

# 使用 L<sup>A</sup>T<sub>E</sub>X 撰写逻辑学论文

王轶

ynw@xixilogic.org

西溪逻辑论坛第 38 期

浙大西溪校区教学主楼 259 会议室 2016-09-30

( 〇 )

引子



# 如何进行文符混排论文的实际写作？

- Word + 特殊字体 + 公式编辑器/MathType 等?
  - ① 不同字体间协调性不够
  - ② 公式编辑器等生成的 object 多了以后，文档打开很慢，程序易死
  - ③ 录入不便



# 如何进行文符混排论文的实际写作？

- Word + 特殊字体 + 公式编辑器/MathType 等?
  - ① 不同字体间协调性不够
  - ② 公式编辑器等生成的 object 多了以后，文档打开很慢，程序易死
  - ③ 录入不便
- 如何改进？
  - 换用更稳定的类 Word 排版软件？（特别是支持矢量图的）
    - Openoffice/Libreoffice + TexMaths/Jex/OOolatex 插件，以矢量图方式嵌入
    - 依然无法解决字体协调性和录入不便的问题



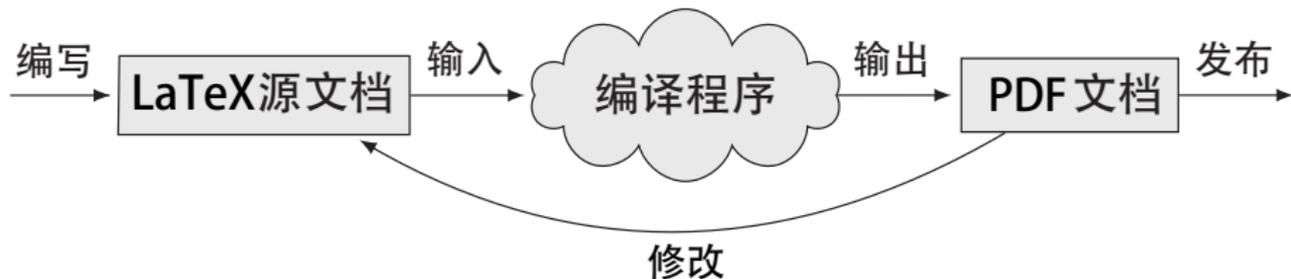
# 如何进行文符混排论文的实际写作？

- Word + 特殊字体 + 公式编辑器/MathType 等?
  - ① 不同字体间协调性不够
  - ② 公式编辑器等生成的 object 多了以后，文档打开很慢，程序易死
  - ③ 录入不便
- 如何改进？
  - 换用更稳定的类 Word 排版软件？（特别是支持矢量图的）
    - Openoffice/Libreoffice + TexMaths/Jex/OOolatex 插件，以矢量图方式嵌入
    - 依然无法解决字体协调性和录入不便的问题
  - 设计专用字体，并设计“输入法”支持该字体？
    - 符号间距、拼写检查等问题受制于排版软件



$\text{T}_{\text{E}}\text{X}$  (来自  $\tau\acute{\epsilon}\chi\nu\eta$ , 读音 /'tɛx/ “泰赫”), Donald E. Knuth 1976

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  (读音 /'la:tɛx/ 或 /'leɪ:tɛx/), Leslie Lamport 1985 对  $\text{T}_{\text{E}}\text{X}$  扩充以更丰富的文档处理功能



## 当下使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 写作的基本流程



# TEX 及其常用扩充

命令 / 引擎 \ 格式	plain T <sub>E</sub> X	L <sup>A</sup> T <sub>E</sub> X	ConT <sub>E</sub> Xt	输出
T <sub>E</sub> X / ε-T <sub>E</sub> X	tex/etex			} DVI
pdfT <sub>E</sub> X	tex pdftex	latex pdflatex	texexec	
X <sub>Ǝ</sub> T <sub>E</sub> X	xetex	xelatex	texexec	} DVI PDF
LuaT <sub>E</sub> X	dviluatex luatex	dvilualatex lualatex	context	

当下主流:

- 英文: pdflatex
- 中文: xelatex (pdflatex + CJK 宏包方式已逐渐被淘汰)

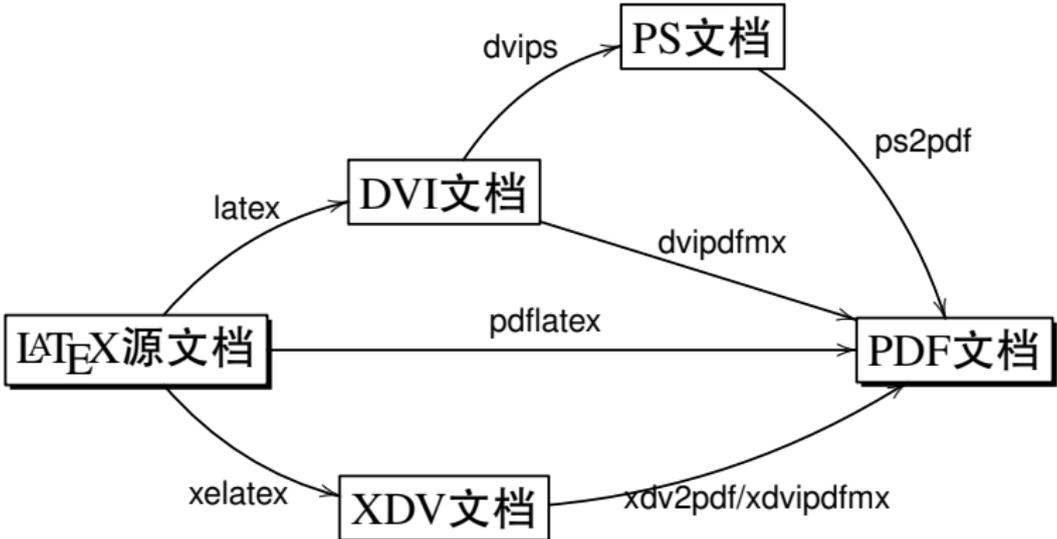


## \* DVI、PS 和 PDF

- 三者都是页面描述语言 (Page Description Language, PDL)
- **DVI**:  $\text{T}_{\text{E}}\text{X}$  最早生成设备相关的文档格式, 1979 年转用设备无关 (DeVice Independent, DVI) 的格式
  - DVI 不支持嵌入字体和图形, 只算中间格式, 最终打印还需转化为应用支持的格式
- **PostScript (PS)**: 1984 年 Adobe 推出, 早期激光打印机支持的标准语言
  - PS 语言过于复杂, PS 驱动芯片成本高
  - 1990 年代中期廉价喷墨打印机出现后逐渐式微
- **PDF**: 1993 年 Adobe 推出, 2007 年开放大部分的格式标准
  - 语言部分是 PS 的子集
  - 其它功能大大增强, 可嵌入先进字体、透明图、动图等



# L<sup>A</sup>T<sub>E</sub>X 写作流程细化





# 安装 T<sub>E</sub>X 及其扩展系统

- 除用于编译的核心引擎和工具外，还有大量宏包、第三方工具和文档等
- CTAN (Comprehensive T<sub>E</sub>X Archive Network)  
<http://www.ctan.org>  
统一管理 T<sub>E</sub>X 相关的各类资料
- 安装 T<sub>E</sub>X 发行版：
  - Windows 平台：TeX Live 或 MiKTeX
  - Linux 平台：TeX Live
  - Mac 平台：TeX Live (= TeX Live + Ghostscript + 几个 Mac 软件)

# L<sup>A</sup>T<sub>E</sub>X 文档编辑器



参考: [https://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](https://en.wikipedia.org/wiki/Comparison_of_TeX_editors)

- L<sup>A</sup>T<sub>E</sub>X 代码本质上是纯文本, 任何文本软件都可作为编辑器
  - 以 Windows 中的“写字板”为例:  
写代码 => 保存为 `ex.tex` 文档 => 命令行运行 `pdflatex ex.tex`

# L<sup>A</sup>T<sub>E</sub>X 文档编辑器



参考: [https://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](https://en.wikipedia.org/wiki/Comparison_of_TeX_editors)

- L<sup>A</sup>T<sub>E</sub>X 代码本质上是纯文本, 任何文本软件都可作为编辑器
  - 以 Windows 中的“写字板”为例:  
写代码 => 保存为 `ex.tex` 文档 => 命令行运行 `pdflatex ex.tex`
- 很多文本编辑器提供 L<sup>A</sup>T<sub>E</sub>X 语法支持和/或编译支持:
  - 例: Emacs、VIM、Eclipse、SciTE、Notepad++ 等

注意: 跟 Unix 类系统一样, TeX 中严格区分大小写

# L<sup>A</sup>T<sub>E</sub>X 文档编辑器



参考: [https://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](https://en.wikipedia.org/wiki/Comparison_of_TeX_editors)

- L<sup>A</sup>T<sub>E</sub>X 专用编辑器举例
  - 传统 ( 代码界面 ): Kile (Linux)、WinEdt (Win)、TeXshop (Mac)
  - 分屏: Texstudio、TeXworks、TeXpen、Texmaker、Texiteasy
  - 所见即所得: Bakoma TeX
  - 所见即所指: LyX、Scientific Workplace (Win)、Cassiopeia (Mac)
  - 在线编辑器: ShareLaTeX、Overleaf、Authorea、Papeeria
  - 手机应用 ( 在线编译 ): Verbosus、Texpad

# L<sup>A</sup>T<sub>E</sub>X 文档编辑器



参考: [https://en.wikipedia.org/wiki/Comparison\\_of\\_TeX\\_editors](https://en.wikipedia.org/wiki/Comparison_of_TeX_editors)

- L<sup>A</sup>T<sub>E</sub>X 专用编辑器举例
  - 传统 ( 代码界面 ): Kile (Linux)、WinEdt (Win)、TeXshop (Mac)
  - 分屏: Texstudio、TeXworks、TeXpen、Texmaker、Texiteasy
  - 所见即所得: Bakoma TeX
  - 所见即所指: LyX、Scientific Workplace (Win)、Cassiopeia (Mac)
  - 在线编辑器: ShareLaTeX、Overleaf、Authorea、Papeeria
  - 手机应用 ( 在线编译 ): Verbosus、Texpad
- 选择的重要制约因素: 1. 工作平台, 2. 是否需要中文支持
- 初学者推荐使用: TeXworks



# L<sup>A</sup>T<sub>E</sub>X 对比 Word 的优势

- 内容与样式分离
  - 统一样式设计 + 局部可调整，整齐划一，不会出错
  - 所有编号自动生成，无需手动调整
- 符号输入快
- 环境和布局功能突出
- 便捷矢量图形绘制和嵌入
- 索引、目录、参考文献等功能强于 Word
- 纯文本，易复制，易统一更改，内容可注释和恢复
- 稳定性、(版本和跨平台)兼容性、可扩展性

总之：L<sup>A</sup>T<sub>E</sub>X 入门难，学习曲线平缓，更易制作高品质的文档



# 讲座思路

按照应用场景分为如下几项

- ① L<sup>A</sup>T<sub>E</sub>X 文档基本架构
- ② 熟悉 L<sup>A</sup>T<sub>E</sub>X 符号体系
- ③ 撰写英文期刊/会议论文
- ④ 撰写中文期刊/会议论文
- ⑤ 撰写中文学位论文、书籍
- ⑥ 制作演示文档

(一)

# L<sup>A</sup>T<sub>E</sub>X 文档基本架构



# 基本代码结构

使用 ShareLaTeX 运行

The screenshot shows the ShareLaTeX online editor interface. The left pane contains the following LaTeX source code:

```

1 \documentclass{article}
2 \begin{document}
3 Let me give it a try!
4 \end{document}

```

The right pane shows the rendered output of the code, which is the text "Let me give it a try!". Above the rendered output is a "Recompile" button and some utility icons. The browser address bar shows the URL <https://www.sharelatex.com>.



# 稍微复杂一点的文档结构

```

\documentclass{article}
\begin{document}
\title{This is a Sample Paper}
\author{Im Nobody}
\maketitle
\begin{abstract}
This is really just an example.
\end{abstract}
\section{Introduction}Bla bla.
\section{Exciting!}
\subsection{Too young}
\subsection{Too simple}
\begin{itemize}
\item simple 1
\item simple 2 \cite{mrx05}
\end{itemize}
\bibliographystyle{plain}
\bibliography{test}
\end{document}

```

This is a Sample Paper

Im Nobody  
August 28, 2016

**Abstract**  
This is really just an example.

**1 Introduction**  
Bla bla.

**2 Exciting!**

2.1 Too young

2.2 Too simple

- simple 1
- simple 2 [1]

**References**

[1] Mr. X. Something great. *JJJ*, 2005.



# 文档框架结构

- L<sup>A</sup>T<sub>E</sub>X 文档的区域：
  - 导言区：主要负责文档设计
  - 内容区：主要负责文档内容。以 book 类型的文档为例：
    - `\frontmatter`，涵盖正文前的前言、目录等
    - `\mainmatter`，涵盖正文章节
    - `\backmatter`，涵盖正文后的参考文献、索引、附录、后记等
    - `\appendix`，进入附录章节



# 导言区的主要工作

- 样式设置 ( 字体、大小、间距、编号等, 需宏包加强功能 )
  - 文档类、页面尺寸、边距、奇偶页 (geometry), 页眉、页脚 (fancyhdr)
  - 全文默认字体 ( 中、英、符可分别设置 )、字号、文字编码、行距
  - 文档拆分、标题、章节、段落、缩进、断字、段尾孤字、断页
  - 定义、定理、证明等 (amsthm, ntheorem, thmtools)
  - 列表 (enumitem)、矩阵、图片 (graphicx, subcaption, wrapfig)、表格 (diagbox, multirow, multicol, colortbl)、引用/诗歌/摘要等
  - ( 章节、图片、表格 ) 目录 (tocloft, tocbibind)、脚注 (footmisc)、边柱、参考文献 (natbib)、索引 (splitidx)、超链接 (hyperref) 等
- 功能增强
  - 中文 (xecjk, ctex), 画图 (tikz, xy), 色彩 (xcolor), 额外符号 (amssymb, latexsym, mathrsfs, pifont 等), 各类划线 (ulem), 导入 PDF (pdfpages), 代码环境 (verbatim, fancyvbr, cprotect, listings), 矢列证明 (bussproofs), 语义图 (qtree)
- 自定义命令

(二)

# 熟悉 L<sup>A</sup>T<sub>E</sub>X 符号体系



# 本节内容可使用如下文档架构

```
\documentclass{article}
```

```
\usepackage{amssymb,amsmath}
```

%使用美国数学学会制作的符号和数学宏包

```
\begin{document}
```

```
\end{document}
```



# 两种模式

## 文本模式

- 直接输入
- 放在 `\text{...}` (amsmath 包) 或 `\textrm{...}` 中
- 放在 `\mbox{...}` 等盒子中
- 常规: `a M / @ - + = | , ; . [ ] ?`
- 需转义: `\ $ % ^ _ { } ~ & #`
- 自动转义: `-- --- ` `` ' "`
- 其他转义: `æ œ å á § § Û Ì ā`
- 空格只计一次

## 数学模式

- 行内公式: 放在 `$...$` 或 `\(...\)` 中
- 行间公式: 放在 `$$...$$` 或 `\[...]` 中
- 放在 `equation`、`array` 等数学环境中
- 常规: `aM/@ - + = | , ; . [ ] ?`
- 数学字符转义: `\mathbb{N} \mathcal{M} \mathfrak{M} \mathcal{L} \in \subseteq \neq \sim \forall \square`
- 上下标: `a_1^{n^2}`
- 数学环境: 矩阵、分段函数、分式等
- 文本自动转义无效: `-- "`
- 空格无效, 符号间距自动设置



# 符号转义的本质

- ASCII、Unicode 等符号编码；最新 Unicode 9.0 版（16 年 6 月）规定了 12,8237 个符号
- 符号转义 1：“助记符”，通过键盘字符的序列得到编码中的符号



# 符号转义的本质

- ASCII、Unicode 等符号编码；最新 Unicode 9.0 版（16 年 6 月）规定了 12,8237 个符号
- 符号转义 1：“助记符”，通过键盘字符的序列得到编码中的符号
- 字体：将符号编码中规定的符号具体实现
- 字体文件格式：
  - 点阵：PCF、BDF、FON、PSF、AFM 等
  - 矢量：TrueType (Apple)、Postscript (Type 1 等; Adobe)、OpenType (MS)
  - 笔画：METAFONT (Knuth)



# 符号转义的本质

- ASCII、Unicode 等符号编码；最新 Unicode 9.0 版（16 年 6 月）规定了 12,8237 个符号
- 符号转义 1：“助记符”，通过键盘字符的序列得到编码中的符号
- 字体：将符号编码中规定的符号具体实现
- 字体文件格式：
  - 点阵：PCF、BDF、FON、PSF、AFM 等
  - 矢量：TrueType (Apple)、Postscript (Type 1 等; Adobe)、OpenType (MS)
  - 笔画：METAFONT (Knuth)
- 符号转义 2：“输入法”，给定字体，通过键盘输入字符“取出”字体中相应字符



# 字体样式

字体族	带参数命令	声明命令	TeX 兼容命令
罗马/衬线	<code>\textrm\{...\}</code>	<code>\rmfamily</code>	<code>\rm</code>
无衬线	<code>\textsf\{...\}</code>	<code>\sffamily</code>	<code>\sf</code>
打字机/等宽	<code>\texttt\{...\}</code>	<code>\ttfamily</code>	<code>\tt</code>

字形	带参数命令	声明命令	TeX 兼容命令
直立	<code>\textup\{...\}</code>	<code>\upshape</code>	
意大利	<code>\textit\{...\}</code>	<code>\itshape</code>	<code>\it</code>
斜体	<code>\textsl\{...\}</code>	<code>\slshape</code>	<code>\sl</code>
小型大写	<code>\textsc\{...\}</code>	<code>\scshape</code>	<code>\sc</code>

系列/重量	带参数命令	声明命令	TeX 兼容命令
中等	<code>\textmd\{...\}</code>	<code>\mdseries</code>	
宽粗	<code>\textbf\{...\}</code>	<code>\bfseries</code>	<code>\bf</code>

注：TeX 兼容命令在叠加时会出问题，尽量不使用。



# 字体样式展示

多数英文字体都缺少部分样式。这里以演示文档使用的 txfonts ( 由 Adobe Times/Helvetica、URW Nimbus 等拼合而成 ) 为例：

字体族 重量 字形	罗马/衬线		无衬线		打字机/等宽	
	中等	宽粗	中等	宽粗	中等	宽粗
直立	Logic	<b>Logic</b>	Logic	<b>Logic</b>	Logic	<b>Logic</b>
意大利	<i>Logic</i>	<b><i>Logic</i></b>	同斜体	同斜体	<i>Logic</i>	<b><i>Logic</i></b>
斜体	<i>Logic</i>	<b><i>Logic</i></b>	<i>Logic</i>	<b><i>Logic</i></b>	<i>Logic</i>	<b><i>Logic</i></b>
小型大写	LOGIC	<b>LOGIC</b>	LOGIC	<b>LOGIC</b>	LOGIC	<b>LOGIC</b>

注 1: 命令 `\textnormal{...}` 中的文字采用文档默认样式

注 2: `{...}` 是局部环境, 其中命令的效果不会扩及花括号外



# 中文字体样式

汉语字体并无英语中那么多样式，但需要约定在各种情况下如何处理，下表是 CTeX 预设的转换方式：

字体族	罗马/衬线		无衬线		打字机/等宽	
重量	中等	宽粗	中等	宽粗	中等	宽粗
字形						
直立	宋体	黑体	黑体	<b>黑体加粗</b>	仿宋	仿宋加粗
意大利	楷体	黑体	黑体	<b>黑体加粗</b>	仿宋	仿宋加粗
斜体	楷体	黑体	黑体	<b>黑体加粗</b>	仿宋	仿宋加粗
小型大写	宋体	黑体	黑体	<b>黑体加粗</b>	仿宋	仿宋加粗

注：中英文字体分开设置，X<sub>3</sub>L<sup>A</sup>T<sub>E</sub>X 中可方便地对上述设定加以修改。



# 字号

- L<sup>A</sup>T<sub>E</sub>X 10 档字号命令：
  - `\tiny`、`\scriptsize`、`\footnotesize`、`\small`、`\normalsize`、`\large`、`\Large`、`\LARGE`、`\huge`、`\Huge`
  - 上述为相对大小，受文档基本字号影响
- CTeX 宏包提供了 `\zihao{...}` 命令：8 号到初号共 16 档
- 设置任意字号：`\fontsize{尺寸}{行距}\selectfont`

注 1：尺寸单位有 pt, pc, cm, mm, ex, em 等

注 2：局部调整字号记得放在 `{...}` 内



# 符号输入

如何知道某个符号的代码?

- 通过编辑器上的按钮输入 (不推荐!)



# 符号输入

如何知道某个符号的代码?

- 通过编辑器上的按钮输入
- 通过编辑器上的按钮找出代码然后记住
- 查阅 “The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List” 中的 symbols 文档
- Detexify (<http://detexify.kirelabs.org>) 应用
- 终极大法：直接输入 Unicode 代码
  - ① 从 TeX 系统已安装的字体（或相应代码表）中找到符号的编码
  - ② `\usefont{<encoding>}{<family>}{<series>}{<shape>}`
  - ③ `\symbol{编码}`
  - ④ 注意：有些字体会对具体字符重定义！
- 需要慢慢积累



# 汉字编码

## 国标系列

标准	汉字数
GB2312-1980	6763
GB12345-90 (主繁)	6866
GB13000.1-1993	20902
GBK (非国标)	21003
GB18030-2000	27533
GB18030-2005	70244

Windows 使用 GB 编码

Unix 类使用 Unicode

## Unicode 系列

版本	时间	CJK (统一/其它)	合计
1.0.1	1992	20902/12	20914
3.0	1999	6582/0 (Ext-A)	27496
3.1	2001	42711/0 (Ext-B)	70207
4.1	2005	22/0	70229
5.1	2008	8/0	70237
5.2	2009	4149/8 (Ext-C)	74394
6.0	2010	222/0 (Ext-D)	74616
8.0	2015	5762/9 (Ext-E)	80388

X<sub>Y</sub>T<sub>E</sub>X 和 LuaT<sub>E</sub>X 原生支持 Unicode (UTF-8)

# ( 三 )

## 撰写英文期刊/会议论文

1. LNCS 文档类
2. Elsevier 文档类
3. ACM 文档类

文档下载:

<http://xixilogic.org/docs/latex-templates.zip>

# ( 四 )

## 撰写中文期刊/会议论文

- 逻辑学中文期刊通常不支持  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  格式
- 逻辑学中文会议通常没有文档格式要求
- 《逻辑学研究》期刊支持  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  并提供模板

# (五)

## 撰写中文学位论文、书籍

见讲义



# 编译

- 书籍编译方式：

`xelatex -> bibtex -> zmakeindex -> xelatex -> xelatex`

- 如果是不存在交叉引用（以及目录、参考文献、索引等）的论文，`xelatex` 一次编译即可
- TeXstudio 默认会自动判断编译几次（有时很让人恼火）



# 编译

- 书籍编译方式：

`xelatex -> bibtex -> zmakeindex -> xelatex -> xelatex`

- 如果是不存在交叉引用（以及目录、参考文献、索引等）的论文，`xelatex` 一次编译即可
- TeXstudio 默认会自动判断编译几次（有时很让人恼火）
- 编译提示：TeXstudio 和 TeXworks 等编辑器支持

```
% !TEX encoding = UTF-8
```

```
% !TEX program = xelatex
```

```
% !TEX root = ../main.tex
```

```
% !TEX spellcheck = fr-FR
```

```
% !BIB program = bibtex
```



# 多文件编译

`\include{<文件名>}` %加载章文件，自动使用设置空白页

`\input{<文件名>}` %加载文件内容，不做额外处理



# 风格设计

## 风格设计应独立于具体内容

- 封面、标题页设计
  - titlepage 环境支持标题页设计，但不理想；
  - 如果是书籍出版，可交由出版社处理
  - ConT<sub>E</sub>Xt 的封面设计功能很强大
- 版式设定
  - 使用 geometry 宏包设置（不建议使用 L<sup>A</sup>T<sub>E</sub>X 自带设置方式）
  - 书籍一般使用双面模式：twoside 选项
  - 行距：setspace, \linespread, \setstretch 等
- 章节格式设计：参考 C<sub>T</sub>E<sub>X</sub> 宏包文档
- 批注版、竖排版、多栏设计等都有相应处理办法
- \* 10.5pt: (1) fix-cm (2) 11 磅缩印



# 中文处理

## 中文字体中文样式



# 目录处理的难题

- 出版社往往根据 Word 中的效果来要求你设置目录样式，其中一些可能跟 CTeX 宏包的默认设置不一致，例如：
  - 章节名和页码之间的连接点应上下居中（默认靠下）
  - 章这一级不加连接点
  - 前言、图片目录、索引等列在目录中，但目录本身不在目录中
  - 目录首页不要页眉页脚，奇偶页页眉要区分



# 目录处理

- 多类目录:

`\tableofcontents`

`\listoffigures`

`\listoftables`

理论上, 任何环境都可以生成一个目录列表



# 目录处理

- 多类目录：

`\tableofcontents`

`\listoffigures`

`\listoftables`

理论上，任何环境都可以生成一个目录列表

- 目录修改：tocloft 宏包

- 手动将某节加入目录

`\phantomsection` %参见hyperref宏包

`\addcontentsline{toc}{section}{节名}`

- 规定目录列入几级章节：`\setcounter{tocdepth}{2}`

- 将参考文献、索引等列入目录，但不含目录本身：

`\usepackage[nottoc]{tocbibind}`



# 参考文献处理 (1)

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  默认文献格式和引擎:  $\text{BibT}_{\text{E}}\text{X}$

- 参考文献库文件后缀: .bib
- bib 编辑软件: jabref, bibdesk, mendeley 等



# 参考文献处理 (1)

**L<sup>A</sup>T<sub>E</sub>X 默认文献格式和引擎: BibT<sub>E</sub>X**

- 参考文献库文件后缀: .bib
- bib 编辑软件: jabref, bibdesk, mendeley 等

- 
- 加载参考文献库:

`\bibliographystyle{plain}` %参考文献样式

`\bibliography{wenxian}` %文献库文件名: wenxian.bib

- 也可自行书写 thebibliography 环境



# 参考文献处理 (1)

L<sup>A</sup>T<sub>E</sub>X 默认文献格式和引擎: BibT<sub>E</sub>X

- 参考文献库文件后缀: .bib
- bib 编辑软件: jabref, bibdesk, mendeley 等

- 
- 加载参考文献库:

`\bibliographystyle{plain}` %参考文献样式

`\bibliography{wenxian}` %文献库文件名: wenxian.bib

- 也可自行书写 thebibliography 环境

- 
- 正文引用: `\cite{bid1,bid2}` %引用所列编号的文献

- 功能增强宏包: natbib, 以文献样式 plainnat 为例:

- `\citep{Lampport1994}` 得到 [Lampport, 1994]

- `\citet{Lampport1994}` 得到 Lampport [1994]



# 参考文献处理 (2)

## 参考文献的文献样式

- 格式考量：
  - 要不要编号？编号格式？
  - 各项的排序
  - 作者名、标题、发表时间等的格式和排序
  - 正文引用处的格式
- 常见类型：plain, unsrt, named, alpha, abbrev 等
- .bst 文件，位于 [texm-dist]/bibtex/bst 文件夹下
- 定制 bst：运行命令 `latex makebst`



# 参考文献处理 (3)

功能增强:  $\text{BIBLATEX}$ , 便于中文参考文献处理

- $\text{BIBTEX}$  格式过于简单, 功能不够强
- 新架构  $\text{BIBLATEX}$  支持全 Unicode 功能、文献分割、用户字段等



# 参考文献处理 (3)

功能增强: BibLaTeX, 便于中文参考文献处理

- BibTeX 格式过于简单, 功能不够强
- 新架构 BibLaTeX 支持全 Unicode 功能、文献分割、用户字段等

---

## ● 导言区:

- `\usepackage{biblatex}` %加载宏包
- `\addbibresource{references.bib}` %加载库文件, 可多个

## ● 文献列表区:

- `\printbibliography` %默认方式显示
- `\printbibliography[title={参考文献}]` %修改列表标题
- `\printbibliography[type=book]` %只打印书籍文献
- `\printbibliography[notttype=article]` %不打印期刊文献

## ● 更多功能参考 BibLaTeX 和 Biber 的文档



# 索引处理

- 导言区：

`\usepackage{makeidx}` %编制索引引用

`%\usepackage{splitidx}` %支持多个索引

`\usepackage{balance}` %末页平衡宏包

`\makeindex` %开启索引功能

- 正文区

`\balance` %命令之后，末页平衡

`\nobalance` %取消平衡

`\printindex` %放在需要放置索引的位置，生成索引

- 编译：`makeindex` 命令，运行于 `.idx` 文件

- UTF-8 中文索引排序：`zhmakeindex` 命令



# 距离的精细操控

$$4+3=7\$\\$$

$$\$(\neg p \rightarrow \neg \Box p)\$\\$$

$$\$\Box q = s\$\\$$

$$\$\sim = \{(1,2), (3,2)\}\$$$

$$4 + 3 = 7$$

$$(\neg p \rightarrow \neg \Box p)$$

$$\Box q = s$$

$$\sim = \{(1,2), (3,2)\}$$

$$\backslash\mathord$$

$$\backslash\mathrel$$

$$\backslash\mathop$$

$$\$\Box \mathrel \Box q = s\$\\$$

$$\$\{\sim\} = \{(1,2), (3,2)\}\$\\$$

$$\$\mathord \sim = \{(1,2), (3,2)\}\$$$

$$p \Box q = s$$

$$\sim = \{(1,2), (3,2)\}$$

$$\sim = \{(1,2), (3,2)\}$$



# 学会定义命令

`\newcommand{新命令名}[元数]{定义项}`

`\renewcommand{已定义命令名}[元数]{定义项}`

例：

- `\newcommand{\ra}{\rightarrow}`
- `\newcommand{\ab}[1]{\langle#1\rangle}`
- `\newcommand{\truthset}[2]{\langle#1\rangle^{\#2}}`
- `\newcommand{\lang}{\ensuremath{\mathcal{L}}}`
- `\renewcommand{\iff}{\text{iff}}`



# 学会定义命令

`\newcommand`{**新命令名**}[**元数**]{**定义项**}

`\renewcommand`{**已定义命令名**}[**元数**]{**定义项**}

例:

- `\newcommand{\ra}{\rightarrow}`
- `\newcommand{\ab}[1]{\langle#1\rangle}`
- `\newcommand{\truthset}[2]{\langle#1\rangle^{\#2}}`
- `\newcommand{\lang}{\ensuremath{\mathcal{L}}}`
- `\newcommand{\lang}{\ensuremath{\mathcal{L}}\xspace}`  
( 需 `xspace` 宏包支持 )
- `\renewcommand{\iff}{\text{iff}}`



# 盒子与间距

- 调整距离

- 行内: `\hspace{...}` `\hfill` `\quad` `\qquad` `\xspace`
- 行间: `\vspace{...}` `\vfill` `\smallskip` `\medskip` `\bigskip`
- 页间: `\newpage` `\clearpage` `\cleardoublepage`

- 盒子

- 封装: `\mbox` `\parbox` `\fbox` `minipage` 环境
- 盒子调整: `\raisebox` `\scalebox` `\rotatebox`



# L<sup>A</sup>T<sub>E</sub>X 实用工具

- 英文拼写检查：
  - Excalibur (for Mac)  
<http://excalibur.sourceforge.net>
- 新旧内容对比: `latexdiff`
- 小型编辑器：
  - LaTeXit (for Mac)  
<http://www.chachatelier.fr/latexit/>
  - Ekee (for Linux)  
<http://rlehy.free.fr/>
  - Laeqed (Java 程序, 仅支持 png 输出)



# 制作演示文档

- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ : Beamer 宏包
- $\text{C}^{\text{o}}\text{nT}_{\text{E}}\text{Xt}$ : 原生, 超强!



# TEX 还能做什么？

- 编诗集？
- 做棋谱？
- 做贺卡？
- PDF 时钟？



# 入门阅读文献

- 刘海洋,《 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  入门》, 电子工业出版社, 2013
- 包太雷,《 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  Notes: 雷太赫排版系统简介 (第二版)》  
下载: 搜索 lnotes2.pdf
- *LaTeX Primer*
- *A (Not So) Short Introduction to  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$*   
中文翻译:《一份不太简短的  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  介绍》  
下载: CTAN 网站搜索 “lshort” 和/或 “lshort-zh-cn”
- *$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  for Word Processor Users*  
下载: CTAN 网站搜索 “latex4wp”
- 学会使用搜索引擎寻找答案 (特别是英文社区)

# 谢谢!



最后  
有追求，懂取舍

内容高于形式

如果实在用不惯 L<sup>A</sup>T<sub>E</sub>X，那就放弃吧！