

\LaTeX 入门

刘海洋

leoliu.pku@gmail.com

北京大学

2013 年 3 月 30 日

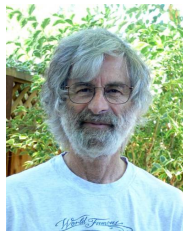


谈谈历史

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 $\text{T}_{\text{E}}\text{X}$ 排版系统。但或许因为在 $\text{T}_{\text{E}}\text{X}$ 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, Dijkstra 奖得主。为了准备他的著作《The Great American Concurrency Book》而编写了一组基于 $\text{T}_{\text{E}}\text{X}$ 的宏, 即 \LaTeX , 后交给 $\text{\LaTeX}3$ 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 $\text{T}_\text{E}\text{X}/\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 大概是什么？

- 写毕业论文，输入公式据说很方便



印象里 $\text{T}_\text{E}^\text{X}$ / $\text{L}^\text{A}^\text{T}_\text{E}^\text{X}$ 大概是什么？

- 写毕业论文，输入公式据说很方便
- 专业投稿要用，别的格式不要



印象里 $\text{T}_\text{E}\text{X}/\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 大概是什么？

- 写毕业论文，输入公式据说很方便
- 专业投稿要用，别的格式不要
- 写书的工具，有的老师用它



印象里 $\text{T}_\text{E}\text{X}/\text{L}^\text{A}\text{T}_\text{E}\text{X}$ 大概是什么？

- 写毕业论文，输入公式据说很方便
- 专业投稿要用，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 $\text{T}_\text{E}^\text{X}$ / \LaTeX 是什么？

$\text{T}_\text{E}^\text{X}$ 来自 technology 的希腊词根 $\tau\epsilon\chi$ ，读音 $[\text{t}\epsilon\chi]$

\LaTeX = Lamport $\text{T}_\text{E}^\text{X}$ ，读音 $['\text{la}:\text{t}\epsilon\chi; '\text{leit}\epsilon\chi]$ 或者随便



人们说 $\text{T}_\text{E}\text{X}$ / \LaTeX 是什么？

$\text{T}_\text{E}\text{X}$ 来自 technology 的希腊词根 $\tau\epsilon\chi$ ，读音 $[\text{t}\epsilon\chi]$

\LaTeX = Lamport $\text{T}_\text{E}\text{X}$ ，读音 $['\text{la}:\text{t}\epsilon\chi; '\text{leit}\epsilon\chi]$ 或者随便

- $\text{T}_\text{E}\text{X}$ 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。
- $\text{T}_\text{E}\text{X}$ 是一种计算机宏语言。同类型的语言有 C 语言的宏、Linux 下的 M4；功能和形式相近而实际有所不同的是 HTML、XML。
- \LaTeX 是定义在 $\text{T}_\text{E}\text{X}$ 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。



人们说 $\text{T}_\text{E}\text{X}/\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 是什么？

$\text{T}_\text{E}\text{X}$ 来自 technology 的希腊词根 $\tau\epsilon\chi$ ，读音 [tɛx]

$\text{L}_\text{A}\text{T}_\text{E}\text{X}$ = L^Amp^Ate $\text{T}_\text{E}\text{X}$ ，读音 ['lɑ:tɛx; 'leɪtɛx] 或者随便

- $\text{T}_\text{E}\text{X}$ 是一种专业排版软件。与它在各方面最类似的是方正的书版；功能相近而用法不同的有方正的飞腾创意，Adobe 的 PageMaker、InDesign 等
- $\text{T}_\text{E}\text{X}$ 是一种计算机宏语言。同类型的还有 C 语言的宏、Linux 下的 M4；功能和用途相近而实际有所不同的有 HTML、XML。
- $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 是定义在 $\text{T}_\text{E}\text{X}$ 语言上的一大组宏命令，一种排版系统。它提供了结构化排版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 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}
```



L^AT_EX 到底是什么?——从左到右的转换

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]

$$c = \sqrt{a^2 + b^2} \quad (1)$$

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。

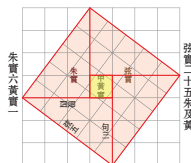


图 1: 弦圖

参考文献

[1] 古之贤人. 周髀算经. 古贤人居所, 古代.

```
\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}
```



L^AT_EX 到底是什么?——从左到右的转换

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]

$$c = \sqrt{a^2 + b^2} \quad (1)$$

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，加差實亦為弦實。



图 1: 弦圖

参考文献

[1] 古之贤人. 周髀算经. 古贤人居所, 古代.

```
\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



Part I

组织文档结构



提纲

- 1 编写结构化文档
- 2 \LaTeX ：结构化文档语言



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



提纲

- 1 编写结构化文档
- 2 L^AT_EX：结构化文档语言



L^AT_EX 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。



L^AT_EX 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	<code>\part</code>	可选的最高层
0	chapter	<code>\chapter</code>	<code>report</code> , <code>book</code> 类最高层
1	section	<code>\section</code>	<code>article</code> 类最高层
2	subsection	<code>\subsection</code>	
3	subsubsection	<code>\subsubsection</code>	<code>report</code> , <code>book</code> 类 默认不编号、不编目录
4	paragraph	<code>\paragraph</code>	默认不编号、不编目录
5	subparagraph	<code>\subparagraph</code>	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	<code>\part</code>	可选的最高层
0	chapter	<code>\chapter</code>	<code>report</code> , <code>book</code> 类最高层
1	section	<code>\section</code>	<code>article</code> 类最高层
2	subsection	<code>\subsection</code>	
3	subsubsection	<code>\subsubsection</code>	<code>report</code> , <code>book</code> 类 默认不编号、不编目录
4	paragraph	<code>\paragraph</code>	默认不编号、不编目录
5	subparagraph	<code>\subparagraph</code>	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：**省略后缀名**

- `\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}
```



Part II

填写文档内容



提纲

3 L^AT_EX 基础

4 正文文本

5 公式

6 列表与文本块

7 图表与浮动环境

8 自动化工具



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.

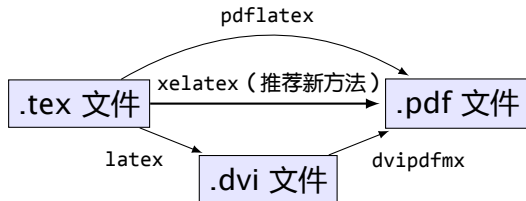
中文几乎没有改变：

```
\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 T_EX 语言，L^AT_EX 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 T_EX 语言，L^AT_EX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面，用中括号表示可选参数

`\cmd{arg1}{arg2}\`

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

L^AT_EX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

T_EX 的分数 $\frac{1}{2}$ `1 \over 2`



语法结构

相比原始的 T_EX 语言，L^AT_EX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面，用中括号表示可选参数

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

L^AT_EX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

T_EX 的分数 $\frac{1}{2}$ `1 \over 2`

- 环境

`\begin{env}`

.....

`\end{env}`

L^AT_EX 的矩阵 `\begin{matrix} ... \end{matrix}`

T_EX 的矩阵 `\matrix{...\cr ...\cr}`



语法结构

相比原始的 T_EX 语言，L^AT_EX 的语法结构被限制为相对固定的形式。

- 命令：参数总在后面，用中括号表示可选参数

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

L^AT_EX 的分数 $\frac{1}{2}$ `\frac{1}{2}`

T_EX 的分数 $\frac{1}{2}$ `1 \over 2`

- 环境

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

L^AT_EX 的矩阵 `\begin{matrix} ... \\ ... \end{matrix}`

T_EX 的矩阵 `\matrix{... \cr ... \cr}`

- 注释：以符号 % 开头，该行在 % 后面的部分。



L^AT_EX 宏：命令与环境

L^AT_EX 中的宏可分为命令与环境：

L^AT_EX 宏：命令与环境

L^AT_EX 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 L^AT_EX 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。



L^AT_EX 宏：命令与环境

L^AT_EX 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 L^AT_EX 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

3 \LaTeX 基础

4 正文文本

5 公式

6 列表与文本块

7 图表与浮动环境

8 自动化工具



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
```

```
$ \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

3 \LaTeX 基础

4 正文文本

5 公式

6 列表与文本块

7 图表与浮动环境

8 自动化工具

提纲

3 \LaTeX 基础

4 正文文本

5 公式

- 数学公式
- 科技功能

6 列表与文本块

7 图表与浮动环境



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 $\backslash[$ 和 $\backslash]$ 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]
\binom{5}{0} \cdot 1^5 + \binom{5}{1} \cdot 1^4 \cdot 1 \\
+ \binom{5}{2} \cdot 1^3 \cdot 1^2 \\
+ \binom{5}{3} \cdot 1^2 \cdot 1^3 \\
+ \binom{5}{4} \cdot 1 \cdot 1^4 + \binom{5}{5} \cdot 1^5
\end{multlined} \\
&= \binom{5}{0} + \binom{5}{1} + \binom{5}{2} + \binom{5}{3} \\
&\quad + \binom{5}{4} + \binom{5}{5}
\end{align*}

```



提纲

3 L^AT_EX 基础

4 正文文本

5 公式

- 数学公式
- 科技功能

6 列表与文本块

7 图表与浮动环境



siunitx : 数字单位的一揽子解决方案

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

$$-1.235 \times 10^{96}$$
$$299\,792\,458\,\text{m/s}$$
$$2\,\text{m} \times 7\,\text{m} \times 3.5\,\text{m}$$


siunitx : 数字单位的一揽子解决方案

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

$$-1.235 \times 10^{96}$$

$$299\,792\,458\,\text{m/s}$$

$$2\,\text{m} \times 7\,\text{m} \times 3.5\,\text{m}$$

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

$$\begin{array}{c} -234\,532 \\ 13.55 \\ 0.9 \times 10^{37}\text{km} \end{array}$$


siunitx : 数字单位的一揽子解决方案

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

$$-1.235 \times 10^{96}$$

$$299\,792\,458\,\text{m/s}$$

$$2\,\text{m} \times 7\,\text{m} \times 3.5\,\text{m}$$

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

$-234\,532$ 13.55 $0.9 \times 10^{37}\text{km}$

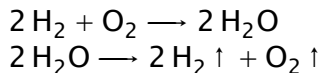
注：siunitx 的代码有整个 \LaTeX 内核那么长。



mhchem : 编写化学式

mhchem 宏包是在 T_EX 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 `\ce` 一条命令：

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



提纲

3 L^AT_EX 基础

4 正文文本

5 公式

6 列表与文本块

7 图表与浮动环境

8 自动化工具

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>
main() {
    puts("hello world.");
}
\end{verbatim}
```

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



提纲

- 3 \LaTeX 基础
- 4 正文文本
- 5 公式
- 6 列表与文本块
- 7 图表与浮动环境**
- 8 自动化工具



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

可以使用如下网站来自动生成表格代码
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`、`hhline`、`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=15cm]{foo.pdf}
```



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

- 3 \LaTeX 基础
- 4 正文文本
- 5 公式
- 6 列表与文本块
- 7 图表与浮动环境
- 8 自动化工具

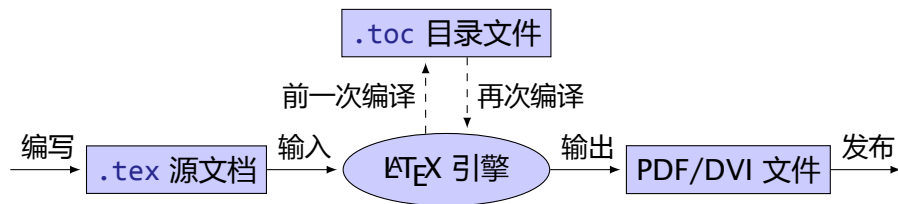


提纲

- 3 \LaTeX 基础
- 4 正文文本
- 5 公式
- 6 列表与文本块
- 7 图表与浮动环境
- 8 自动化工具
 - 目录与引用



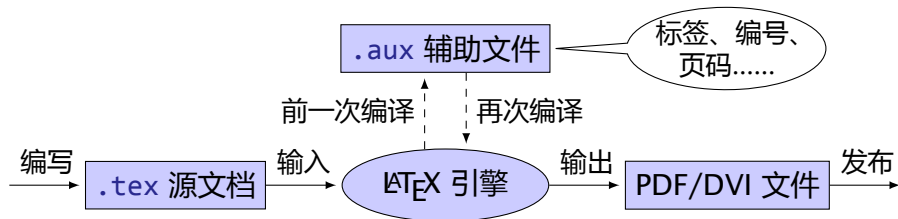
目录



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

3 \LaTeX 基础

4 正文文本

5 公式

6 列表与文本块

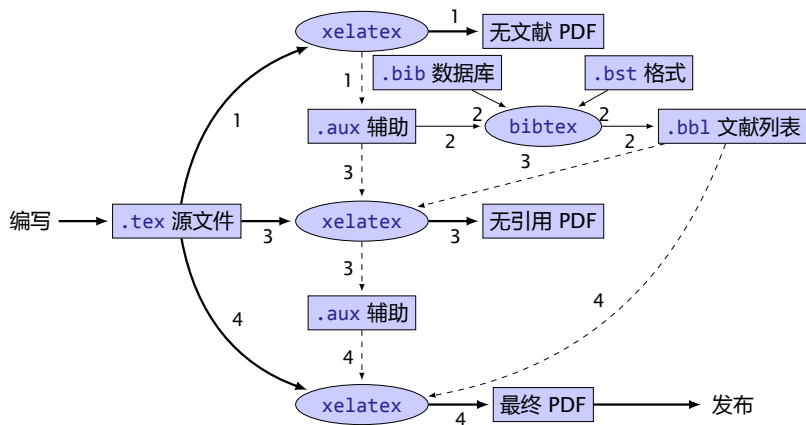
7 图表与浮动环境

8 自动化工具

- 目录与引用



BibTeX 工作原理



图：BibTeX 编译处理流程。这里以 X₃LaTeX 为例。



设置文献格式

- 选用合适的 `.bst` 格式



设置文献格式

- 选用合适的 `.bst` 格式
- `natbib` 与作者-年格式



设置文献格式

- 选用合适的 `.bst` 格式
- `natbib` 与作者-年格式
- 利用 `custom-bib` 产生格式文件



设置文献格式

- 选用合适的 `.bst` 格式
- `natbib` 与作者-年格式
- 利用 `custom-bib` 产生格式文件
- `biblatex`：文献处理的新方向

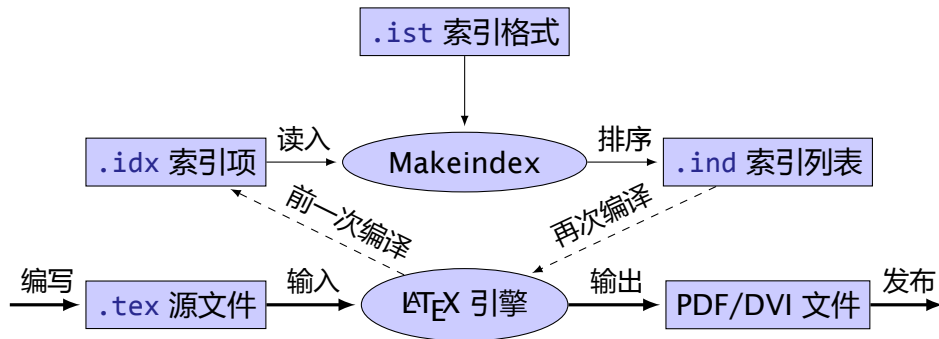


提纲

- 3 \LaTeX 基础
- 4 正文文本
- 5 公式
- 6 列表与文本块
- 7 图表与浮动环境
- 8 自动化工具
 - 目录与引用



Makeindex 工作原理



图：L^AT_EX 索引处理流程



自动处理索引

`imakeidx` 宏包可以自动处理索引项。示例：

```
\documentclass[UTF8]{ctexart}
\usepackage{imakeidx}
\makeindex[title={名词索引}]
\makeindex[name=persons,title={人名索引}]

\begin{document}
... \index{名词}
... \index[persons]{人名}
...
\printindex           % 输出名词索引
\printindex[persons] % 输出人名索引
\end{document}
```



产生词汇表

词汇表可以使用 `glossaries` 宏包产生。示例：

```
\documentclass[UTF8]{ctexart}
\usepackage{glossaries}
\makeglossaries
\begin{document}
\newglossaryentry{gloss}{
  name=glossary,
  description={A vocabulary with annotations for a particular subject},
  plural=glossaries}
\Glspl{gloss} are important for technical documents.
\newglossaryentry{sec}{
  name= 分节,
  description={把文章分成章节}}
\gls{sec} 对于长文档非常重要。
\printglossaries
\end{document}
```



提纲

- 3 \LaTeX 基础
- 4 正文文本
- 5 公式
- 6 列表与文本块
- 7 图表与浮动环境
- 8 自动化工具



hyperref 产生超链接与书签



animate 产生动画

animate 宏包可以用来产生动画效果。



animate 产生动画

animate 宏包可以用来产生动画效果。



Part III

设计文档格式



提纲

- 10 基本原则
- 11 使用宏包
- 12 格式控制功能
- 13 格式应用于文档



格式与内容分离

格式与内容分离是 \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

格式与内容分离是 \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 $\text{\LaTeX} 2_{\epsilon}$ 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`、`quote` 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

格式与内容分离是 \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 $\text{\LaTeX} 2_{\epsilon}$ 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`、`quote` 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，北大刘玗的 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：`\emph{important}`.
不好：`\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

10 基本原则

11 使用宏包

12 格式控制功能

13 格式应用于文档



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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 语言扩大到所有宏包代码。

现代 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 语言扩大到所有宏包代码。

现代 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 文档离不开第三方宏包，但应合理使用：

- 尽可能多地用宏包实现功能



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 $\text{T}_\text{E}\text{X}$ 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 $\text{T}_\text{E}\text{X}$ 语言扩大到所有宏包代码。

现代 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 文档离不开第三方宏包，但应合理使用：

- 尽可能多地用宏包实现功能
- 尽可能排除不需要的宏包



提纲

- 10 基本原则
- 11 使用宏包
- 12 格式控制功能**
- 13 格式应用于文档



字体字号

字体

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

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



对齐



空白间距



版面布局



分页断行



盒子



提纲

- 10 基本原则
- 11 使用宏包
- 12 格式控制功能
- 13 格式应用于文档**



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{\textit{#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{\textcolor{blue}{#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{\textcolor{blue}{#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

titlesec 宏包



浮动标题

caption 宏包

列表环境

enumitem 宏包

Part IV

实现全新功能



提纲

14 T_EX : 更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



提纲

14 T_EX：更多设计工具

- 间距与盒子操作
- 断行换页
- 段落形状

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理

间距与盒子操作

(缺)



提纲

14 T_EX : 更多设计工具

- 间距与盒子操作
- 断行换页
- 段落形状

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



断行分页参数控制

- L^AT_EX 层 : `\sloppy`、`sloppypar` 环境 , `\raggedright`、`\raggedleft` , `ragged2e` 宏包、`microtype`
- T_EX 层 : 罚点 `\penalty`、糟糕程序 `\tolerance` 和其他长度。



提纲

14 T_EX : 更多设计工具

- 间距与盒子操作
- 断行换页
- 段落形状

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理

首字下沉

使用 `lettrine` 宏包可以产生首字下沉的功能：

```
\lettrine{T}{he} \TeX{} in \LaTeX{} refers  
to Donald Knuth's \TeX{} typesetting system.  
The \LaTeX{} program is a special version of  
\TeX{} that understands \LaTeX{} commands.
```

The T_EX in L^AT_EX refers to Donald Knuth's T_EX typesetting system. The L^AT_EX program is a special version of T_EX that understands L^AT_EX commands.



shapepar 宏包

```
\heartpar{...}
```

绿 草 苍 苍， 白
雾 茫 茫， 有 位 佳 人， 在 水 一
方。绿草萋萋，白雾迷离，有位佳人，
靠水而居。我愿逆流而上，依偎在她身
旁。无奈前有险滩，道路又远又长。我
愿顺流而下，找寻她的方向。却见依
稀仿佛，她在水的中央。我愿逆流而
上，与她轻言细语。无奈前有险
滩，道路曲折无已。我愿顺流
而下，找寻她的足迹。却
见仿佛依稀，她在
水中伫立。



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



提纲

14 T_EX：更多设计工具

15 神秘的宏语言

- 神秘的定义
- 分支循环
- 字符类型码
- 展开控制
- 文件读写

16 阅读和编写宏包

17 字体机制

T_EX 定义的形式

- `\def`, `\gdef`
- `\edef`
- `\let`
- `\futurelet`, `\@ifnextchar`



提纲

14 T_EX：更多设计工具

15 神秘的宏语言

- 神秘的定义
- 分支循环
- 字符类型码
- 展开控制
- 文件读写

16 阅读和编写宏包

17 字体机制

T_EX 的分支

- `\ifXXX ... \else ... \fi`



T_EX 的分支

- `\ifXXX ... \else ... \fi`
- `\ifcase`



T_EX 的分支

- `\ifXXX ... \else ... \fi`
- `\ifcase`



T_EX 的分支

- `\ifXXX ... \else ... \fi`
- `\ifcase`

```
% 将参数 #2 输出 2^#1 次
\def\recur#1#2{%
  \ifnum #1=0\relax%
    #2 %
  \else
    \setcounter{enumi}{#1}%
    \addtocounter{enumi}{-1}%
    \recur{\value{enumi}}{#2 #2}%
  \fi}
\recur0a\recur1b\recur2c\recur3d
```

a b b c c c c d d d d d d d d



ifthen 与 etoolbox

- `\ifthenelse`



ifthen 与 etoolbox

- \ifthenelse
- etoolbox



循环语句

- `\loop...\repeat`



循环语句

- `\loop...\repeat`
- `\@for`, `\@tfor`



循环语句

- `\loop...\repeat`
- `\@for`, `\@tfor`
- `\@whilenum`, `\@whilesw`



提纲

14 T_EX：更多设计工具

15 神秘的宏语言

- 神秘的定义
- 分支循环
- 字符类型码
- 展开控制
- 文件读写

16 阅读和编写宏包

17 字体机制

概说

类别	意义	类别	意义
0	转义符 \	8	下标 -
1	组开始 {	9	可忽略的字符 NUL
2	组结束 }	10	空格 □
3	数学环境 \$	11	字母 A,...,Z 和 a,...,z
4	表格对齐 &	12	其它字符 其它字符
5	换行 换行符	13	活动符 ~
6	参数 #	14	注释符 %
7	上标 ^	15	无用符 DEL



`\makeatletter` 与内部命令

- `\makeatletter` 与 `\makeatother`
- 内部命令的规则



DNA 示例

```
\begingroup
\catcode`A\active \catcode`T\active
\catcode`C\active \catcode`G\active
\gdef A{\textcolor{red}{\string A}}
\gdef T{\textcolor{green}{\string T}}
\gdef C{\textcolor{blue}{\string C}}
\gdef G{\textcolor{black}{\string G}}
\endgroup
\newcommand\DNA[1]{\begingroup
  \scantokens{\catcode`A\active \catcode`C\active
    \catcode`G\active \catcode`T\active
    \ttfamily#1\endinput}%
  \endgroup}
\DNA{TCGTAGTCTGCT}
```

TCGTAGTCTGCT



verbatim 的实现

开始的核心代码：

```
\let\do\@makeother \dospecials  
\obeylines \verbatim@font \@noligs  
\hyphenchar\font\m@ne  
\everypar \expandafter{\the\everypar \unpenalty}%
```



verbatim 的实现

开始的核心代码：

```
\let\do\@makeother \dospecials
\obeylines \verbatim@font \@noligs
\hyphenchar\font\m@ne
\everypar \expandafter{\the\everypar \unpenalty}%
```

结束的核心代码：

```
{\catcode`\ =\active%
\gdef\@vobeyspaces{\catcode`\ \active\let \@xobeysp}}
\begingroup \catcode `|=0 \catcode `[= 1
\catcode `]=2 \catcode `\<=12 \catcode `\\=12
\catcode`\ =12
\gdef|@xverbatim#1\end{verbatim}[#1|end[verbatim]]
\gdef|@sxverbatim#1\end{verbatim*}[#1|end[verbatim*]]
\endgroup
```



提纲

14 TeX：更多设计工具

15 神秘的宏语言

- 神秘的定义
- 分支循环
- 字符类型码
- 展开控制
- 文件读写

16 阅读和编写宏包

17 字体机制

展开控制命令

- `\expandafter`, `\expandnext` (`etexttools`)
- `\noexpand`, `\edef`
- `\aftergroup`
- `\the`



提纲

14 TeX：更多设计工具

15 神秘的宏语言

- 神秘的定义
- 分支循环
- 字符类型码
- 展开控制
- 文件读写

16 阅读和编写宏包

17 字体机制

T_EX 层读写

```
\documentclass{article}
\newwrite\commentfile
\openout\commentfile=\jobname.comment
\begin{document}
\section{Test}
text one
\write\commentfile{\noexpand\item
  comment one at page \thepage}
text two
\write\commentfile{\noexpand\item
  comment two in section \thesection}
\section{Comments}
\closeout\commentfile
\begin{enumerate}
  \input{\jobname.comment}
\end{enumerate}
\end{document}
```



L^AT_EX 辅助文件

```
\documentclass{article}
\makeatletter
\newcommand\addcommentitem[1]{%
  \write\@auxout{\noexpand\@writefile{comment}{\noexpand\item #1}}}
\newcommand\printcomment{%
  \section*{Comments}
  \begin{enumerate}
    \item[] Here are some comments:
    \@starttoc{comment}
  \end{enumerate}}
\makeatother
\begin{document}
\section{Test}
text one
\addcommentitem{comment one at page \thepage}
text two
\addcommentitem{comment two in section \thesection}
\printcomment
\end{document}
```



使用 newfile 宏包

```
\documentclass{article}
\usepackage{newfile}
\newoutputstream{comment}
\openoutputfile{\jobname.comment}{comment}
\begin{document}
\section{Test}
text one
\addtostream{comment}{\noexpand\item
  comment one at page \thepage}
text two
\addtostream{comment}{\noexpand\item comment two}
\section{Comments}
\closeoutputstream{comment}
\begin{enumerate}
  \input{\jobname.comment}
\end{enumerate}
\end{document}
```



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



基本读物

- $\text{T}_{\text{E}}\text{X}$ book, $\text{T}_{\text{E}}\text{X}$ by Topic: 宏语言书籍



基本读物

- T_EXbook, T_EX by Topic: 宏语言书籍
- `source2e`, `classes` 是 L^AT_EX 2_ε 的源代码说明



基本读物

- T_EXbook, T_EX by Topic: 宏语言书籍
- `source2e`, `classes` 是 L^AT_EX 2_ε 的源代码说明
- `clsguide` 宏包作者必读的手册



提纲

14 $\text{T}_\text{E}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

- 阅读与修改
- 编写与维护

17 字体机制

18 中文处理



读懂宏包代码

- 查看 docstrip 带文档的源代码



读懂宏包代码

- 查看 docstrip 带文档的源代码
- 快速浏览与搜索



读懂宏包代码

- 查看 docstrip 带文档的源代码
- 快速浏览与搜索
- 进行实验



读懂宏包代码

- 查看 docstrip 带文档的源代码
- 快速浏览与搜索
- 进行实验



读懂宏包代码

- 查看 docstrip 带文档的源代码
- 快速浏览与搜索
- 进行实验

实例：

- `indentfirst`



读懂宏包代码

- 查看 docstrip 带文档的源代码
- 快速浏览与搜索
- 进行实验

实例：

- `indentfirst`
- `clrscode`



提纲

14 $\text{T}_\text{E}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

- 阅读与修改
- 编写与维护

17 字体机制

18 中文处理



编写与维护宏包

- 编写开发代码和文档

编写与维护宏包

- 编写开发代码和文档
- 使用 docstrip : `sty2dtx`



编写与维护宏包

- 编写开发代码和文档
- 使用 docstrip : `sty2dtx`
- 版本控制与代码托管



编写与维护宏包

- 编写开发代码和文档
- 使用 docstrip : `sty2dtx`
- 版本控制与代码托管
- 上传 CTAN 发布



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

- NFSS
- $\text{T}_{\text{E}}\text{X}$ 字体
- 驱动与引擎

18 中文处理



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- \fontencoding{enc}



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- `\fontencoding{enc}`
- `\fontfamily{fam}`



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- \fontencoding{enc}
- \fontfamily{fam}
- \fontseries{series}



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- \fontencoding{enc}
- \fontfamily{fam}
- \fontseries{series}
- \fontshape{shape}



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- \fontencoding{enc}
- \fontfamily{fam}
- \fontseries{series}
- \fontshape{shape}
- \fontsize{size}{lineskip}



使用 NFSS

字体族	\rmfamily		\sffamily		\ttfamily	
系列 形状	\mdseries	\bfseries	\mdseries	\bfseries	\mdseries	\bfseries
\upshape	Text	Text	Text	Text	Text	Text
\itshape	<i>Text</i>	<i>Text</i>	同 <i>slanted</i>	同 <i>slanted</i>	<i>Text</i>	同 <i>slanted</i>
\slshape	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>	<i>Text</i>
\scshape	Text	缺	缺	缺	Text	缺

表：字体的坐标：族、形状和系列 (Latin Modern)

直接选择坐标：

- \fontencoding{enc}
- \fontfamily{fam}
- \fontseries{series}
- \fontshape{shape}
- \fontsize{size}{lineskip}
- \usefont{enc}{fam}{series}{shape}



X_YTeX 下的 NFSS

- 西文：fontspec 机制。

```
\setmainfont{Font}
```

```
\setsansfont{Font}
```

```
\setmonofont{Font}
```

```
\newfontfamily\cmd{Font}
```



X_YTeX 下的 NFSS

- 西文：fontspec 机制。

```
\setmainfont{Font}  
\setsansfont{Font}  
\setmonofont{Font}  
\newfontfamily\cmd{Font}
```

- 中文：xeCJK 机制。

```
\setCJKmainfont{Font}  
\setCJKsansfont{Font}  
\setCJKmonofont{Font}  
\setCJKfamilyfont{fam}{Font}
```



定制 NFSS

示例：使用某数学符号。

```
\documentclass{article}
\DeclareFontFamily{U}{matha}{\hyphenchar\font45}
\DeclareFontShape{U}{matha}{m}{n}{
  <5> <6> <7> <8> <9> <10> gen * matha
  <10.95> matha10 <12> <14.4> <17.28> <20.74> <24.88> m
}{}
\DeclareSymbolFont{matha}{U}{matha}{m}{n}
\DeclareFontSubstitution{U}{matha}{m}{n}
\DeclareMathSymbol{\abxcup}{\mathbin}{matha}{'131}
\begin{document}
$A\cup B \abxcup C$
\end{document}
```



提纲

14 T_EX：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

- NFSS
- T_EX 字体
- 驱动与引擎

18 中文处理

T_EX 字体机制

- `\font` 命令



T_EX 字体机制

- `\font` 命令
- `.tfm` 文件



提纲

14 T_EX：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

- NFSS
- T_EX 字体
- 驱动与引擎

18 中文处理



不同驱动使用的字体

- Dvips, dvipdfm: METAFONT, Type 1



不同驱动使用的字体

- Dvips, dvipdfm: METAFONT, Type 1
- dvipdfmx, pdfT_EX: METAFONT, Type 1, TrueType



不同驱动使用的字体

- Dvips, dvipdfm: METAFONT, Type 1
- dvipdfmx, pdf \TeX : METAFONT, Type 1, TrueType
- X \TeX : METAFONT, Type 1, TrueType, OpenType



字体映射文件

- `psfonts.map`



字体映射文件

- psfonts.map
- dvipdfm.map



字体映射文件

- psfonts.map
- dvipdfm.map
- pdftex.map



字体映射文件

- psfonts.map
- dvipdfm.map
- pdftex.map
- cid-x.map



字体映射文件

- psfonts.map
- dvipdfm.map
- pdftex.map
- cid-x.map
- updmap 工具



字体映射文件

- psfonts.map
- dvipdfm.map
- pdftex.map
- cid-x.map
- updmap 工具
- zhwinfonts.tex



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理



困难

- 用 $\text{T}_\text{E}\text{X}$ 处理中文最大的困难是： $\text{T}_\text{E}\text{X}$ 是基于 ASCII 文字编码编写的程序。最早的 $\text{T}_\text{E}\text{X}$ 只支持 7 位的标准 ASCII 码，因此输出带重音符号的拉丁字母也需要使用宏命令。后来新的实现转用扩展 ASCII 码，可以每次以一个字节（8 位）来处理，因而能比较方便地使用拉丁文字，`inputenc` 宏包、`T1` 编码等就是这个时代的产物。但 CJK 文字是多字节编码，原有的 $\text{T}_\text{E}\text{X}$ 无法直接处理整个的汉字。 $\text{T}_\text{E}\text{X}$ 使用的每个字体也最多只能包含 256 个符号，汉字的显示就成问题。



困难

- 用 $\text{T}_\text{E}\text{X}$ 处理中文最大的困难是： $\text{T}_\text{E}\text{X}$ 是基于 ASCII 文字编码编写的程序。最早的 $\text{T}_\text{E}\text{X}$ 只支持 7 位的标准 ASCII 码，因此输出带重音符号的拉丁字母也需要使用宏命令。后来新的实现转用扩展 ASCII 码，可以每次以一个字节（8 位）来处理，因而能比较方便地使用拉丁文字，`inputenc` 宏包、`T1` 编码等就是这个时代的产物。但 CJK 文字是多字节编码，原有的 $\text{T}_\text{E}\text{X}$ 无法直接处理整个的汉字。 $\text{T}_\text{E}\text{X}$ 使用的每个字体也最多只能包含 256 个符号，汉字的显示就成了问题。
- 除了字符集和字体的问题，汉字的断行（ $\text{T}_\text{E}\text{X}$ 依赖空格和分词断行）、标点规则也都是 $\text{T}_\text{E}\text{X}$ 在设计时所没有考虑的。



困难

- 用 $\text{T}_\text{E}\text{X}$ 处理中文最大的困难是： $\text{T}_\text{E}\text{X}$ 是基于 ASCII 文字编码编写的程序。最早的 $\text{T}_\text{E}\text{X}$ 只支持 7 位的标准 ASCII 码，因此输出带重音符号的拉丁字母也需要使用宏命令。后来新的实现转用扩展 ASCII 码，可以每次以一个字节（8 位）来处理，因而能比较方便地使用拉丁文字，`inputenc` 宏包、`T1` 编码等就是这个时代的产物。但 CJK 文字是多字节编码，原有的 $\text{T}_\text{E}\text{X}$ 无法直接处理整个的汉字。 $\text{T}_\text{E}\text{X}$ 使用的每个字体也最多只能包含 256 个符号，汉字的显示就成问题。
- 除了字符集和字体的问题，汉字的断行（ $\text{T}_\text{E}\text{X}$ 依赖空格和分词断行）、标点规则也都是 $\text{T}_\text{E}\text{X}$ 在设计时所没有考虑的。
- 中文版式、标题汉化等。这部分没有两两项困难，不过更影响用户体验的。这也是为什么学校有很多老师还在用以前的 `CCT` 系统，`ctex` 宏包的主要工作也在于此。



提纲

14 $\text{T}_\text{E}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理

- 字体选择

- 标点

- 翻译与格式设定



预处理方式：CCT

(缺内容)



逐字节处理：CJK

CJK 将汉字的前一个字节的字符类别码改变为 13，使它定义为一个命令，然后将两个字节的汉字转换为带两个数字参数的命令：

汉字：

```
\CJKchar[GBK]{186}{186}\CJKchar[GBK]{215}{214}
```



逐字节处理：CJK

CJK 将汉字的前一个字节的字符类别码改变为 13，使它定义为一个命令，然后将两个字节的汉字转换为带两个数字参数的命令：

汉字：

```
\CJKchar[GBK]{186}{186}\CJKchar[GBK]{215}{214}
```

一套汉字字体被分成数十套字体，根据数字参数选择字体和字体中的符号：

```
{\font\x="gbksong43"\x\char199}%  
{\font\x="gbksong65"\x\char105}
```



原生 Unicode 支持：X_YT_EX

在 X_YT_EX 中可以直接把汉字作为单个 Unicode 字符处理，同时字体也可以直接使用操作系统的汉字字体。因此汉字的支持是原生的：

```
\font\song="宋体" at 24pt  
\song 宋体字
```

宋体字



原生 Unicode 支持：X_YT_EX

在 X_YT_EX 中可以直接把汉字作为单个 Unicode 字符处理，同时字体也可以直接使用操作系统的汉字字体。因此汉字的支持是原生的：

```
\font\song="宋体" at 24pt  
\song 宋体字
```

宋体字

但要支持汉字字体和西文字体的自动切换，就需要特殊的技术。X_YT_EX 提供了一种字符给分类的机制，每类字符作为一个状态，从一个字符到另一个字符发生状态转移时，可以在两个字符之间插入任意的 T_EX 代码。利用这种基于有限自动机的机制，可以让 L^AT_EX 在中文和西文字体之间自动切换。



提纲

14 $\text{T}_{\text{E}}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理

- 字体选择
- 标点
- 翻译与格式设定



标点处理要解决的问题

- 标点禁则



标点处理要解决的问题

- 标点禁则
- 标点压缩



不同中文处理方式下的标点处理

- CJK：按字符判断禁则



不同中文处理方式下的标点处理

- CJK：按字符判断禁则
- CJKpunct：预读取下一字符处理压缩



不同中文处理方式下的标点处理

- CJK：按字符判断禁则
- CJKpunct：预读取下一字符处理压缩
- xeCJK：按字符类处理禁则和压缩



提纲

14 $\text{T}_\text{E}\text{X}$ ：更多设计工具

15 神秘的宏语言

16 阅读和编写宏包

17 字体机制

18 中文处理

- 字体选择

- 标点

- 翻译与格式设定



翻译与格式设定

- 汉化中文部件



翻译与格式设定

- 汉化中文部件
- 字号大小



翻译与格式设定

- 汉化中文部件
- 字号大小
- 缩进处理



翻译与格式设定

- 汉化中文部件
- 字号大小
- 缩进处理
- 汉字编号

