 古之贤人. 周髀算经. 古贤人居所, 古代.

安装并更新 TEX 发行版软件

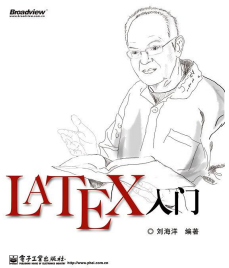
- TEXLive 2019（每年夏天更新），macOS 下称为 MacTeX 2019
- MiKTeX 2.9（Windows）
- 网页在线版 `https://www.overleaf.com/`

各个大学的毕业论文模板可能需要更新 TEX 发行版后才能使用。如果不要求最新，Linux 环境下也可以使用软件源里的版本（APT 大法）。

准备一些靠谱的教程

- 英文：印度 TUG 的 LaTeX Tutorials: A Primer——简明实用
  https://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf
- 中文：黄新刚的 LaTeX Notes——生动有趣
  http://dralpha.altervista.org/zh/tech/lnotes2.pdf
- 英文书籍：A Guide to LaTeX, 4ed（影印版《LaTeX 实用教程》）

- 中文书籍：本人的《LaTeX 入门》



了解从哪儿解决疑难

- 在线手册：在你电脑上用 texdoc 命令调出，或 https://texdoc.net/
- 周围熟悉 LaTeX 的人
- 英文社区：http://tex.stackexchange.com 等
- 中文社区：LaTeX 工作室等

第一部分

组织文档结构

提纲

文档由什么组成?

- 标题
- 前言/摘要
- 目录
- 正文
  - 篇、章、节、小节、小段
    - 文字、公式
    - 列表：编号的、不编号的、带小标题的
    - 定理、引理、命题、证明、结论
    - 诗歌、引文、程序代码、算法伪码
    - 制表
    - 画图
- 文献
- 索引、词汇表

纲举目张

编写结构化文档

LaTeX 支持结构化的文档编写方式，也只有具有良好结构的文档才适合使用 LaTeX 来编写。

纲举目张

编写结构化文档

LaTeX 支持结构化的文档编写方式，也只有具有良好结构的文档才适合使用 LaTeX 来编写。

步骤：

- 拟定主题
- 列出提纲
- 填写内容
- ~~调整格式~~不要在意格式

Markdown：最简单的结构标记语言

- 各级标题：对应于文章章节
- 两种列表：编号、不编号
- 强调文字：弱、强
- 插入代码：行内代码、大段代码
- 插图与链接
- 一些扩展（如数学公式）

Markdown 演示

纯文本

Typora

LyX：结构化的文档写作系统

LyX 是一个图形界面的接近“所见即所得”效果的文档处理软件。LyX 可以模拟 LaTeX 的大部分功能，也可以生成 LaTeX 代码。

LyX 不是 LaTeX 编辑器，它不能编辑任意的 LaTeX 文档代码。

LaTeX：结构化文档语言

可以用任何文本编辑器编写，可以使用专门的编辑器（如 TeXworks）或通用的代码编辑器（如 VS code）。

提纲

LaTeX 文档基本结构

以 document 环境为界，document 环境前是导言部分（preamble）；环境内部是正文部分；环境之后的部分被忽略。

在导言区进行格式设置，正文部分套用格式。

LaTeX 文档基本结构

以 document 环境为界，document 环境前是导言部分（preamble）；环境内部是正文部分；环境之后的部分被忽略。

在导言区进行格式设置，正文部分套用格式。

```
%%% 简单文档
% 导言：格式设置
\documentclass{ctexart}
\usepackage[b5paper]{geometry}
% 正文：填写内容
\begin{document}
使用 \LaTeX
\end{document}
```

文档部件

- 标题：\title, \author, \date ——\maketitle
- 摘要/前言：abstract 环境 / \chapter*
- 目录：\tableofcontents
- 章节：\chapter, \section,...
- 附录：\appendix + \chapter或\section ...
- 文献：\bibliography
- 索引：\printindex

文档划分

大型文档：\frontmatter、\mainmatter、\backmatter

文档划分

大型文档：\frontmatter、\mainmatter、\backmatter

一般文档：\appendix

文档划分

大型文档：\frontmatter、\mainmatter、\backmatter

一般文档：\appendix

层次	名称	命令	说明
−1	part	\part	可选的最高层
0	chapter	\chapter	report, book 类最高层
1	section	\section	article 类最高层
2	subsection	\subsection	
3	subsubsection	\subsubsection	report, book 类默认不编号、不编目录
4	paragraph	\paragraph	默认不编号、不编目录
5	subparagraph	\subparagraph	默认不编号、不编目录

表: 章节层次

文档划分

大型文档：\frontmatter、\mainmatter、\backmatter

一般文档：\appendix

层次	名称	命令	说明
−1	part	\part	可选的最高层
0	chapter	\chapter	report, book 类最高层
1	section	\section	article 类最高层
2	subsection	\subsection	
3	subsubsection	\subsubsection	report, book 类默认不编号、不编目录
4	paragraph	\paragraph	默认不编号、不编目录
5	subparagraph	\subparagraph	默认不编号、不编目录

表: 章节层次

secnumdepth 编号的深度，tocdepth 编目的深度。默认值均为 3。

磁盘文件组织

小文档将所有内容写在同一个目录中。对比较大的文档，可以将文档分成多个文件，并划分文件目录结构：

- 主文档，给出文档框架结构
- 按内容章节划分不同的文件
- 使用单独的类文件和格式文件设置格式
- 用小文件隔离复杂的图表

磁盘文件组织

小文档将所有内容写在同一个目录中。对比较大的文档，可以将文档分成多个文件，并划分文件目录结构：

- 主文档，给出文档框架结构
- 按内容章节划分不同的文件
- 使用单独的类文件和格式文件设置格式
- 用小文件隔离复杂的图表

相关命令：

- `\documentclass`：读入文档类文件（`.cls`）
- `\usepackage`：读入一个格式文件——宏包（`.sty`）
- `\include`：分页，并读入章节文件（`.tex`）
- `\input`：读入任意的文件

文档框架示例

```
% language-main.tex
\documentclass{book}
\usepackage{makeidx}
\makeindex
\title{Languages} \author{someone}
\begin{document}
\frontmatter
\maketitle
\tableofcontents
\mainmatter
\include{intro}
\include{class}
\backmatter
\include{appendix}
\bibliography{foo}
\printindex
\end{document}
```

```
% intro.tex
\part{Introduction}
  \chapter{Background}
```

```
% class.tex
\part{Classification}
 \chapter{Natural Language}
 \chapter{Computer Languages}
  \section{Machine Languages}
  \section{High Level Languages}
   \subsection{Compiled Language}
   \subsection{Interpretative Language}
   \subsubsection{Lisp}
    \paragraph{Common Lisp}
    \paragraph{Scheme}
   \subsubsection{Perl}
```

```
% appendix.tex
  \chapter{Appendix}
```

第二部分

填写文档内容

提纲

迟到的 Hello world.

找个东西输入文本：

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

迟到的 Hello world.

找个东西输入文本：

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

编译代码得到结果：

Hello world.

迟到的 Hello world.

找个东西输入文本：

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

编译代码得到结果：

Hello world.

中文几乎没有改变：

```
\documentclass{ctexart}
\begin{document}
今天你吃了吗?
\end{document}
```

得到：

今天你吃了吗?

迟到的 Hello world.

找个东西输入文本：

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

编译代码得到结果：

Hello world.

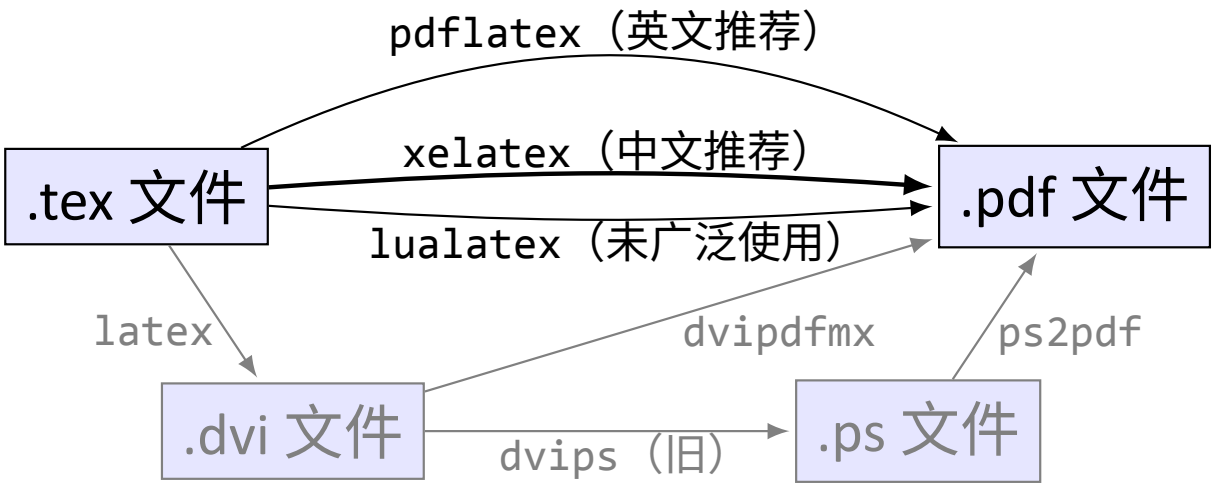
中文几乎没有改变：

```
\documentclass{ctexart}
\begin{document}
今天你吃了吗?
\end{document}
```

得到：

今天你吃了吗?

具体如何编译：



为了生成目录、引用信息，往往需要若干次编译

语法结构

相比原始的 TeX 语言，LaTeX 的语法结构被限制为相对固定的形式。

语法结构

相比原始的 TeX 语言，LaTeX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面花括号表示，用中括号表示可选参数

```
\cmd{arg1}{arg2}\\
\cmd[opt]{arg1}{arg2}
```

LaTeX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

TeX 的分数 $1 \over 2$ `1 \over 2`

语法结构

相比原始的 TeX 语言，LaTeX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面花括号表示，用中括号表示可选参数

```
\cmd{arg1}{arg2}\\
\cmd[opt]{arg1}{arg2}
```

LaTeX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

TeX 的分数 $1 \over 2$ `1 \over 2`

- 环境

```
\begin{env}
  ......
\end{env}
```

LaTeX 的矩阵 `\begin{matrix} ... \\ ... \end{matrix}`

TeX 的矩阵 `\matrix{...\cr ...\cr}`

语法结构

相比原始的 TeX 语言，LaTeX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面花括号表示，用中括号表示可选参数

```
\cmd{arg1}{arg2}\\
\cmd[opt]{arg1}{arg2}
```

LaTeX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

TeX 的分数 $1 \over 2$ `1 \over 2`

- 环境

```
\begin{env}
  ......
\end{env}
```

LaTeX 的矩阵 `\begin{matrix} ... \\ ... \end{matrix}`

TeX 的矩阵 `\matrix{...\cr ...\cr}`

- 注释：以符号 % 开头，该行在 % 后面的部分。

LaTeX 宏：命令与环境

LaTeX 中的宏可分为命令与环境：

LaTeX 宏：命令与环境

LaTeX 中的宏可分为命令与环境：

命令 命令通常以反斜线开头，可以带零到多个参数。命令也可以是直接输出某种结果；也可以改变一个状态，此时 LaTeX 用花括号 `{}` 分组或环境作为状态改变的作用域。
例如 `\em abc` 改变字体以强调一些文字，得到 abc；而带参数的命令 `\emph{abc}` 可得到同样的效果。

LaTeX 宏：命令与环境

LaTeX 中的宏可分为命令与环境：

命令 命令通常以反斜线开头，可以带零到多个参数。命令也可以是直接输出某种结果；也可以改变一个状态，此时 LaTeX 用花括号 {} 分组或环境作为状态改变的作用域。
例如 `\em abc` 改变字体以强调一些文字，得到 abc；而带参数的命令 `\emph{abc}` 可得到同样的效果。

环境 环境的格式为

```
\begin{env}
  环境的内容
\end{env}
```

例如右对齐：

```
\begin{flushright}
文字
\end{flushright}
```

文字

提纲

正文文本

直接输入正文文本。

用空格分开单词。一个换行符等同于一个空格，多个空格的效果与一个相同。

自然段分段是空一行。

正文符号

一些符号被 LaTeX 宏语言所占用，需要以命令形式输入：

```
\# \$ \% \& \{ \}
\textbackslash
```

$ % & { } \

键盘上没有的符号用命令输入。

```
\S \dag \ddag \P \copyright
\textbullet \textregistered
\texttrademark \pounds
```

§†‡¶©•®™£

更多的符号需要使用符号字体包。（看 symbols 文档）

提纲

提纲

数学模式

数学模式下字体、符号、间距与正文都不同，一切数学公式（包括单个符号 n, π）都要在数学模式下输入。

- 行内（inline）公式：使用一对符号 `$ $` 来标示。如 `$a+b=c$`。
- 显示（display）公式。
  - 简单的不编号公式用命令 `\[` 和 `\]` 标示。（不要使用双美元符号 `$$ $$`）
  - 基本的编号的公式用 `equation` 环境。
  - 更复杂的结构，使用 `amsmath` 宏包提供的专门的数学环境。（不要使用 `eqnarray` 环境）

数学结构

- 上标与下标：用 ^ 和 _ 表示。
- 上下画线与花括号：\overline, \underline, \overbrace, \underbrace
- 分式：\frac{分子}{分母}
- 根式：\sqrt[次数]{根号下}
- 矩阵：使用 amsmath 宏包提供的专门的矩阵环境 matrix, pmatrix, bmatrix 等。特别复杂的矩阵（如带线条）使用 array 环境作为表格画出。

数学符号

- 数学字母 a, b, α, Δ，数学字体 \mathbb（$\mathbb{R}$）、\mathcal（$\mathcal{P}$）等
- 普通符号：如 \infty（∞），\angle（$\angle$）
- 二元运算符：$a + b, a - b$ 及 $a \oplus b$
- 二元关系符：$a = b, a \leq b$
- 括号：$\langle a, b \rangle$，使用 \left, \right 放大
- 标点：逗号、分号（\colon）

amsmath 与 mathtools

amsmath 是基本的数学工具包，在包含数学公式的文档中几乎无处不在。mathtools 则对 amsmath 做了一些补充和增强。

amsmath 与 mathtools

amsmath 是基本的数学工具包，在包含数学公式的文档中几乎无处不在。mathtools 则对 amsmath 做了一些补充和增强。

例子：

$$
\begin{aligned}
2^5 &= (1+1)^5 \\
&= \binom{5}{0}\cdot 1^5 + \binom{5}{1}\cdot 1^4\cdot 1 + \binom{5}{2}\cdot 1^3\cdot 1^2 \\
&\quad + \binom{5}{3}\cdot 1^2\cdot 1^3 + \binom{5}{4}\cdot 1\cdot 1^4 + \binom{5}{5}\cdot 1^5 \\
&= \binom{5}{0} + \binom{5}{1} + \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5}
\end{aligned}
$$

示例代码

```
\begin{align*}
2^5 &= (1+1)^5 \\
    &= \begin{multlined}[t]
      \binom50\cdot 1^5 + \binom51\cdot 1^4 \cdot 1
        + \binom52\cdot 1^3 \cdot 1^2 \\
      + \binom53\cdot 1^2 \cdot 1^3
        + \binom54\cdot 1 \cdot 1^4 + \binom55\cdot 1^5
    \end{multlined} \\
    &= \binom50 + \binom51 + \binom52 + \binom53
        + \binom54 + \binom55
\end{align*}
```

提纲

siunitx：数字单位的一揽子解决方案

```
\num{-1.235e96} \\
\SI{299792458}{m/s} \\
\SI{2x7x3.5}{m}
```

-1.235×10^{96}
299 792 458 m/s
2 m × 7 m × 3.5 m

siunitx：数字单位的一揽子解决方案

```
\num{-1.235e96} \\
\SI{299792458}{m/s} \\
\SI{2x7x3.5}{m}
```

-1.235×10^{96}
299 792 458 m/s
2 m × 7 m × 3.5 m

```
\begin{tabular}{|S|}\hline
-234532\\ 13.55 \\ .9e37km \\
\hline
\end{tabular}
```

−234 532
13.55
0.9×10^{37} km

siunitx：数字单位的一揽子解决方案

```
\num{-1.235e96} \\
\SI{299792458}{m/s} \\
\SI{2x7x3.5}{m}
```

-1.235×10^{96}
299 792 458 m/s
2 m × 7 m × 3.5 m

```
\begin{tabular}{|S|}\hline
-234532\\ 13.55 \\ .9e37km \\
\hline
\end{tabular}
```

−234 532
13.55
0.9×10^{37} km

注：siunitx 的代码有整个 LaTeX 内核那么长。

chemformula：编写化学式

chemformula 宏包（过去用 mhchem）是在 TEX 中定义新语法规则的典范。它让化学反应式的书写比数学式还要容易，绝大部分功能只需要 \ch 一条命令：

```
\ch{2 H2 + O2 -> 2 H2O}\\
\ch{2H2O -> 2 H2 ^ + O2 ^}
```

$$2\,H_2 + O_2 \longrightarrow 2\,H_2O$$
$$_2H_2O \longrightarrow 2\,H_2\uparrow + O_2\uparrow$$

提纲

列表环境

- `enumerate` 编号
- `itemize` 不编号
- `description` 有标题

定理类环境

- \newtheorem 定义定理类环境，如
  \newtheorem{thm}{定理}[section]
- 使用定理类环境，如

```
\begin{thm}
一个定理
\end{thm}
```

定理

一个定理

诗歌与引文

- verse
- quote
- quotation

抄录代码

- \verb 命令，如

```
代码 \verb|#include <stdio.h>|
```

代码 #include <stdio.h>

- verbatim

```
\begin{verbatim}
#include <stdio.h>
int main() {
    puts("hello world.");
}
\end{verbatim}
```

```
#include <stdio.h>
int main() {
    puts("hello world.");
}
```

高级代码：语法高亮

- 使用 listings 宏包

```
\begin{lstlisting}[language=C,
  basicstyle=\ttfamily,
  stringstyle=\color{blue}]
#include <stdio.h>
int main() {
  puts("hello world.");
}
\end{lstlisting}
```

```
#include <stdio.h>
int main() {
  puts("hello world.");
}
```

- minted 宏包（调用 Pygment）

算法结构

- clrscode 宏包（算法导论）
- algorithm2e 宏包
- algorithmicx 宏包的 algpseudocode 格式

算法结构：clrscode 示例

```
% \usepackage{clrscode}
\begin{codebox}
\Procname{$\proc{Merge-Sort}(A,p,r)$}
\li \If $p<r$
\li \Then $q \gets \lfloor(p+r)/2\rfloor$
\li   $\proc{Merge-Sort}(A,p,q)$
\li   $\proc{Merge-Sort}(A,q+1,r)$
\li   $\proc{Merge}(A,p,q,r)$
    \End
\end{codebox}
```

MERGE-SORT(A, p, r)

1 if $p < r$
2 then $q \leftarrow \lfloor (p + r)/2 \rfloor$
3 MERGE-SORT(A, p, q)
4 MERGE-SORT$(A, q + 1, r)$
5 MERGE(A, p, q, r)

提纲

画表格

使用 tabular 环境。

```
\begin{tabular}{|rr|}
\hline
输入& 输出\\ \hline
$-2$ & 4 \\
0 & 0 \\
2 & 4 \\ \hline
\end{tabular}
```

输入	输出
−2	4
0	0
2	4

可以使用一些工具生成表格代码，例如
https://www.tablesgenerator.com/latex_tables

功能各异的表格宏包

- 单元格处理：multirow、makecell
- 长表格：longtable、xtab
- 定宽表格：xtabular
- 表线控制：booktabs、diagbox、arydshln
- 表列格式：array
- 综合应用：tabu

插图

使用 graphicx 宏包提供的 \includegraphics 命令。

```
\includegraphics[width=2cm]{pkulogo.pdf}
```

代码画图

优先使用外部工具画图，特别是可视化工具，例如一般的矢量图用 Inkscape、Illustrator 甚至 PowerPoint（保存为 pdf 格式），数学图形用 MATLAB、matplotlib 之类。

如果有合适的宏包，某些特定类型的图形也可以用 LaTeX 代码作图。现代 LaTeX 绘图宏包很多基于 TikZ。

浮动体

- `figure` 环境
- `table` 环境
- 其他环境可以使用 `float` 宏包得到

浮动体的标题用 `\caption` 命令得到，自动编号。

提纲

提纲

目录

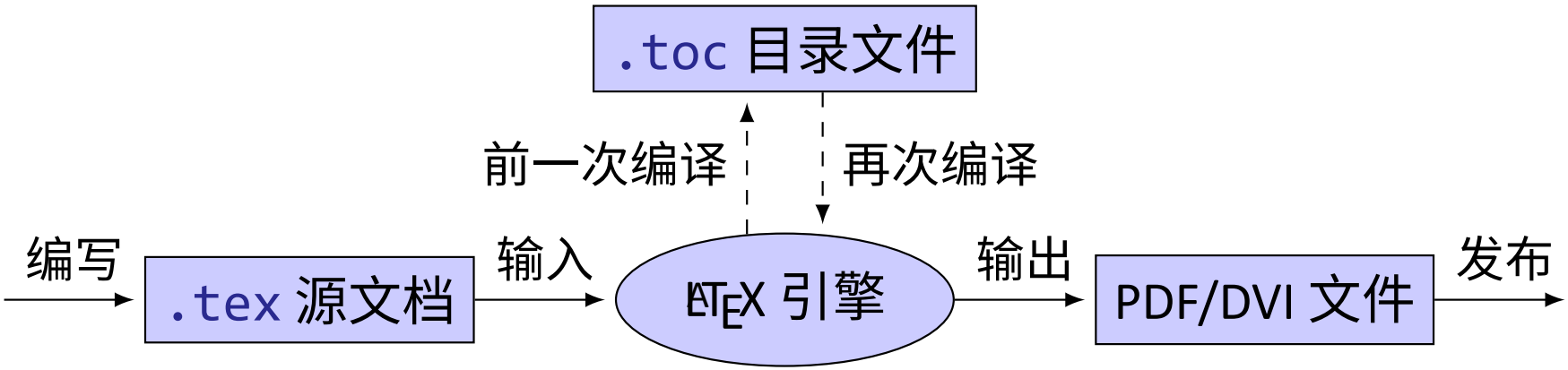


图: LaTeX 章节目录生成示意图

交叉引用工作原理

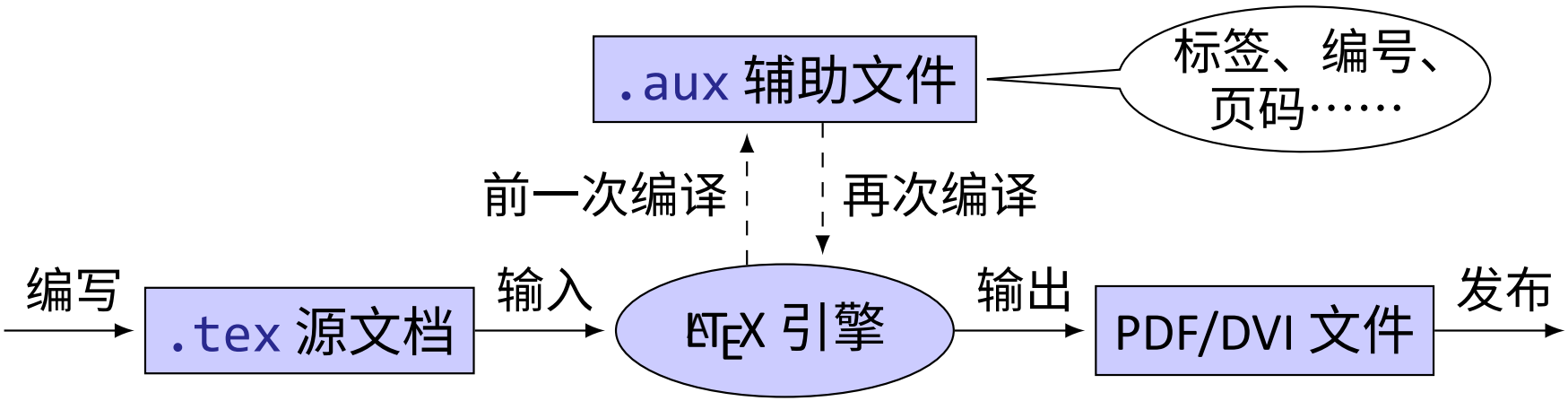


图: LaTeX 交叉引用生成示意图

hyperref：PDF 的链接与书签

hyperref 产生链接和书签的原理与普通的交叉引用相同。hyperref 会在 PDF 中写入相应的“锚点”代码，在其他地方引用。交叉引用的代码并入 .aux 文件，目录的代码并入 .toc 文件，PDF 书签则产生单独的 .out 文件。

提纲

BibTeX 工作原理

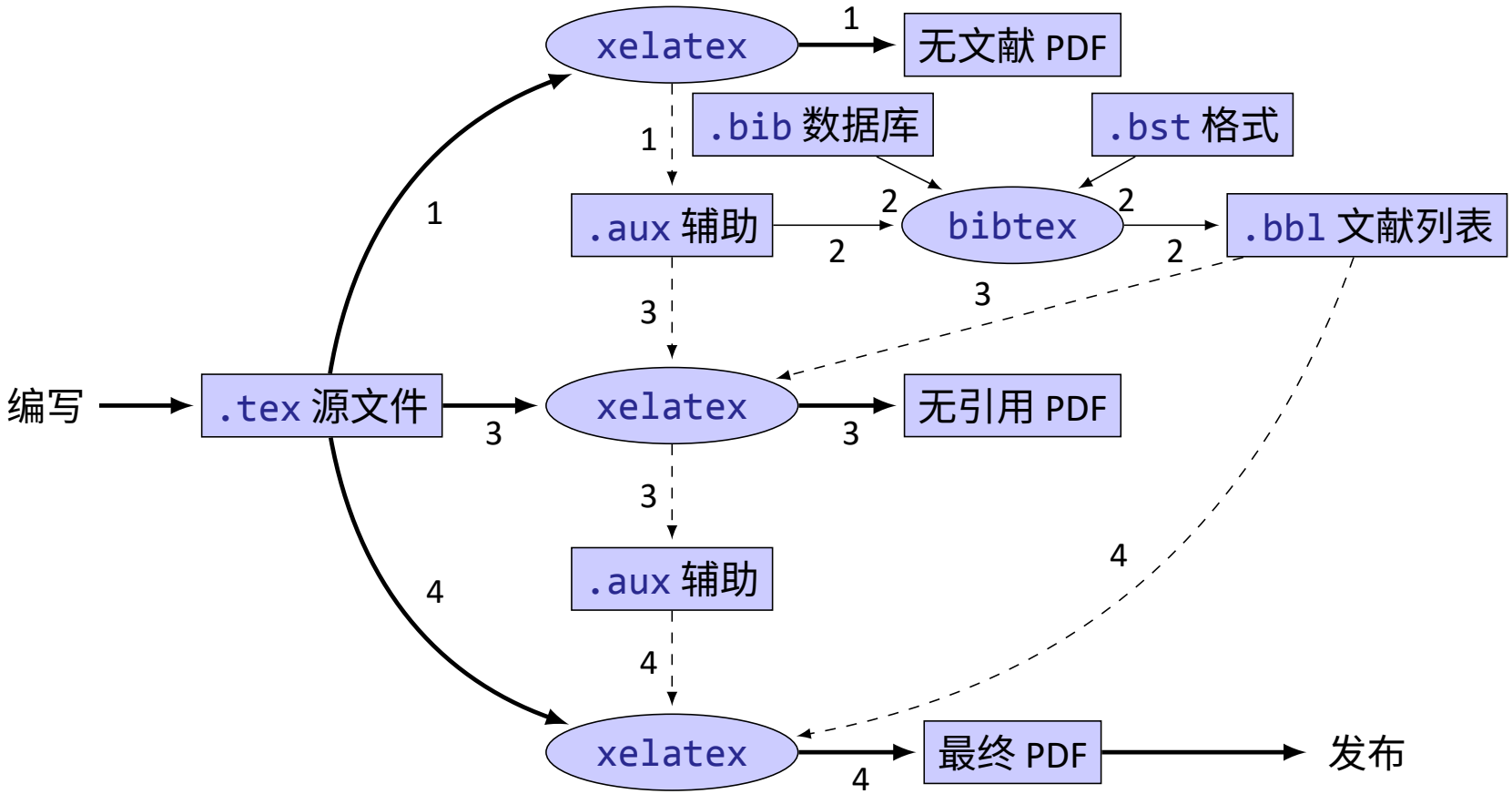


图: BibTeX 编译处理流程。这里以 XeLaTeX 为例。

设置文献格式

- 选用合适的 `.bst` 格式，比如 `plainnat`，`gbt7714-plain`。

设置文献格式

- 选用合适的 `.bst` 格式，比如 `plainnat`，`gbt7714-plain`。
- `natbib` 与作者-年格式

设置文献格式

- 选用合适的 `.bst` 格式，比如 `plainnat`，`gbt7714-plain`。
- `natbib` 与作者-年格式
- 利用 `custom-bib` 产生定制的格式文件

设置文献格式

- 选用合适的 `.bst` 格式，比如 `plainnat`，`gbt7714-plain`。
- `natbib` 与作者-年格式
- 利用 `custom-bib` 产生定制的格式文件
- `biblatex` + Biber：文献处理的新方式

第三部分

设计文档格式

提纲

~~格式与内容分离~~不要在意细节

~~格式与内容分离~~不要在意细节是 LaTeX 的一大“卖点”。它使得 LaTeX 不仅仅是 TeX 这样一种排版语言，也是一种文档编写工具。LaTeX 是面向文档作者本人的排版语言。

~~格式与内容分离~~不要在意细节

~~格式与内容分离~~不要在意细节是 LaTeX 的一大“卖点”。它使得 LaTeX 不仅仅是 TeX 这样一种排版语言，也是一种文档编写工具。LaTeX 是面向文档作者本人的排版语言。

在 LaTeX 的设计中，将文档的格式设计与内容分离开来。标准的 LaTeX 2_ε 文档类具有相对固定的排版格式，作者编写文档只使用 `\title`、`\section`、`abstract` 这样的命令或环境，而不必考虑其具体实现。而有关格式的细节代码，则被封装在文档类、宏包中，或在导言区分离编写。

~~格式与内容分离~~不要在意细节

~~格式与内容分离~~不要在意细节是 LaTeX 的一大“卖点”。它使得 LaTeX 不仅仅是 TeX 这样一种排版语言，也是一种文档编写工具。LaTeX 是面向文档作者本人的排版语言。

在 LaTeX 的设计中，将文档的格式设计与内容分离开来。标准的 LaTeX 2_ε 文档类具有相对固定的排版格式，作者编写文档只使用 `\title`、`\section`、`abstract` 这样的命令或环境，而不必考虑其具体实现。而有关格式的细节代码，则被封装在文档类、宏包中，或在导言区分离编写。

出版社提供的投稿用文档类，以及清华薛瑞尼编写的 Thuthesis 模板，北大刘玓的 pkuthss 模板，就是将事先设计好的格式交给文档作者使用的结果。

使用内容相关的命令与环境

但是，格式与内容的分离不仅需要格式设计者的努力，也需要作者在填写内容时遵循分离原则。基本的方法就是只使用与内容相关的命令和环境。

使用内容相关的命令与环境

但是，格式与内容的分离不仅需要格式设计者的努力，也需要作者在填写内容时遵循分离原则。基本的方法就是只使用与内容相关的命令和环境。

- 推荐：`It is \emph{important}.`
  不好：`It is \textit{important}.`

使用内容相关的命令与环境

但是，格式与内容的分离不仅需要格式设计者的努力，也需要作者在填写内容时遵循分离原则。基本的方法就是只使用与内容相关的命令和环境。

- 推荐：`It is \emph{important}.`
  不好：`It is \textit{important}.`
- 推荐：`\caption{流程图}`
  不好：`\textbf{图 1:} 流程图`

使用内容相关的命令与环境

但是，格式与内容的分离不仅需要格式设计者的努力，也需要作者在填写内容时遵循分离原则。基本的方法就是只使用与内容相关的命令和环境。

- 推荐：`It is \emph{important}.`
  不好：`It is \textit{important}.`
- 推荐：`\caption{流程图}`
  不好：`\textbf{图 1:} 流程图`
- 推荐：`\begin{verse} 诗行 \end{verse}`
  不好：`\begin{center} 诗行 \end{center}`
  糟糕：`~~~~~~~~~~~~~~~~~~~~~~~~~` 诗行

提纲

使用宏包

作用 宏包将可重用的代码提取出来，相当于其他程序语言中的“库”。使用宏包可以用简单的接口实现非常复杂的功能，有些对于个人来说是“不可能的任务”。

使用宏包

作用 宏包将可重用的代码提取出来，相当于其他程序语言中的“库”。使用宏包可以用简单的接口实现非常复杂的功能，有些对于个人来说是“不可能的任务”。

问题 第三方宏包可能破坏 TeX 设计的“向前兼容性”；不同宏包之间如果出现兼容性问题更难解决。——使用宏包会将兼容性问题从 TeX 语言扩大到所有宏包代码。

使用宏包

作用 宏包将可重用的代码提取出来，相当于其他程序语言中的“库”。使用宏包可以用简单的接口实现非常复杂的功能，有些对于个人来说是“不可能的任务”。

问题 第三方宏包可能破坏 TeX 设计的“向前兼容性”；不同宏包之间如果出现兼容性问题更难解决。——使用宏包会将兼容性问题从 TeX 语言扩大到所有宏包代码。

现代 LaTeX 文档离不开第三方宏包，但应合理使用：

使用宏包

作用 宏包将可重用的代码提取出来，相当于其他程序语言中的“库”。使用宏包可以用简单的接口实现非常复杂的功能，有些对于个人来说是“不可能的任务”。

问题 第三方宏包可能破坏 TeX 设计的“向前兼容性”；不同宏包之间如果出现兼容性问题更难解决。——使用宏包会将兼容性问题从 TeX 语言扩大到所有宏包代码。

现代 LaTeX 文档离不开第三方宏包，但应合理使用：

- 尽量不造轮子

使用宏包

作用 宏包将可重用的代码提取出来，相当于其他程序语言中的“库”。使用宏包可以用简单的接口实现非常复杂的功能，有些对于个人来说是“不可能的任务”。

问题 第三方宏包可能破坏 TeX 设计的“向前兼容性”；不同宏包之间如果出现兼容性问题更难解决。——使用宏包会将兼容性问题从 TeX 语言扩大到所有宏包代码。

现代 LaTeX 文档离不开第三方宏包，但应合理使用：

- 尽量不造轮子
- 尽量排除不需要的宏包

提纲

字体字号

字体

- \rmfamily, \textrm{...}
- \sffamily, \textsf{...}
- \ttfamily, \texttt{...}

字号：\Huge, \LARGE, \Large, \large, \normalsize, \small, \footnotesize, \scriptsize, \tiny

中文字号：\zihao{5}、\zihao{-3}

对齐

\centering、\raggedleft、\raggedright

空白间距

```
\hspace{2cm}
\vspace{3mm}
```

版面布局

`geometry` 宏包

`fancyhdr` 宏包等

分页断行

\linebreak、\\

\pagebreak、\newpage、\clearpage、\cleardoublepage

盒子

\mbox{内容}

\parbox{4em}{内容}、minipage

提纲

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

- 直接设置相关参数。如设置 `\parindent`、`\parskip`、`\linespread`、`\pagestyle`。

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

- 直接设置相关参数。如设置 `\parindent`、`\parskip`、`\linespread`、`\pagestyle`。
- 修改部分命令定义。如修改 `\thesection`、`\labelenumi`、`\descriptionlabel`、`\figurename`。

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

- 直接设置相关参数。如设置 `\parindent`、`\parskip`、`\linespread`、`\pagestyle`。
- 修改部分命令定义。如修改 `\thesection`、`\labelenumi`、`\descriptionlabel`、`\figurename`。
- 利用工具宏包完成设置。如使用 `ctex` 宏包设置中文格式，使用 `tocloft` 宏包设置目录格式。

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

- 直接设置相关参数。如设置 `\parindent`、`\parskip`、`\linespread`、`\pagestyle`。
- 修改部分命令定义。如修改 `\thesection`、`\labelenumi`、`\descriptionlabel`、`\figurename`。
- 利用工具宏包完成设置。如使用 `ctex` 宏包设置中文格式，使用 `tocloft` 宏包设置目录格式。

使用在导言区单独设置格式

如果预定义的格式不符合需要，就需要设置修改。经常文档作者本人就是格式设计者，此时更应该注意不要把格式和内容混在一起。

- 直接设置相关参数。如设置 `\parindent`、`\parskip`、`\linespread`、`\pagestyle`。
- 修改部分命令定义。如修改 `\thesection`、`\labelenumi`、`\descriptionlabel`、`\figurename`。
- 利用工具宏包完成设置。如使用 `ctex` 宏包设置中文格式，使用 `tocloft` 宏包设置目录格式。

传统的文档中经常修改 LaTeX 的内部命令，如重定义内部命令 `\l@section` 来修改目录格式。这体现了当初 LaTeX 设计的不足：没有提供足够的用户层接口来调整格式。不过这类方法比较晦涩，一般应该使用相关宏包功能代替。

利用自定义命令和环境

对于 LaTeX 没有直接提供的格式，可以使用自定义的命令和环境实现语义的接口。

利用自定义命令和环境

对于 LaTeX 没有直接提供的格式，可以使用自定义的命令和环境实现语义的接口。

例如，为程序名称定义一个命令：

```
\newcommand\prg[1]{\textsf{#1}}
```

利用自定义命令和环境

对于 LaTeX 没有直接提供的格式，可以使用自定义的命令和环境实现语义的接口。

例如，为程序名称定义一个命令：

```
\newcommand\prg[1]{\textsf{#1}}
```

这不仅提供了更具意义的名字，而且为未来的修改和扩充提供条件：

```
\newcommand\prg[1]{%
  \textcolor{blue}\texttt{#1}\index{#1 程序}}
```

利用自定义命令和环境

对于 LaTeX 没有直接提供的格式，可以使用自定义的命令和环境实现语义的接口。

例如，为程序名称定义一个命令：

```
\newcommand\prg[1]{\textsf{#1}}
```

这不仅提供了更具意义的名字，而且为未来的修改和扩充提供条件：

```
\newcommand\prg[1]{%
  \textcolor{blue}\texttt{#1}\index{#1 程序}}
```

注意：各种直接修改输出格式的命令，如字体、字号、对齐、间距的命令，都应该放在文档格式设置或自定义的命令、环境中，而避免在正文中直接使用。

章节标题

ctex 宏包及文档类，用 \ctexset 定制。西文用 titlesec 等。

```
\ctexset {
  chapter = {
    beforeskip = 0pt,
    fixskip = true,
    format = \Huge\bfseries,
    nameformat = \rule{\linewidth}{1bp}\par
                 \bigskip\hfill\chapternamebox,
    number = \arabic{chapter},
    aftername = \par\medskip,
    aftertitle = \par\bigskip\nointerlineskip
                 \rule{\linewidth}{2bp}\par}}
\newcommand\chapternamebox[1]{%
  \parbox{\ccwd}{\linespread{1}\selectfont\centering #1}}
```

浮动标题

caption 宏包

列表环境

enumitem 宏包