

L^AT_EX Writing

Kang Y

Sep 18, 2023

Contents

I L^AT_EX 科技论文排版	iii
L^AT_EX 环境安装	v
0.1 Texlive	v
0.2 Editor	v
0.2.1 Texstudio	v
宏包介绍	ix
0.3 常用宏包	ix
0.4 格式设置类宏包	ix
0.5 字体类宏包	ix
0.6 插图类宏包	ix
0.7 表格类宏包	x
0.8 公式类宏包	x
0.9 参考/注释类宏包	x
文档类型	xi
0.10 文档类型	xi
0.10.1 期刊	xi
中文双列	xi
0.11 文档结构	xi
0.12 目录及章节编号	xii
0.12.1 目录编号	xii
0.12.2 章节编号	xii
字体处理	xiii
0.13 文字格式	xiii
0.13.1 字体	xiii
0.13.2 字体大小	xiii
0.13.3 字体修饰	xiv
0.13.4 全局字体设置	xiv
0.13.5 局部字体设置	xvii
0.14 中文字体	xvii
0.15 CJK	xviii
0.15.1 中文字体	xix
0.15.2 字号、行距	xix
格式排版	xxi
0.16 段间距	xxi
0.17 缩进	xxi
0.18 行距	xxi
0.19 自动换行	xxi

0.20	两端对齐	xxi
0.21	页面布局	xxii
0.22	页眉页脚	xxii
图表处理		xxv
0.23	图形	xxv
0.23.1	常规的图片插入方法	xxv
0.23.2	调用法插入图片	xxvi
0.23.3	图片位置的排版	xxvi
0.23.4	图片引用	xxviii
0.23.5	图形编号	xxviii
0.23.6	图形的浮动和环绕	xxix
文献引用		xxxv
0.24	西文参考文献	xxxv
0.25	中文参考文献	xxxvi
0.26	格式	xxxvi
0.27	插入引文	xxxvii
0.28	引文文件	xxxvii
批注与修订		xxxix
0.29	changes	xxxix
0.30	todonotes	xxxix
0.31	annmark	xl
0.32	latexdiff	xli

Part I

L^AT_EX 科技论文排版

L^AT_EX 环境安装

0.1 Texlive

T_EX 是一种排版语言，通过命令来控制文档格式。T_EX 在设计过程中采用了所思即所得的思想，与常见的文字排版工具 Word 相反。对于一些不熟悉命令行或缺乏计算机基础的人来说使用起来门槛过高，于是美国计算机学家莱斯利·兰伯特 (Leslie Lamport) 基于 T_EX 开发了扩展版 T_EX，即 L^AT_EX。与 T_EX 相比，使用者不需要高深的计算机知识和命令行操作方法也可以使用 L^AT_EX 进行文档的排版工作。L^AT_EX 在数学公式方面的排版极为便利，尤其是在对一些复杂的表格和数学公式，也可以轻松完成。

L^AT_EX 作为一种排版语言，需要一个载体才能进行正常的工作。而让 L^AT_EX 发挥作用的一个发行版本 TexLive，目前来说是一种比较完成的持续维护更新的发行版本。对与未曾接触过 L^AT_EX 的使用者来说是比较容易使用的一个工具。L^AT_EX 和 TexLive 之间的关系好比，C 语言和 C 语言编译器之间的关系。没有编译器来对语言进行编译，机器便无法识别、工作。而在 TexLive 这个发行版本的编译器中来自带了一个简易的文本编辑器，可以用来完成一些简单的工作。

0.2 Editor

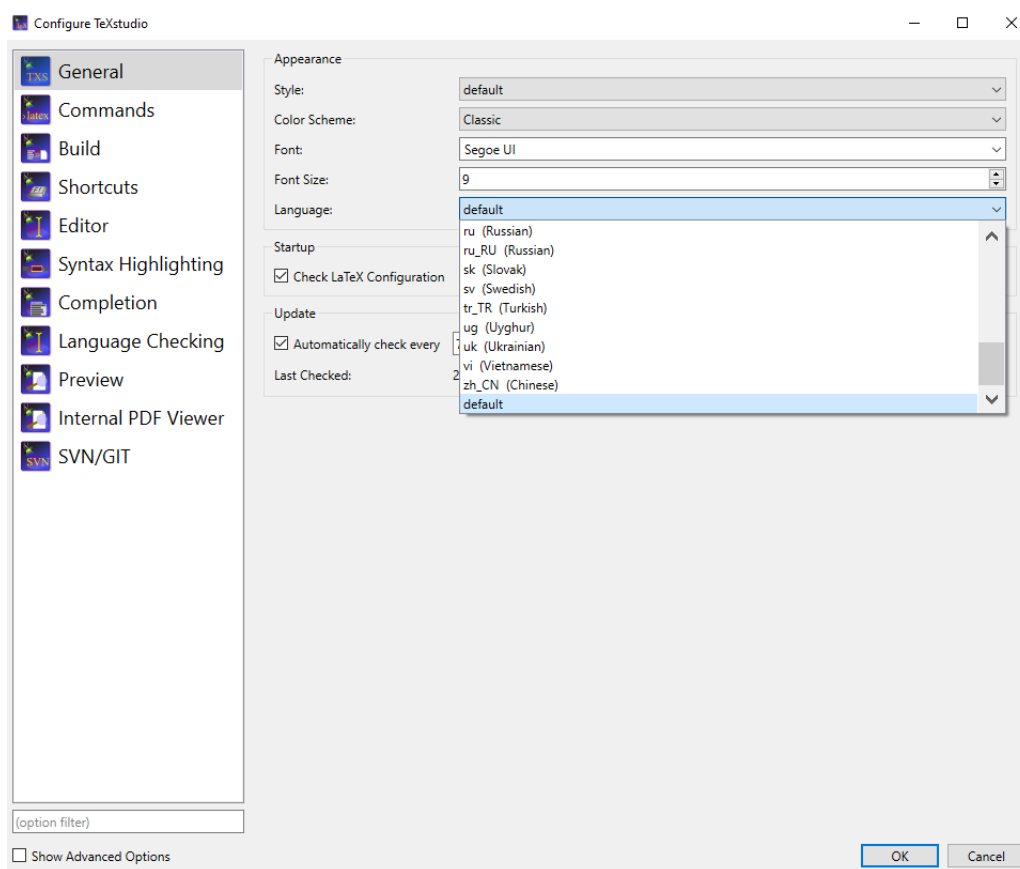
TexLive 这个发行版中附带了一个功能相对简单的编辑器 Texworks，该编辑器的界面相对简陋，对初次使用使用者来说可以完成一些简单的不含复杂表格和数学公式的文档，但面对复杂的排版任务和以 L^AT_EX 为主要排版工具的使用者来说，Texworks 编辑器的使用起来略显不便，对与提高排版效率来说不是那么容易。于此同时，有 Texstudio 和 Texmaker 两款 L^AT_EX 专用编辑器和 vim、Sublime 等兼容编辑器。

可以实现 L^AT_EX 文档编写的软件较多，通常可以分为两类：可以实现 texlive 调用、编译的编辑软件，如 Texstudio、Texworks、Emacs；通过插件来调用 texlive 实现编译的软件，如 Vim、VScode 等。texlive 自带的 Texworks 编辑器轻量化，虽然功能上没有残缺，但是不适合打开较大的文件，且每个窗口只能打开一个 tex 文档，不便于结构化文档撰写和管理，同时界面过于简陋，对初学者的使用友好程度相对较低。VSCode 虽然可以用来编写 tex 文档，但是其本身不是为编写 tex 文档而开发的，需要经过一番配置才可以运行，且软件本身过于庞大臃肿，不支持文档预览。Emacs 和 Vim 两个强大的文本编辑器，经过一番配置之后，完全可以胜任 L^AT_EX 的撰写和编译工作，但是这二者的入门学习成本相对较高。因此，重点介绍 Texstudio 编辑器。

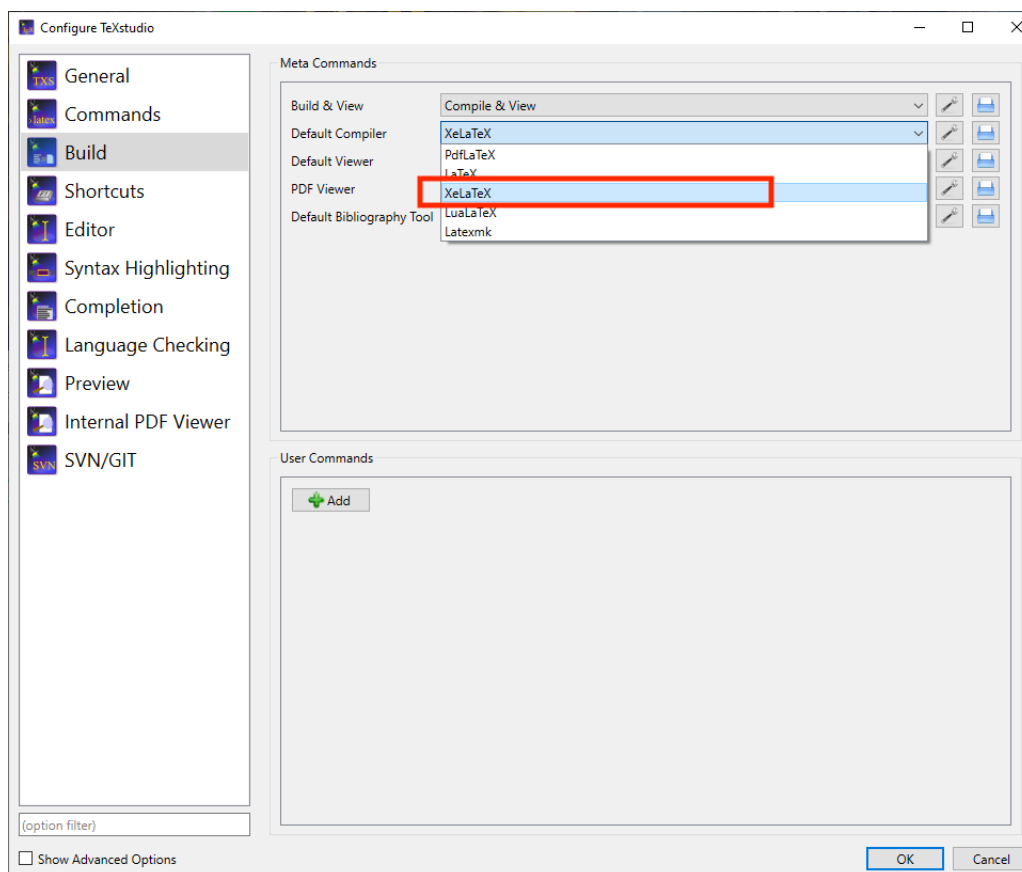
0.2.1 Texstudio

TeXstudio 是一款跨平台 (Windows、Mac、Linux) 的免费软件，用来编辑 L^AT_EX 文档。TeXstudio 软件针对 L^AT_EX 设计了众多的快捷按键和便利的功能。例如：文件结构视图、代码补全和折叠、语法高亮和语法检查、行内实时预览和代码块、内置浏览等。TeXstudio 对初学者入门 L^AT_EX 完全够用，且学习的门槛相对较低。而非常熟悉 L^AT_EX 的用户，可以在 Texstudio 软件内进行自定义宏，提高编写效率。Texstudio 软件的下载途径：官网 (目前很难访问)，sourceforge (第三方的开源软件集合)，清华大学的镜像。软件安装完成之后，用户仅需要做一点点设置工作：

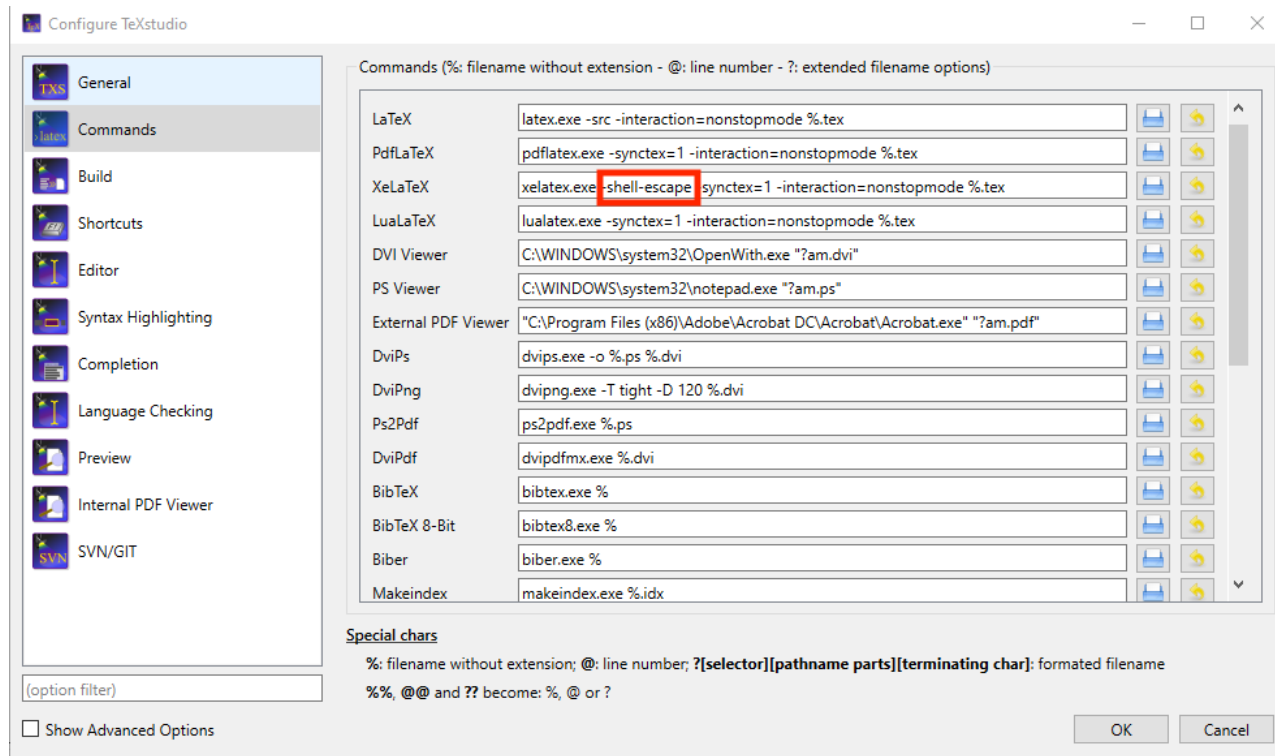
1. 用户根据自己的需要来设置显示语言和字体。读者可以根据自己的需要来设置语言和显示字体。如需要采用中文编写文件，则必须在 `option>Configure Texstudio` 中，设置编译器为 `xelatex`。如果在使用过成中用到了 `minted` 环境，则还需要在编译参数前添加一个选项：`-shell-escape`。



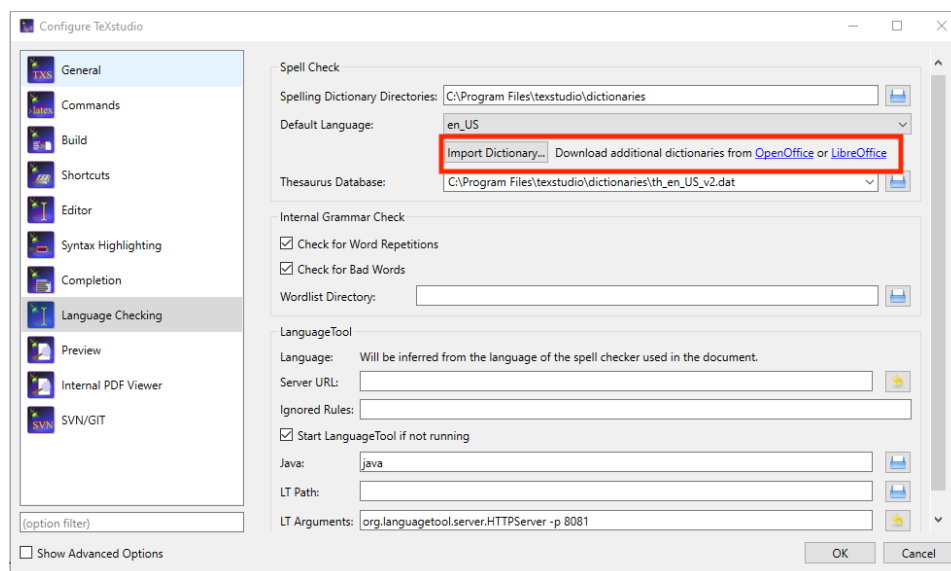
2. 根据排版的文字类型（中文或西文）来设置 Texstudio 在编译过程中调用的编译器类别：L^AT_EX 或 `xelatex`。



3. 大多数情况下，用户不需要更改 xelatex 编译器的参数。但在用到 minted 环境时，需要在 xelatex 后补充参数 ‘-shell-escape’，即 ‘xelatex.exe -shell-escape -synctex=1 -interaction=nonstopmode %tex’。



TeXstudio 提供了语言拼写检查功能，用户可以自己从软件提供的跳转链接，从 libreoffice 网站来下载相应语言的词典（词库），供软件进行拼写检查。



Texstudio 软件会对编译过程中产生的错误和警告会在软件底部的‘Message’栏显示给用户。通过双击改错误或警告，光标会自动跳转至相应的位置：如果是 T_EX 文档中的写法错误，光标会自动跳转至出现错误的行；如果是其他错误，如宏包调用问题、字体调用问题，则会跳转至相应的宏包文件。

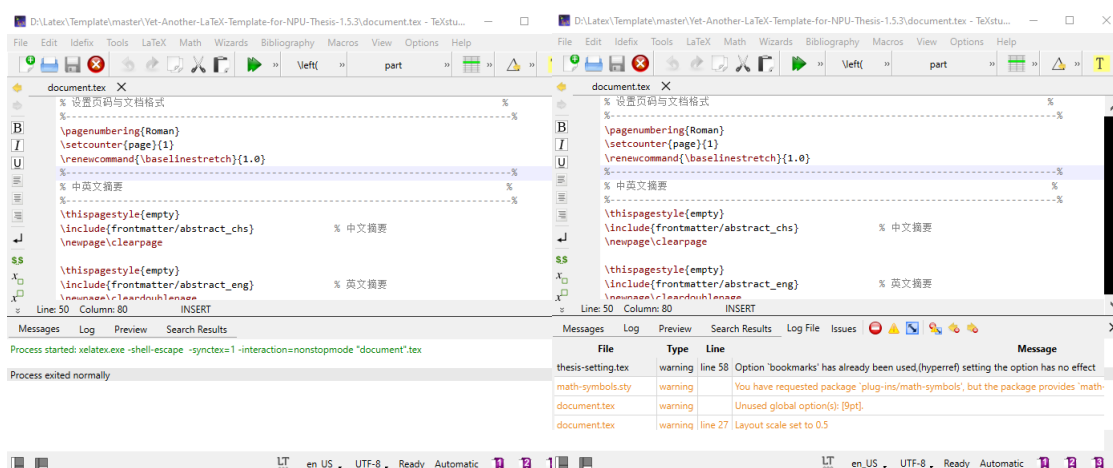


Figure 1: Message 信息栏

宏包介绍

0.3 常用宏包

L^AT_EX 中常用的宏包可以分为几个大类：整体环境设置、图形类、表格类、公式类，此处仅介绍使用率较高的一些宏包。

0.4 格式设置类宏包

ctex：中文 L^AT_EX 文档框架，默认包含了 ctexart、ctexrep、ctexbook 三种不同的文档类型。对与中文使用者来说可以免去较多的格式设定问题。

geometry：用于设定输出文章的纸张大小、页边距等尺寸相关的内容。

layout：用于对文章内容的页面布局，横版、竖版等布局进行设定。

fancyhdr：用来自定义页眉、页脚的内容、格式。

afterpage：用来处理图表在页面的浮动。*float* 和 *afterpage* 的区别是啥？

multicol：提供了全文和局部分栏处理功能。但该功能会影响浮动体（图表）的处理，建议在分栏环境内不使用浮动体，例如图、表格等。

titlesec：用来对文档内容的标题格式进行设定。一级标题、二级标题等的字体、大小、位置、颜色等进行设定。

titlesoc：用于对整个文档的目录格式管理。

hyperref：给文档的内容提供一个超文本的链接功能。

minted：用于提供格式化的代码引用环境，包含了大量的编程语言以及非编程语言。同时需要配合 Pygments（Python）软件包以及 ‘-shell-escap’ 编译参数。

0.5 字体类宏包

xeCJK：提供中文字体设置

fontspec：提供西文字体的自定义设置

0.6 插图类宏包

graphicx：用于文档内容的插图处理，插图位置、插图大小（比例缩放、非比例缩放）、插图类型以及多行图片的环境。

subfigure：用于多子图及子图标题的设置。

wrapfig：用于特殊情况的插图设定，例如文字环绕、半环绕等情形。

caption：用于对文档中的插图标题进行自定义格式设定，字体、位置、大小等。

sidecap：用于一些特殊情形的图题设定，例如外文文档的侧边图题设定。

metalogo：提供 L^AT_EX Logo。

0.7 表格类宏包

array、table：提供普通的表格设置功能。

longtable：提供了跨页表格功能。

tabularx：提供了表格内嵌套高级功能的功能。

dcolumn：表格内多位小数的对其功能，从小数点对齐。

multirow：单元格合并功能。

makecell：单元格内设置单元格。

diagbox：左上角表头内斜线表格。

booktabs：三线表。

rotfloat：旋转表格

0.8 公式类宏包

amsmath：提供数学公式的编辑环境，如行间公式和行内公式等。

ppchtex：提供化学符号。

harpoon：提供上下带箭头的字符。

0.9 参考/注释类宏包

bibtex：文献类型。

natbib：文献引用显示类型。

footbib：脚注式引用。

gbt7714：显示中文文献格式。

gloss：附录中的注释表。

listings：插入代码。

文档类型

0.10 文档类型

L^AT_EX 中的文档类型 (documentclass) 可以分为：期刊、报告、书籍、幻灯片 (beamer)、书信等。

0.10.1 期刊

中文双列

在期刊类型的排版的文档，为了节省版面，经常需要安排双列排版。在 documentclass 中设定相关选取为：twocolumn 即可实现双列排版。但文章标题和摘要（中英文）却又是单列排版，要想实现这种排版，在导言区之后，采用如下代码结构，即可实现标题和中英的单列排版。

```
\twocolumn[
\begin{@twocolumnfalse} % let the two column mode fail temporary.
  \maketitle % maketitle command must put in this environment.
  \begin{document} % after this, abstract and main session start.
  \begin{abstract}
    \textbf{Abstract}:
    \noindent Keywords:
  \end{abstract}
\end{@twocolumnfalse}]
```

该模式下，占据单列位置的图片可以在 h 参数的作用下，在插入图片的位置显示；而双列的图片和表格的位置则很难，会被默认放在页面的底部或顶部。

0.11 文档结构

Book 类型的文章结构：

\part,\chapter,\section,\subsection,\subsubsection。一般用来排版较大的学术出版物（或学位论文）。

Arctile 类型的文章结构：

\part,\section,\subsection,\subsubsection, 以及 Abstract。一般用来排版会议、期刊稿件等文档。

Report 类型的文章结构：

\part,\chapter,\section,\subsection,\subsubsection, Abstract 单独成页且包含页码。

0.12 目录及章节编号

0.12.1 目录编号

目录在不同的文章类型中的要求不尽相同，学术文章中很少用的目录，而学位论文的排版中则需要出现目录，而且列表和插图也要体现在目录中，因此目录进行深度控制。在 \LaTeX 中采用 $\text{\setcounter{tocdepth}{X}}$ 命令来实现不同深度的目录。 $\text{\setcounter{secnumberdepth}{Y}}$ 来实现章节编号的深度。

- -1 part
- 0 chapter
- 1 section
- 2 subsection
- 3 subsection
- 4 paragraph
- 5 subparagraph

利用 \renewcommand 命令，将文档的一些标题从英文转化为中文标题。如： $\text{\renewcommand\abstractname\hei摘要}$ ，将英文 abstractname 改为摘要。

0.12.2 章节编号

\LaTeX 默认的章节编号大多是以阿拉伯数字开始的，但是对于一些经常采用非阿拉伯数字编号的使用者来说就显得有点不友好。因此，在进行章节编号的时候，使用者就需要根据自己的需要对章节的编号进行自定义。 \LaTeX 中的提供了几类编号方式： \arabic 阿拉伯数字， \roman 小写的罗马数字， \Roman 大写的罗马数字， \alph 小写字母， \Alph 大写字母。如果使用者希望采用非阿拉伯数字的方式进行编号，利用 \renewcommand 命令进行定义编号的格式。在整理液压传动的时候对章节的编号进行自定义的代码如下：

```
 $\text{\renewcommand\thesection}\color{RGB}{60,113,183}$ 
 $\rightarrow \text{\arabic{chapter}}-\text{\arabic{section}}$  % 设置 section 的格式为章序号-节序号
 $\text{\renewcommand\thesubsection}\text{\chinese{subsection}}、$  % 子节的序号为中文序号 + 顿号
 $\rightarrow \text{\renewcommand\thesubsubsection}\text{\arabic{subsubsection}}.$ 
```

字体处理

0.13 文字格式

0.13.1 字体

在 L^AT_EX 中要使用自定义的字体，先在导言区调用字体宏包 `\usepackage{fontspec}`，然后用三个命令来设置主字体族、无衬线字体族和打字机字体族 `\setmainfont{fontName}`，`\setsansfont{fontName}`，`\setmonofont{fontName}`。当然也可以根据自己的需要对三种字体进行自定义设置，

在正文中采用 `\textrm{}` 和 `\rmfamily`，`\textsf{}` 和 `\sffamily`，`\texttt{}` 和 `\ttfamily` 来调用主字体族、无衬线字体族和打字机字体族。两类命令的区别在于前者是将参数（大括号内的部分）变成对应的字体族，而后者是将该命令作用的区域范围内的文字秉承对应的字体族。例如：`{\rmfamily This is rmfamily and \textrm{this is textrm}}, this is the main font.`

那么排版出来的结果是

This is a rmfamily and This is textrm, this is a the main font.

使用 `\textup{}` 和 `\upshape`，`\textit{}` 和 `\itshape` 来分别主动调用当前字体族的直立体与意大利斜体，用 `\textmd{}` 和 `\mdseries`，`\textbf{}` 和 `\bfseries` 来分别调用当前字体形状的正常体与粗体。

如果想调用第四种低体，可以采用 `\newfontfamily\newfontname{FontName}` 来自定义新字体族，FontName 为新字体名称；在正文中采用 `{\newfontname This is a new font}`。

0.13.2 字体大小

L^AT_EX 中字体的大小取决于定义文档类型时设定的字体大小，又被称为绝对字体大小。表1列出了标准文档中的字体绝对大小，单位为 pt。例如：

```
\documentclass[a4,12pt]{article}
```

此处定义了一个期刊文章类型的文档，采用 A4 纸张大小，绝对字体大小 12pt。如在文章中采用 `\tiny` 字体，则字体的大小会被认为是 6pt，参见表1。其他的字体大小可按照类似的方法设定。但在实际使用过程中也可以根据需要使用 `\fontsize{<size>}{<base line-skip>}` 进行任意设定。

Table 1: 标准文档类中的字号大小。

字号	10pt	11pt	12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	10.95pt
<code>\normalsize</code>	10pt	10.95pt	12pt

Continued on next page

Continued from previous page

字号	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14.4pt
<code>\Large</code>	14.4pt	14.4pt	17.28pt
<code>\LARGE</code>	17.28pt	17.28pt	20.74pt
<code>\huge</code>	20.74pt	20.74pt	24.88pt
<code>\Huge</code>	24.88pt	24.88pt	24.88pt

0.13.3 字体修饰

在 \LaTeX 中对文字加粗、斜体、强调等操作分别通过以下代码实现：

```
% 加粗
\textbf{} % 文本加粗
\boldmath{} % 数学环境内加粗，需要调用 amsmath 宏包
\boldsymbol{} % 可以对希腊字母加粗，需要调用 amsmath 宏包

% 斜体
\textit{}

% 强调
\emph{}
```

特别的，当需要对局部文件进行下划线与其他文字进行区别的时候需要注意，默认的 `\underline` 命令可以实现文字的下划线功能，但是当下划线部分的文字段落过长需要换行的时候，该命令就无法实现了。因此，需要对此进行重点说明。在 \LaTeX 中英文和中文的下划线换行过程中需要采用不同的宏包进行处理。具体用法如下：

```
% 下划线
\underline{} % 此命令可以在文字下面生成下划线，但无法在行末进行自动换
↪ 行。

% 在中文环境中进行换行时采用以下宏包和命令
\usepackage{CJKfntef}
\uuline{}

% 在英文环境中进行换行时采用以下宏包和命令
\usepackage{ulem}
\uline
```

0.13.4 全局字体设置

\LaTeX 中的字体可以分为三类：衬线字体（罗马体）、无衬线字体和等宽字体（打字机字体）。

“*fontspec*”宏包用来设置文档中的西文字体环境。在文档中采用 `\setmainfont{} \setsansfont{} \setmonofont{} \setCJKmainfont{} \setCJKsansfont{} \setCJKmonofont{}` 三种命令分别来设置西文字体中的衬线体、无衬线体和等宽字体。在使用过程中根据自己的需要采用其中一个命令或多个名来设置相应的字体。

“*xeCJK*”宏包负责在文档中提供中日韩文字环境。在文档中要显示中日韩文字时，需要 `\usepackage{xeCJK}` 命令来调用宏包，使用 `\setCJKmainfont \setCJKsansfont \setCJKmonofont{}` 三种命令来设置相应的中文字体。

fontspec 和 *xeCJK* 的使用方法如下：


```
% 拉丁字母设置
\setmainfont{font name}[font features]% 中括号内为字体的特征，如黑体、
↪ 斜体等。
\setsansfont{font name}[font features]
\setmonofont{font name}[font features]
% 中文字体设置
\setCJKmainfont{font name}[font features]
\setCJKsansfont{font name}[font features]
\setCJKmonofont{font name}[font features]
% 举例
\setCJKmainfont{SimSun}[BoldFont=SimHei,ItalicFont=KaiTi]
```

上面的案例中 SimHei, KaiTi 为操作系统中已经存在的中文字体名称。

关于操作系统中已经存在的中文字体名称的查询方法：在 Windows 系统中打开终端程序 cmd，输入命令“fc-list :lang=zh”来查询系统中的中文字体，会输出所有中文字体在系统中的相关信息。‘fc-list’命令在系统安装 texlive 之后即可直接使用，但 cmd 程序中对中文的支持并不友好，此处展示了采用 git 软件输出的结果，如图2。该命令中的字符均为英文状态输入，包括冒号。

图2中，红色框位置的内容即为中文字体在系统中的名称，在 L^AT_EX 中调用字体，需要输入系统中的字体名称。

在使用 ctex 宏包时，经常会遇到各种 warning，如：

warning 1

```
Package fontspec warning: Font "FandolSong-Regular" does not contain
requested
(fontspec)                Script ``CJK"
```

warning 2

```
Package xeCJK Warning: Redefining CJKfamily ``\CJKrmdefault'
(xeCJK)                (FandolSong-Regular(0))
```

warning 3

```
Package xeCJK Warning: Redefining CJKfamily ``\CJKsfdefault'
(xeCJK)                (FandolHei-Regular)
```

warning 4

```
Package xeCJK Warning: Redefining CJKfamily ``\CJKttdefault'
(xeCJK)                (FandolFang-Regular)
```

warning1 和 warning2 可以通过对 ctex 包添加参数来实现。

```
\usepackage[fontset=none]{ctex}
```

warning2-4 可以采用打补丁的方式进行修正。

```
\usepackage{ctex}
\usepackage{xpatch}
```

```

Clancey@Akido MINGW64 ~ (master)
$ fc-list :lang=zh
C:/WINDOWS/fonts/STSONG.TTF: STSong, 华文宋体:style=Regular
C:/WINDOWS/fonts/FZSTK.TTF: FZShuTi, 方正舒体:style=Regular
C:/WINDOWS/fonts/simsun.ttc: NSimSun, 新宋体:style=Regular, 常规
C:/WINDOWS/fonts/simfang.ttf: FangSong, 仿宋:style=Regular, Normal, obyčejné, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/WINDOWS/fonts/Dengb.ttf: DengXian, 等线:style=Bold
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolKai-Regular.otf
C:/WINDOWS/fonts/SIMLI.TTF: LiSu, 隶书:style=Regular
C:/WINDOWS/fonts/msyh.ttc: Microsoft YaHei UI:style=Regular, Normal, obyčejné, Normale, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolHei-Bold.otf: F
C:/WINDOWS/fonts/Deng.ttf: DengXian, 等线:style=Regular
C:/texlive/2022/texmf-dist/fonts/truetype/public/arthic-ttf/bkai00mp.ttf: AR
=Regular
C:/WINDOWS/fonts/STCAIYUN.TTF: STCaiyun, 华文彩云:style=Regular
C:/WINDOWS/fonts/msyh.ttc: Microsoft YaHei, 微软雅黑:style=Regular, Normal, obyčejné, Normale, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/WINDOWS/fonts/msyh1.ttc: Microsoft YaHei, 微软雅黑, Microsoft YaHei Light, 微
C:/WINDOWS/fonts/STXINGKA.TTF: STXingkai, 华文行楷:style=Regular
C:/WINDOWS/fonts/Dengl.ttf: DengXian, 等线, DengXian Light, 等线 Light:style=Lig
C:/WINDOWS/fonts/STZHONGS.TTF: STZhongsong, 华文中宋:style=Regular
C:/WINDOWS/fonts/msyh1.ttc: Microsoft YaHei UI, Microsoft YaHei UI Light:style
C:/texlive/2022/texmf-dist/fonts/truetype/public/arthic-ttf/gkai00mp.ttf: AR
=Regular
C:/WINDOWS/fonts/FZYT.K.TTF: FZYaoTi, 方正姚体:style=Regular
C:/WINDOWS/fonts/msjhl.ttc: Microsoft JhengHei UI, Microsoft JhengHei UI Light
C:/WINDOWS/fonts/STHUPO.TTF: STHupo, 华文琥珀:style=Regular
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolSong-Regular.ot
C:/WINDOWS/fonts/msjhbdt.ttc: Microsoft JhengHei UI:style=Bold, Negreta, tučné, f
Gras, Félkövé, Grassetto, Vet, Halvfet, Pogrubbyony, Negrito, Полужирный, Fet, Kalın, K
C:/WINDOWS/fonts/simhei.ttf: SimHei, 黑体:style=Regular, Normal, obyčejné, Standa
le, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/WINDOWS/fonts/STLITI.TTF: STLiti, 华文隶书:style=Regular
C:/WINDOWS/fonts/msjht.ttc: Microsoft JhengHei, 微软正黑體:style=Regular, Normal
ali, Normál, Normale, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/WINDOWS/fonts/simsun.ttc: SimSun, 宋体:style=Regular, 常规
C:/WINDOWS/fonts/msjhbdt.ttc: Microsoft JhengHei, 微软正黑體:style=Bold, Negreta
havoitu, Gras, Félkövé, Grassetto, Vet, Halvfet, Pogrubbyony, Negrito, Полужирный, Fet
C:/WINDOWS/fonts/msjht.ttc: Microsoft JhengHei UI:style=Regular, Normal, obyčej
ál, Normale, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/WINDOWS/fonts/SIMYOU.TTF: YouYuan, 幼圆:style=Regular
C:/WINDOWS/fonts/STXINWEI.TTF: STXinwei, 华文新魏:style=Regular
C:/texlive/2022/texmf-dist/fonts/truetype/public/arthic-ttf/bsmi00lp.ttf: AR
:style=Regular, Regular
C:/texlive/2022/texmf-dist/fonts/truetype/public/arthic-ttf/gbsn00lp.ttf: AR
e=Regular
C:/WINDOWS/fonts/msjhl.ttc: Microsoft JhengHei, 微软正黑體, Microsoft JhengHei
t, Regular
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolHei-Regular.otf
C:/WINDOWS/fonts/STKAITI.TTF: STKaiti, 华文楷体:style=Regular
C:/WINDOWS/fonts/STXIHEI.TTF: STXihei, 华文细黑:style=Regular
C:/WINDOWS/fonts/simkai.ttf: KaiTi, 楷体:style=Regular, Normal, obyčejné, Standar
e, Standaard, Normalny, Обычный, Normálne, Navadno, Arrunta
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolSong-Bold.otf:
C:/WINDOWS/fonts/msyhbdt.ttc: Microsoft YaHei UI:style=Bold, Negreta, tučné, fed,
s, Félkövé, Grassetto, Vet, Halvfet, Pogrubbyony, Negrito, Полужирный, Fet, Kalın, Krep
C:/texlive/2022/texmf-dist/fonts/opentype/public/fandol/FandolFang-Regular.ot
Regular
C:/WINDOWS/fonts/msyhbdt.ttc: Microsoft YaHei, 微软雅黑:style=Bold, Negreta, tučn
tu, Gras, Félkövé, Grassetto, Vet, Halvfet, Pogrubbyony, Negrito, Полужирный, Fet, Kali
C:/WINDOWS/fonts/STFANGSO.TTF: STFangsong, 华文仿宋:style=Regular

```

Figure 2: fc-list 命令结果

0.13.5 局部字体设置

在 \LaTeX 中如需要对正文中局部字体类型进行自定义, 可以通过两类命令来实现: `\textrm{}` 和 `\rmfamily, \textsf{}` 和 `\sffamily, \texttt{}` 和 `\ttfamily`。这三组命令分别调用衬线字体 (罗马体)、无衬线字体、等宽字体 (打字机字体)。这两类命令的区别在于: 前者是将参数 (括号内的部分) 设置成对应的字体族, 而后者将该命令的作用区域内的文字变成对应的字体族。

```
1 \textrm{This is a example.} % text 系列命令对大括号内的字体起作用
2 {\rmfamily This is a example.} %family系列命令,
   ↳ 需要将自己和需要定义字体的内容都放入大括号内。
```

0.14 中文字体

现行的 \LaTeX 版本使用 UTF-8 作为默认的文字编码。使用拉丁字母的文档保存为 UTF-8 编码, 即可采用 `pdflatex` 进行直接编译, 但是非拉丁字母, 如西里尔字母、希腊字母、阿拉伯字母及东南亚文字等, 依然无法直接在 \LaTeX 中直接使用。而较为现代的 \TeX 引擎, 如 \XeTeX 和 \LuaTeX , 将非拉丁字母文档保存为 UTF-8 编码, 借助 `fontspec` 宏包即可实现非拉丁字母的输入。特别的, 在使用 \LaTeX 进行中文排版时, 需要处理对中文字体的支持, 包括: 字体设置、行距、断句、标点符号的位置、编译引擎的选择 (\XeTeX) 等问题。

如果作者在有中文排版需要时, 需要在 \LaTeX 中使用中文字体。中文字体的设置方法有两种。第一种采用 `ctex` 文档类型, 将英文文档的对应类型变为缩写并在前面加 `ctex`。如 `ctexart`, `ctexrep` 等。直接使用命令 `\setCJKmainfont{fontName}`, `\setCJKsansfont{fontName}`, `\setCJKmonofont{fontName}` 来分别设置正文的主中文字体族, 无衬线中文字体族和打字机中文字体族。

第二种方法, 在 `tex` 文档的导言区调用 `\usepackage{xCJK}` 宏包, 然后使用命令 `\setCJKmainfont{fontName}`, `\setCJKsansfont{fontName}`, `\setCJKmonofont{fontName}` 来分别设置正文的主中文字体族, 无衬线中文字体族和打字机中文字体族。但这种方法只定义了字体类型, 对中文文章结构并未做优化, 如 `section` 标题依旧为英文, 需要作者自行修改。`ctex` 类文档对中文结构的文档做了优化。

在进行中文文献排版过程中, 写作人员会被要求采用按一定的字号来设置字体, 如一号、二号等。在此, 提供便利的方法, 以供读者在排版供中使用。

```
\newcommand{\yihao}{\fontsize{26pt}{36pt}\selectfont} % 一
↳ 号, 1.4 倍行距
\newcommand{\erhao}{\fontsize{22pt}{28pt}\selectfont} % 二
↳ 号, 1.25 倍行距
\newcommand{\xiaoer}{\fontsize{18pt}{18pt}\selectfont} % 小
↳ 二, 单倍行距
\newcommand{\sanhao}{\fontsize{16pt}{24pt}\selectfont} % 三
↳ 号, 1.5 倍行距
\newcommand{\xiaosan}{\fontsize{15pt}{22pt}\selectfont} % 小
↳ 三, 1.5 倍行距
\newcommand{\sihao}{\fontsize{14pt}{21pt}\selectfont} %
↳ 四号, 1.5 倍行距
\newcommand{\banxiaosi}{\fontsize{13pt}{19.5pt}\selectfont} % 半
↳ 小四, 1.5 倍行距
\newcommand{\xiaosi}{\fontsize{12pt}{18pt}\selectfont} %
↳ 小四, 1.5 倍行距
```


在

LaTeX 中设置中文字体时，通常采用 `\CJKfamily` 命令来实现。例如：`\CJKfamily` 命令，自身如果不带任何参数，会对命令之后的所有文本起作用。而通常在进行科技论文的写作中会采用多种字体，因此，需要采用 `\CJKfamilysong`，`\CJKfamilyhei` 等命令来设置字体。设置为黑体。

0.15.1 中文字体

```
\newcommand{\song}{\CJKfamily{song}} % 宋体
\newcommand{\fs}{\CJKfamily{fs}} % 仿宋体
\newcommand{\kai}{\CJKfamily{kai}} % 楷体
\newcommand{\hei}{\CJKfamily{hei}} % 黑体
\newcommand{\li}{\CJKfamily{li}} % 隶书
```

0.15.2 字号、行距

设置字体的大小需要在导言区加入 `\usepackage{type1cm}`，其中 `cm` 为 Computer Modern 的缩写。

`\fontsize` 字号行距 这个命令对其后所有文本都起作用，在使用此命令后需要用 `\selectfont` 才能使字体大小设置起作用。

字号	初号	小初	一号	小一	二号	小二	三号	小三
大小	42pt	36pt	26pt	24pt	22pt	18pt	16pt	15pt
1.5 倍行距	63pt	54pt	39pt	36pt	33pt	27pt	24pt	22.5pt
字号	四号	小四	五号	小五	六号	小六	七号	小七
大小	14pt	12pt	10.5pt	9pt	7.5pt	6.5pt	5.5pt	5pt
1.5 倍行距	21pt	18pt	15.75pt	13.5pt	11.25pt	9.75pt	8.2pt	7.5pt

错误：

```
\LaTeX{} font warning: font shape `OT1/cmr/m/n' in size <15> not
↪ available
```

```
\LaTeX{} font warning: size substitutions with differences
up to 0.6 pt have occurred
```

解决方案：`\usepackage{type1cm}`

格式排版

0.16 段间距

在不同的段之间，也可以设置段间距（默认为 0）。通过 `\setlength{\parskip}{parSkip}` 来实现。比如说，我想设置段间距为 2em，则使用 `\setlength{\parskip}{2em}` 即可。这样，在不同两段之间的距离，为段间距 + `\baselineskip` 的距离。这里值得注意的是，由于 `\\` 是断行不是分段，所以由 `\\` 引起的下一行与本行之间的距离，只有 `\baselineskip`，而不加上 `\parskip`

0.17 缩进

在 `article` 及 `ctexart` 中，都默认给每一段的段首进行了缩进（在下一篇文章中我们会看到，`article` 中的每一个章节后的首段不会缩进）。我们可以通过 `\setlength{\parindent}{parIndent}` 来控制缩进距离，比如说，我想控制段首缩进 2em，则应写 `\setlength{\parindent}{2em}`。这个命令会使该命令之后的所有段的缩进都变成这个值。如果要取消全部段落的段首缩进，则可以写 `\setlength{\parindent}{0em}`。如果要单独取消某一行的段首缩进，则在该行的段首写 `\noindent` 即可。

0.18 行距

因此，在我们改变行距时，如果想把行距调整成精确的值，一般用 `\fontsize{fontSize}{baseLineSkip}\selectfont` 来实现；如果想成比例地调整行距，比如单倍行距，双倍行距，则使用 `\linespread{lineSpread}\selectfont` 来实现。比如说，双倍行距就是 `\linespread{2}\selectfont`。

此外，我们也可以用一个名叫 `setspace` 的宏包。也就是说，在导言区使用 `\usepackage{setspace}`。然后使用 `\setstretch{lineSpread}` 来设置 `linespread`（不用加）。这个命令的好处在于会同时修改数学公式、浮动体等的间距，使之与正文间距适应。

0.19 自动换行

`LaTeX` 中换行的方法针对文字（西文和中文）在操作上略有不同，`LaTeX` 默认对西文进行换行是可以采用 `\ulem package`。

对带有下划线的文本进行换行时，如果为英文内容，可采用 `ulem package` 提供的 `\uline` 命令或采用 `soul package` 提供的 `\ul` 命令来实现。

0.20 两端对齐

两端对齐常用的有两种方法：`beginsloppypar` 环境和 `ragged2e package` 配合 `\justifying` 命令。

0.21 页面布局

在 L^AT_EX 中, 设置纸张大小的参数是 `papersize`。而 `papersize` 的参数如下: `a0paper`, `a1paper`, `a2paper`, `a3paper`, `a4paper`, `a5paper`, `a6paper`, `b0paper`, `b1paper`, `b2paper`, `b3paper`, `b4paper`, `b5paper`, `b6paper`, `c0paper`, `c1paper`, `c2paper`, `c3paper`, `c4paper`, `c5paper`, `c6paper`, `b0j`, `b1j`, `b2j`, `b3j`, `b4j`, `b5j`, `b6j`, `ansiapaper`, `ansipaper`, `ansicpaper`, `ansidpaper`, `ansiepaper`, `letterpaper`, `executivepaper`, `legalpaper`。

自定义页面大小: 使用 `geometry package`, 结合 `\geometry` 命令, 设置纸张长宽 (`paperwidth=10cm`, `paperheight=15cm`), 页边距 (`left=0.2cm`, `right=2.5cm`, `top=0.5cm`, `bottom=0.5cm`), 文字区域的长宽 (`textwidth=7.3cm`) 等参数。完整命令如下:

```
\geometry{paperwidth=10cm,paperheight=15cm,left=0.2cm,right=2.5cm,
↪ top=0.5cm,bottom=0.5cm,textwidth=7.3cm,textwidth=12cm}
```

0.22 页眉页脚

在用 L^AT_EX 排版文章、书籍时, 缺省定义了四种页眉页脚的格式:

<code>empty</code>	没有页眉和页脚
<code>plain</code>	没有页眉, 页脚中部放置页码
<code>headings</code>	没有页脚, 页眉包含章节的标题和页码
<code>myheadings</code>	没有页脚, 页眉页码和使用者所定义的信息

对于 `article` 类文档缺省使用 `plain` 格式, 而 `book` 类文档则使用 `headings` 格式。一般情况下, L^AT_EX 提供的这四种格式基本可以满足大多数的排版需求。但是在某些特殊的情况下, 特别是使用者想在某一些页面使用自定义的页眉页脚或全文使用自定义的页眉页脚, 这时就会遇到很多限制和麻烦。例如, 仅想在某些页面不出现任何页眉页脚, 而其余部分按照正常情况还显示页眉页脚, 这个时候就可以使用 `\pagestyle` 命令, 其用法为 `\pagestyle{empty}`, 将其放置在需要取消页眉页脚的部分, 变可实现取消页眉页脚的作用。用法如下:

```
%\input{preface} % 引言部分
\newpage % 原文档为双面, 故此处插入空白页面
\thispagestyle{empty} % 清除引言部分的页眉页脚信息

\usepackage{cite}
```

这里是正文 `\cite{bib2,bibt2,bibt3}`

但对于那些完全需要自定义页眉页脚的格式的使用者来说, 还需要使用额外的宏包“`fancyhdr`”。该宏包的主要通过定义 `\fancyhead` 和 `\fancyfoot` 两个命令对页眉页脚进行控制, 这两个命令的参数如下。在编辑教材的时候, 由于书籍采用了双面显示, 奇数页的页眉不同且没有页脚的显示方式, 故利用“`fancyhdr`”对页眉页脚进行了重新定义。使用的代码如下。

```
\requirepackage{fancyhdr}
\fancyhf{}
%fancyfoot[c]{\color{structurecolor}\scriptsize\thepage} % 取消页脚
↪ 页码
\if@twoside
\fancyhead[CO]{\color{structurecolor}\cnormal\leftmark} %leftmark
↪ 章节名称, 奇数页页眉中部显示章节名称
```


E 偶数页
O 奇数页
L 页眉或页脚的左边部分
C 页眉或页脚的中间部分
R 页眉或页脚的右边部分
H 页眉
F 页脚

```
\fancyhead[OR]{\color{structurecolor}\cnormal\thepage} % 奇数页右侧  
↪ 页码  
\fancyhead[EL]{\color{structurecolor}\cnormal\thepage} % 偶数页左侧  
↪ 页码  
\fancyhead[CE]{\color{structurecolor}\cnormal\@title} % @title 教材  
↪ 名称, 偶数页页眉中部显示教材名称  
\else  
\fancyhead[R]{\color{structurecolor}\cnormal\rightmark}  
\fi
```


图表处理

0.23 图形

0.23.1 常规的图片插入方法

通常在 \LaTeX 中，通常使用的图片都是 jpg 和 png 格式。前者存储更小，但质量有损失；而后者可以做到无损压缩。但是这二者都不能做到放大不失真。为了达到放大不失真的效果，就需要采用矢量图。目前的矢量图的格式有：svg, eps, pdf。

在 \LaTeX 中标准插图方法为：

```
\begin{figure}[htbp]
\centering
\includegraphics[sc]
  ↪ ale=0.5]{filena}
  ↪ me.png}
\caption{name of
  ↪ figure}
\end{figure}
```



Figure 3: name of figure

`\includegraphics` 命令支持的图片格式为 jpg, png, eps, pdf。不支持 svg 格式的矢量图。如果要插入 svg 格式的矢量图，则需要采用 svg 宏包。使用方法如下：

```
\usepackage{svg}
\begin{figure}
\centering
\includesvg{filename}
  ↪ .svg}
\caption{name of svg}
\end{figure}
```



Figure 4: name of svg

如果在使用 svg 格式的矢量图时，出现字体和大小不正常现实的时候，采用 `\usepackage[inkscapelatex=false]{svg}`。

插入图片的时候，可以给 `\includegraphics` 命令加上 option 选项，可以实现比例和长宽的自定义。

`\caption` 命令在 `\includegraphics` 的前后决定了图题在图片的前后位置。

特别的对 eps 格式的图片，由三种方法进行调用。

第一种，采用 psfig 宏包。

```
\usepackage{psfig}
\psfig{figure= 图片文件名,height= 高度} \\
\psfig{figure= 图片文件名,width= 宽度}
```

第二种，采用 epsfig 宏包。

```

\begin{figure}
\centering
\includegraphics[width=
  ↪ h=5cm]{filename.p}
  ↪ ng}
\caption{name of
  ↪ figure}
\end{figure}

```



Figure 5: name of figure

```

\usepackage{epsfig}
\epsfig{figure= 图片文件名,height= 高度} \\ % 或者
\epsfig{figure= 图片文件名,width= 宽度}

```

第三种，采用 epsf 宏包。

```

\usepackage{epsf}
\epsfxsize= 宽度\epsffile{图片文件名} \\ % 或者
\epsfysize= 高度\epsffile{图片文件名}

```

其中的“高度”和“宽度”是指希望图片打印的高度和宽度，必须给出单位，可用厘米 (cm) 或英寸 (in)。

0.23.2 调用法插入图片

在 \LaTeX 文章排版过程中，会遇到一种情形：希望引用文献中的图片，但是无法找到文献中的图片原文件，采用截图的方式可能会使图片的清晰程度下降。可以利用 `includegraphics` 的不同参数选项来实现。举例如下：

```

\usepackage{graphicx}

\begin{figure}
\centering
\includegraphics[page=xx,viewport=xx yy zz
  ↪ nn,clip,scale=rate]{./filename.pdf}
\end{figure}

```

`includegraphics` 的选项中，`page` 表示页码，`viewport` 表示希望引用的图的位置，用 `xx yy` 表示图片的左下角的坐标值，`zz nn` 表示图片的右上角的坐标值，`scale` 用来设置缩放比例。在 \LaTeX 中，默认纸张的左下角为坐标原点。`filename.pdf` 即含有希望引用图片的原文件。

0.23.3 图片位置的排版

1. 浮动

在 \LaTeX 中，图片的位置通常会遵循浮动的原则。在 `figure` 环境添加浮动选项可以控制图片的浮动位置。

`figure` 的参数中，`h` 表示当前位置，`t` 表示顶部，`b` 表示底部，`p` 表示浮动。通常遵循 `h-t-b` 的顺序。单用 `h` 参数是没有效果，常用的选项 `ht`，`htbp`。如果强制需要图片在某个位置不变，可以采用 `float` 宏包，提供的 `[H]` 选项。但出现的空白页情况，则需要手动调节。

2. 环绕

对一些特殊的期刊文章，需要在文章的开始地方有问题环绕图片的情形。此时需要 `wrapfig` 宏包提供的 `wrapfigure` 环境来实现。`wrapfigure` 的参数：`[number]{r}{0.5\textwidth}` 行数，位置，宽度

```

\usepackage{wrapfig}
\begin{wrapfigure}[number]{r}{0.5\textwidth}
  \centering
  \includegraphics[scale=0.8]{filename}
  \caption{title of fig}

```

3. 子图

在文章排版过程中遇到需要多图并列的效果，需要由 subfigure 宏包提供的子图环境来实现。

```

\usepackage{graphicx}
\usepackage{folat}
\usepackage{subfigure}
\begin{document}
  \begin{figure}[H]
    \centering % 图片全局居中
    \subfigure[name1]{
      \label{Fig.sub.1}
      \includegraphics[width=0.45\textwidth]{filename1}}
    \subfigure[name2]{
      \label{Fig.sub.2}
      \includegraphics[width=0.45\textwidth]{filename2}}
    \caption{Main name}
  \end{figure}
\end{document}

```

4. (多行) 并列

学术文献中经常会让多张结果照片并列显示，有时可能还会出现多行并列显示。这个功能可以通过 minipage 环境来实现。

```

\begin{figure}[htbp]
  \begin{minipage}{0.45\linewidth}
    \centering
    \includegraphics[scale=0.15]{filename1.png}
  \end{minipage}
  \begin{minipage}[c]{0.45\linewidth}
    \includegraphics[scale=0.15]{filename2.png}
  \end{minipage}
  \caption{name of figure}
\end{figure}

```

5. 图题设置

L^AT_EX 默认的图题中含有冒号，根据需要将冒号去掉或改成点。

```

\usepackage{caption}
\captionsetup[figure]{label=period} %period 表示图 xx 和标题之间有
↪ 符号“点”。
\captionsetup[table]{label=space} %space 表示图 xx 和标题之间为空格。
↪

```

在排版过程遇到图片或图题距离上下的距离或图片和图题之间的距离需要调整的情形，可以采用一下代码。

```

\vspace{-1cm} % 调整图片与上文的垂直距离
\setlength{\abovecaptionskip}{-0.5cm} % 调整图片标题与图距的距离
\setlength{\belowcaptionskip}{-0.5cm} % 调整图片标题与下文之间距离

```

0.23.4 图片引用

在文章中对图片进行交叉引用方法如下：

```
\usepackage{graphicx}
\usepackage{float}
\usepackage{subfigure}
\usepackage{caption}
\begin{document}
  \begin{figure}[H]
    \centering % 图片全局居中
    \subfigure[name1]{
      \label{Fig.sub.1}
      \includegraphics[width=0.45\textwidth]{filename1}}
    \subfigure[name2]{
      \label{Fig.sub.2}
      \includegraphics[width=0.45\textwidth]{filename2}}
    \caption{Main name}
    \label{Fig.main}
  \end{figure}
```

0.23.5 图形编号

LaTeX 在处理图形编号时，会自动根据 `\begin{figure}` 和 `\end{figure}` 对图形的编号进行自动累加。在实际使用过程中，会经常遇到一些好几个图公用一个图题的情况，如果将这些图都放置在一个环境中，有些图可能会因为纸面大小的关系而无法显示，但是如果分开放置到两个或三个图形环境中，这些图题会自动增加。此时需要使用 `\ContinuedFloat` 命令，即可让图的编号不自动增加，而是和上一个环境链接起来，当一个图形环境来使用。在液压传动教材中的多图排版中采用如下的方式让图形编号不间断。

```
\begin{figure}[htbp]
  content...
\end{figure}
\begin{figure}[htbp]
  \ContinuedFloat % 链接上下两个图形环境，使其编号不自动增加
\end{figure}
```

一般的图形标题都比较短，所以在显示的过程中不会出现比较一些换行的问题。但是对于一些比较长的图题，而且图处于文字环绕的环境中，图题的换行问题就显得比较突出和重要了。图题的换行问题可以利用宏包“caption”配以相应的参数 `hang` 即可解决。caption 产生的图题中的图号和文字之间产生一个冒号，通过 `\captionsetup` 的参数 `labelsep=space` 即可去除这个冒号。

```
→ \requirepackage[font=small,labelfont={bf,color=sturcturcolor},hang]{caption}
→ %hang 解决了图题过长时的自动换行问题
\captionsetup[table]{skip=3pt}
\captionsetup[figure]{skip=3pt}
\captionsetup[labelformat=default,labelsep=space] % 去除图片编号中
→ 的冒号
```

去除图表 caption 中的冒号。

```

\usepackage{caption}
\captionsetup[table]{labelsep=space} % 表
\captionsetup[figure]{labelsep=space} % 图

\captionsetup[labelformat=default,labelsep=space] % 去除冒号

```

0.23.6 图形的浮动和环绕

通常我们在 L^AT_EX 插入图片时，要为图片单独设置一个环境，即 `\begin{figure}` 和 `\end{figure}`。

```

\begin{figure}
  \includegraphics[keyvals]{imagefile}
\end{figure}

```

在这个环境中可以设置输入图片的类型，`\includegraphics` 命令来实现对输入图片的操作。常用的一些基本参数如下：

<code>height</code>	图形的高度（可为任何 TEX 度量单位）。
<code>totalheight</code>	图形的全部高度，可为任何 TEX 度量单位（6/95 增加）。
<code>width</code>	图形的宽度（可为任何 TEX 度量单位）。
<code>scale</code>	图形的缩放因子，设定 <code>scale=2</code> 会使插入的图形的大小为其自然大小的两倍。
<code>angle</code>	设定旋转的角度，以度为单位，顺时针方向为正。

对于经常使用 L^AT_EX 进行论文排版的使用者来说，常见的图片排版问题莫过于多图的并列排版。对于实现多图排版，L^AT_EX 中要多种方式可以实现。当图片的版幅较大，无法两图并排放在一起，如果缩放不影响显示效果的情况下，可以连用 `\includegraphics` 来实现，并列排放的效果。另外也可以利用 `\minipage` 环境来实现多图的并列。

```

\begin{figure}[htbp] % htpb 用来浮动，h-here, t-top, b-bottom
  \centering
  \includegraphics[scale=0.9]{fig0203.pdf}
  \includegraphics[scale=0.9]{fig0204.pdf}
\end{figure}

\begin{minipage}[t]{0.48\textwidth} % 设置图形宽度
  \centering
  \includegraphics{fig0124.pdf}
\end{minipage}
\begin{minipage}[t]{0.48\textwidth}
  \centering
  \includegraphics{fig0125.pdf}
\end{minipage}

```

在对图片进行排版的过程中，可能会遇到图片较小或图片宽度较小、长度方向较大的情况，如果此时对图片进行单独占行进行排版，会留下大量的空白，影响整体的排版的美观性。因此，在排版过程中对图片进行环绕处理。在对图片的环绕处理时利用的 `\begin{wrapfigure}` 和 `\end{wrapfigure}` 环境来实现。`wrapfigure` 的参数主要分为：[行数]、{位置}、{宽度}。

```

\begin{wrapfigure}[num]{r}{8cm} % num 表示要环绕的行数，r 和 l 分别表
  ↪ 示从右侧和左侧环绕；8cm 为环绕留给图形的宽度
  \centering

```



```

\ifOpenSource
\includegraphics[width=5cm]{logo.png}
\else
\includegraphics{fig0201.pdf}
\fi
\caption{能量转换示意}
\label{fig:fig0201}
\end{wrapfigure}

```

对于一些占幅面较大的图片，可能需要横向排布才能显示较全的信息，此时有两种方式来实现图片的旋转功能。`\includegraphics[angle=xxx]{fig.png}` 或利用 `\begin{sidewaysfigure}` 和 `\end{sidewaysfigure}` 来实现。

```

\begin{figure}[htbp]
\centering
\includegraphics[angle=xxx]{imagefile} % 图片旋转，可自定义旋转
↪ 角度
\caption{text}
\end{figure}

\begin{sidewaysfigure} % 图片横向排版环境
\centering
\includegraphics[width=4in]{graphic.eps}
\caption{Sidewaysfigure Figure}
\end{sidewaysfigure}

```

通常情况下，图片和文字是按照一定顺序来进行排放的。但是有些期刊的排版会要求作者将图片和文字分开排版，即文字在前，图片和表格在后的排版方式。在 \LaTeX 中实现这种文字在前、图片在后的排版方式，仅需引入一个 `endfloat` 宏包即可实现。

```

\usepackage[noheads,nolists,nomarkers]{endfloat}
\renewcommand{\efloatseparator}{\mbox{}}
\end{}

\section{表}

\chapter{公式}

\section{常用公式录入}

```

\LaTeX 中数学公式的排版，大部分功能都由 `amsmath` 宏包提供。`amsmath`、
 ↪ `amssymb` 宏包提供了大量的数学符号；`amsthm` 宏包提供了扩展的 \LaTeX 定理
 ↪ 证明格式。

```

\subsection{行内公式}

```

行内公式，顾名思义，公式在文字的行内，属于文字公式混排。此类公式没有公式编号，通常由一对 ```\$"` 包裹，例如：


```
\begin{example}
勾股定理可表示为: $a^2+b^2=c^2$ 。
\end{example}
```

```
\subsection{行间公式}
```

行间公式处于文字段落之外，可以采用 `equation` 环境书写，此时行间公式默认有公式编号。采用 `equation*` 环境可以提供不带编号的行间公式。利

用 `\textbackslash` 命令来进行引用，或利用 `amsmath` 宏包提供的 `\eqref`（会默认对引用添加括号）命令来进行引用。如果需要自定义公式编号，可以采用 `\textbackslash` 命令进行自定义；也可以采用 `\textbackslash` 命令对公式进行自定义编号；或者采用 `\textbackslash` `notag` 取消公式编号。

```
\begin{example}
\begin{equation}
a^2 + b^2 = c^2 \label{Pythagoras}
\end{equation}
公式\eqref{Pythagoras}被称为勾股定理。
\end{example}
```

如果需要不提供编号的行间公式，可以采用 `\textbackslash` 和 `\textbackslash` 提供的行间公式环境或者利用 `displaymath` 环境提供的行间公式。

```
\begin{example}
\[\ a^2 + b^2 = c^2 \]

\begin{displaymath}
a^2 + b^2 = c^2
\end{displaymath}

\begin{equation*}
a^2 + b^2 = c^2
\end{equation*}
\end{example}
```

```
\subsection{行间多行公式}
```

在数学证明方面，可能会出现很长的、需要折行的公式，通过 `amsmath` 宏包提供的 `multline` 环境来完成。

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

在证明过程中可能会用到多个公式联立，并在等号处对其。此时，可以采用 `align` 环境。

```

\begin{example}
  \begin{align}
    x + y &= 5 \\
    3x + y &= 8
  \end{align}
\end{example}

```

如果不需要多行公式对其时，则可以采用 `gather` 环境完成。

```

\begin{example}
  \begin{gather}
    x + y = 5 \\
    2x + y = 8
  \end{gather}
\end{example}

```

`multline`、`gather` 和 `align` 环境都会自动生成公式符号，如果不需要公式编号的时候，在 `gather` 和 `align` 后面添加 “*”，即 `multline*`、`gather*` 和 `align*`。需要注意的是，`align` 和 `gather` 环境会给每一个公式添加编号；如果只需要一个公式编号，则需要将 `align` 或 `gather` 嵌套如 `equation` 环境中即可。

`\subsection{单侧括号公式块}`

```

\begin{example}
  \begin{equation}
    \left. \begin{aligned}
      x + y &> 10 \\
      x - y &> 2
    \end{aligned} \right\} \Rightarrow x^2 - y^2 > 20
  \end{equation}
\end{example}

```

对一个分段函数来说

```

\begin{example}
  \[ |x| = \left\{ \begin{array}{rl}
    -x & \text{if } x < 0, \\
    0 & \text{if } x = 0, \\
    x & \text{if } x > 0.
  \end{array} \right.
\end{example}

```

也可以采用 `amsmath` 宏包提供的 `case` 环境来实现。

```

\begin{example}
  \[ |x| = \begin{cases}
    -x & \text{if } x < 0, \\
  \end{cases}

```

```

0 & \text{if } x = 0, \\
x & \text{if } x > 0.
\end{cases}
\end{example}

```

```
\chapter{参考文献}
```

```
\section{参考文献引用}
```

`\LaTeX{}`中管理参考文献的方法有两类：一、针对文献数量较少时，可以直接采用
 → `thebibliography` 环境来管理参考文献；二、采用 `bib` 文档，调用相应的参考
 → 文献格式来进行管理文献。

```
\subsection{thebibliography}
```

`thebibliography` 方法相对简单，直接将参考文献的内容存放的 `document` 环境之
 → 内，清晰明了。当参考文献的数量相对较少，或对参考文献的格式没有特殊的要
 → 求时，相对来说是一种比较方便的引用方法。

```
\begin{minted}[breaklines,breakanywhere]{tex}
\documentclass{article}
```

```
\begin{document}
```

```
\section{Introduction}
```

这是一个测试案例，用来介绍引用参考文献的引用方法 `\cite{ref1}`。

→ `\\%ref1` 只是一个引用符号，可以是任何字符。

```
\begin{thebibliography}{99}%99 是指参考文献的最大数量，可以自行
→ 设定
```

```
\bibitem{ref1} xxxxxxxxxxxx % 和前面的 ref1 对应，后面添加
→ 参考文献的内容。
```

```
\bibitem{ref2} xxxxxxxx
```

```
\end{thebibliography}
```

```
\end{document}
```

```
\end{}
```

```
\subsection{bib}
```

对参考文献的排版有特殊的格式要求、管理的便捷、大量的参考文献条目、希
 → 望`\TeX{}`文档整体有条理时，可以采用 `bib` 文件的方式进行参考文献管理。

```
\begin{minted}{tex}
```

```
@article{name1,
```

```
title = {标题},
```

```
author = {作者, 多个作者用 and 连接},
```

```
year = {年份},
```

```

journal = {期刊名称},
volume = {卷},
pages = {页码},
abstract = {摘要},
}

@mastersthesis{name2, %masterthesis 表示学位论文。
type = {硕士},
title = {标题},
author = {作者},
year = {年份},
abstract = {摘要},
school = {武汉科技大学},
}

```

在 L^AT_EX 文档中，需要引用文献的地方插入文献使用 \cite 命令。在文档结束之前的地方插入 bib 文件名称。

```

\documentclass{article}

\begin{document}

\section{Introduction}

xxxxxxxxxxxx

\bibliographystyle{style} %style 为希望展示的参考文献格式
\bibliography{reference} %reference 为 bib 文件的名称,
\end{document}

```

文献引用

0.24 西文参考文献

在 L^AT_EX 文档中添加参考文献，需要做好几个步骤：准备 bib 格式的参考文献，在 L^AT_EX 文档中调用相应的宏包，编译生成 pdf 文档。

bib 格式的参考文献必须具备的内容：author,title,journal,volume,year,number。举例如下：

```
@article{name1, %name1 为正文中需要引用的关键词，即 \cite{name1}。
  title = {文章标题},
  author = {作者，多个作者用 and 连接},
  journal = {期刊名},
  volume = {卷},
  number = {页码},
  pages={},
  year = {年份},
  publisher={出版社}
}
```

在 L^AT_EX 文档中，首先在导言区调用 cite 宏包，在正文区内设置需要显示的参考文献风格类型，设置参考文献 bib 文件的位置。L^AT_EX 默认提供的参考文献样式有 8 种：

```
\centering
plain, 按字母的顺序排列，比较次序为作者、年度和标题。
unsrt, 样式同 plain，只是按照引用的先后排序。
alpha, 用作者名首字母 + 年份后两位作标号，以字母顺序排序。
abbrv, 类似 plain，将月份全拼改为缩写，更显紧凑。
ieeetr, 国际电气电子工程师协会期刊样式。
acm, 美国计算机学会期刊样式。
siam, 美国工业和应用数学学会期刊样式。
apalike, 美国心理学学会期刊样式。
```

完整的引用案例如下：

```
\documentclass{article}
\usepackage{cite}

\begin{document}
  这是一个参考文献引用方法的举例\cite{key}。
  \bibliographystyle{unsrt} % 设置参考文献的显示风格。
  \bibliography{reference} % reference为存放参考文献的bib文件名，
  ↳ 该文件默认和tex主文档放在同一个目录下；如果发生了路径变化，
  ↳ 则需要提供相对路径。

\end{document}
```

参考文献的引用需要在导言区添加相应的宏包，在正文中需要引用参考文献的地方，添加相应的文献。添加单个或多个不连续的文献，方式如下：

```
\usepackage{cite}% 导言区添加
```

文献表明：

↪ 这个问题可以通过这种方式来解决\cite{bibtex1,bibtex3,bibtex5}。
↪ %正文区

输入结果如下：

文献表明：这个问题可以通过这种方式来解决 [1,3,5]。

添加多个连续的参考文献，方式如下：

```
\usepackage{cite} % 导言区
```

```
\usepackage[number,sort&compress]{natbib} % 导言区
```

文献表明：这个问题可以通过这种方式来解决\cite{key1,key2,key3}

输出结果如下：

文献表明：这个问题可以通过这种方式来解决 [1-3]。

0.25 中文参考文献

中文参考文献可以遵循 GB/T 7714-2015。使用 L^AT_EX 对中文学术论文进行排版时，可以按照 GB/T 7714-2015 来设置参考文献的格式。用法如下：

```
\documentclass{article}
```

```
\usepackage{ctex}%ctex 宏包中包含了 cite，可以不必单独调用宏包。
```

```
\usepackage{gbt7714}
```

```
\begin{document}
```

充足的睡眠有利用身体健康\cite{key1}

```
\bibliographystyle{gbt7714-numerical} %利用 GB/T
```

↪ 7714-2015宏包来设置参考文献的格式

```
\bibliography{ref} %ref 代表存放参考文献的 bib 文件名称。
```

```
\end{document}
```

0.26 格式

L^AT_EX 在文档最后编排的参考文献格式被成为 ‘style’，调用时采用 \bibliographystyl{style}，\bibliography{reference}。

常见的几种格式如下：

plain，按字母的顺序排列，比较次序为作者、年度和标题. unsrt，样式同 plain，只是按照引用的先后排序. alpha，用作者名首字母 + 年份后两位作标号，以字母顺序排序. abbrev，类似 plain，将月份全拼改为缩写，更显紧凑. ieetr，国际电气电子工程师协会期刊样式. acm，美国计算机学会期刊样式. siam，美国工业和应用数学学会期刊样式. apalike，美国心理学学会期刊样式。

0.27 插入引文

首先要在导言区调用宏包 `\usepackage{cite}`；在正文中需要引用文献的地方，输入 `\cite{bibtex1}`，编译之后正文引用文献的地方显示 [1]。在文中需要输入多个参考文献的地方输入 `\cite{bibtex1,bibtex3,bibtex5}`，其中每个引用文献之间要英文逗号隔开，输出结果为 [1,3,5]。如果要输入的参考文献序号是连续的，则需要在导言区调用新的宏包，`\usepackage[number,sort&compress]{natbib}`，正文中的参考文献 `\cite{bibtex1,bibtex2,bibtex3}`，输出结果为 [1-3]。

有些文档重要求参考文献的引用标志在正文中要体现在右上角，此时要在导言区自定义新命令，`\newcommand{\upcite}[1]{\textsuperscript{\textsuperscript{\cite{#1}}}}`，在正文需要参考文献的地方，用 `\upcite` 命令来插入参考文献，`\upcite{bibtex1}`。

在 \LaTeX 中使用中文参考文献，其格式需要参照 GB/T 7714-2015 来执行，但是在 \LaTeX 本身的支持中却没有该选项。因此，需要通过自定义功能来实现，在 github 上 <https://github.com/zepinglee/gbt7714-bibtex-style>。

0.28 引文文件

本文仅用于总结近期使用 \LaTeX 整理书籍的一些常用宏包的用法。

批注与修订

手动标记：changes, todonotes, annmark, soul + color；或采用自动对比 latexdiff 宏包。

0.29 changes

changes 宏包用来在 L^AT_EX 中实现增加、删除、替换、批注等修订功能。该宏包主要提供了 \added, \deleted, \replaced, \comment 四条修订命令。

在使用过程中需要在导言区调用 changes 宏包，并采用 \definechangesauthor 命令自定义作者信息以及修订过程中所要显示的颜色；不同的作者可以采用不同颜色进行显示。

```
\documentclass{article}
\usepackage{changes}
\definechangesauthor[name={Per cusse}, color=orange]{per}
%\setremarkmarkup{(#2)}
\begin{document}
This is \added[id=per,remark={we need this}]{new} text.
This is \added[remark={has to be in it}]{new} text.
This is \deleted[id=per,remark=obsolete]{unnecessary}text.
This is \replaced{nice}{bad} text.
\listofchanges
\end{document}
```

This is **new** text. This is **new** text. This is **unnecessary**text. This is **nicebad** text.

List of changes

Added (per): new	1	[per 1] we need this
Added: new	1	[1] has to be in it
Deleted (per): unnecessary	1	[per 2] obsolete
Replaced: nice	1	

0.30 todonotes

```
\documentclass{article}
\usepackage{lipsum} % Dummytext
\usepackage{xargs} % Use more than one optional parameter in a new
↪ commands
```

```

\usepackage[pdftex,dvipsnames]{xcolor} % Coloured text etc.
%
\usepackage[colorinlistoftodos,prependcaption,textsize=tiny]{todonotes}
\newcommand{\unsure}[2][1]{\todo[linecolor=red,backgroundcolor=red!25,
bordercolor=red,#1]{#2}}
\newcommand{\change}[2][1]{\todo[linecolor=blue,backgroundcolor=blue!25,
bordercolor=blue,#1]{#2}}
\newcommand{\info}[2][1]{\todo[linecolor=OliveGreen,backgroundcolor=OliveGreen!25,
bordercolor=OliveGreen,#1]{#2}}
\newcommand{\improvement}[2][1]{\todo[linecolor=Plum,backgroundcolor=Plum!25,
bordercolor=Plum,#1]{#2}}
\newcommand{\thiswillnotshow}[2][1]{\todo[disable,#1]{#2}}

\begin{document}
\pagestyle{empty}
\listoftodos[Notes] % Show todo list or not
\todo[inline]{The original todo note without changed colours.\newline Here's another line.}
\lipsum[11]\unsure{Is this correct?}\unsure{I'm unsure about also!}
\lipsum[11]\change{Change this!}
\lipsum[11]\info{This can help me in chapter seven!}
\lipsum[11]\improvement{This really needs to be improved!\newline\newline What was I thinking?!!}
\lipsum[11]
\thiswillnotshow{This is hidden since option `disable' is chosen!}
\improvement[inline]{The following section needs to be rewritten!}
\lipsum[11]
\end{document}

```

0.31 annmark

```

\documentclass[11pt, a4paper, UTF8]{article}
\usepackage{bm}
\usepackage{ctex}
\usepackage{tikz}
\usepackage{marginnote}
\newcommand{\annmark}[1]{%
\textcolor{red}{\bm\langle\color{blue}{#1}\bm\rangle}}%
}%
\newcommand{\ann}[1]{%
\begin{tikzpicture}[remember picture, baseline=-0.75ex]
\node[coordinate] (inText) {};%
\end{tikzpicture}%
\marginpar{%
\renewcommand{\baselinestretch}{1.0}%
\begin{tikzpicture}[remember picture]
\definecolor{orange}{rgb}{1,0.5,0}%

```

```

\draw node[fill=red!20,rounded corners,text width=\marginparwidth]
↪ (inNote){\footnotesize
; %
^^I\end{tikzpicture}%
}%
\begin{tikzpicture}[remember picture, overlay]%
\draw[draw = orange, thick]
([yshift=-0.2cm] inText)
-| ([xshift=-0.2cm] inNote.west)
-| (inNote.west);%
\end{tikzpicture}%
}%
\setlength{\marginparwidth}{2.5cm}
\renewcommand{\baselinestretch}{1.3}
\renewcommand{\thefootnote}{\alph{footnote}}
% 可以改变为希腊字母，默认是阿拉伯数字
% \usepackage{verbatim}
\begin{document}
    这是一段测试用的\annmark{语言}\ann{xxxx},
    ↪ 这里并没有什么特殊之处\footnote{xxxx},
    ↪ 仅仅用来做{示例}\marginnote{xxxx}。
% \begin{comment}
% 此处是注释啊
% \end{comment}
% 一样是静悄悄的罢。
\end{document}

```

0.32 latexdiff

使用 latexdiff 工具对修改前后的 tex 文档进行对比，并自动生成具有差异的 tex 文件，编译之后即可生成对应的修改版文档。latexdiff 工具依赖 perl 语言，需要提供在计算上安装 perl 语言安装包。

```
latexdiff -so draft.tex revised.tex > diff.tex
```