

README

xyx

2024 年 8 月 22 日

目录

0	文章整体结构	1
1	文档类	3
2	常用功能	4
2.1	特殊命令 ~ 导言区	4
2.2	正文区	6
2.2.1	顶级结构, 目录与标题	6
2.2.2	行文操作	6
2.2.3	插入元素	10
3	注意事项	19
3.1	编译一篇文章	19
3.2	处理日语	19
附录		
附录 A	举例环境外的行间公式与正文的间距示例	20
附录 B	列表环境的层级与标识符示例	21
附录 C	Excel 表格导出示例	22

0 文章整体结构

```

~~~~~(魔法注释)~~~~~

~~~~~(特殊命令)~~~~~

\documentclass[全局选项]{文档类} %源代码中的百分号代表此行往后都是注释

~~~~~(导言区)~~~~~

~~~~~(导言区)~~~~~

\begin{document} %此为文章内容开始之标志
\maketitle %输出标题
\tableofcontents %输出目录
\setcounter{page}{0} 将此页（目录页）的页码设为 0
\clearpage %新开一页

~~~~~(正文)~~~~~

\section[在目录里显示的节名]{节名}

~~~~~(正文)~~~~~

\subsection[在目录里显示的小节名]{小节名}

~~~~~(正文)~~~~~

\subsubsection[在目录里显示的子小节名]{子小节名}

~~~~~(正文)~~~~~

\clearpage
\bibliographystyle{unsrt} %标明参考文献的样式
\bibliography{reference.bib} %引入参考文献，大括号内为文献表位置

\clearpage %新起一页
\appendixpage %生成附录页
\noappendicestocpagenum %命令目录中的“附录”字样后方不标页码
\addappheadtotoc %将“附录”字样加入目录中
\appendix %规定此后为附录环境
\section{附录标题}

~~~~~(附录正文)~~~~~

(此后结构可同正文，但习惯上没有subsection和subsubsection)

\end{document} %此为文章（文件）内容结束之标志

```

此后不再附注源代码，可以在本地编译之后，用 [TeXstudio](#) 等编辑器的源代码 ↔ pdf 双向跳转功能对各个效果进行查看。

魔法注释

魔法注释的作用一般是指定编译器. 如果用 \LaTeX 专用编辑器, 或是编译环境较为固定则是没有什么太大必要学. 但是一些人用 Emacs, Vim 等编辑器则需要进行设定. 本文未使用魔法注释.

特殊命令

特殊命令目前仅需要一个加密命令. 但这个加密命令对复制进行的限制, 目前发现只对网页浏览器等自带的 pdf 浏览器有效; 若使用Skim, SumatraPDF等查看则不受限制. 本文仅使用了加密这一条特殊命令. 加密命令的详细使用方法见[此处](#)

1 文档类

\LaTeX , $\text{pdf}\text{\LaTeX}$, $\text{Xe}\text{\LaTeX}$, $\text{Lua}\text{\LaTeX}$, $\text{up}\text{\LaTeX}$ 有对UTF-8字符集的支持, 但默认仅显示ASCII字符集, 需要额外的调用才会生成UTF-8字符集. 其中 $\text{Xe}\text{\LaTeX}$ 与 $\text{Lua}\text{\LaTeX}$ 支持从编译者的电脑字体库中调用字体, 而别的均依赖 \LaTeX 内的字库. 而中文目前已经几乎完全基于UTF-8, 这意味着我们只需要调动相应的字符集就可以让 \LaTeX 输出含中文的.pdf, 也就是 ctex 宏包. 它需要在导言区被调用.

ctex 在不同的编译引擎中会自动选择不同方式实现中文: 它内部包含了不同的宏包, 详见其[官方文档](#). 换言之, 若单独调用其启用中文的核心组件, 也可以实现中文. 但是目录页等字样会保持英文: 对它们的翻译基于 ctex . 东亚文字的数字化开发一般都是一同进行, 所以也可以通过类似的方法轻量化实现日文, 此处不再赘述.

但是这个.tex文件的导言区内并没有调用 ctex 或其他调用中文的宏包, 是因为使用了 ctex 文档类之 ctexart . 另外还有3种 ctex 文档类: ctexrep , ctexbook , ctexbeamer . 它们分别对应了英语格式的 article , report , book , beamer 并有对全角字符的格式优化. ctex 文档类现已集成于国际各主流 \LaTeX 发行版基础组件中.

文档类文件一般包含在基础组件内. 若要向学术期刊投稿, 需要按照他们模板制作文档, 此时他们会提供 \LaTeX 模板文件, 类比定义了 ctexart 的文件. 若在导言区填入了一个非系统自带的, 则这个格式定义文件, .cls文件, 须和文档在同一个文件夹内. 一般学术期刊提供的模板不仅仅是.cls文件, 是一个还包括了更多的样式定义文件的压缩包, 故而推荐将其解压后直接在其文件夹内部进行文档编辑与编译.

2 常用功能

本文采用了适用于大部分化学类文章编写的环境. 从人类认知逻辑上, 我们可以把一个 \LaTeX 源代码分为以下几个区域¹:

- 导言区:
 - 顶级命令区
 - 样式调用区
 - 插件调用区
 - 文章元素规定区
- 文章区:
 - 正文区
 - 参考文献区
 - 附录区.

2.1 特殊命令 ~ 导言区

各具体设定代码详见源文件. 自上到下各代码分别代表着:

1. 顶级命令区:

- (a) 加密 pdf 的特殊命令: perm 后的数字标识了开放的权限, 4 为仅允许低质量打印: 点击打印按钮, 如果输出为一个新 pdf 会发现全部页面都被转化成了低分辨率图片; 2052 为仅允许普通打印; 此

处省略了 userpw 项, 即打开文档时的密码. 须注意: \LaTeX 对此加密命令的兼容性貌似还不好.

- (b) [纸张类型 + 生成标题页²(+ \LaTeX 编译引擎标识 + dvi2pdf 引擎标识³)] + {文档类}.

- 中文用 [a4paper, titlepage]{ctex...};
- 日文用 [a4paper, , titlepage, uplatex, dvipdfmx]{js...};
- 详细的通用文档类列表见[此处](#).

2. 样式调用区:

- (a) 调用物理单位样式的宏包 `siunitx`: 在公式环境中, 可以生成标准样式的带单位物理量;
- (b) 调用化学 (方程) 式的宏包 `mhchem`: 调用它时强制要求必须标明宏包版本号, 否则报错;
- (c) 调用数学公式样式的宏包 `amsmath`: 它提供了多种公式样式以及公式对齐样式;
- (d) 调用数学符号样式的宏包 `amssymb`;
- (e) 调用数学字体样式的宏包 `amsfonts`;
- (f) 调用记号样式的宏包 `metalogo`;
- (g) 调用附录样式的宏包 `appendix`:
 - 基于此宏包, 规定目录中出现“附录”字样, 默认值为不出现, 设为出现时则是出现“Appendices”.

¹大部分我自己取的名, 或者对官方译文有化用

²有则生成, 没有则报错

³此处仅能提示各组件对其优化, 而不是用来指定引擎; 指定引擎另须以魔法注释标明, 且须有编辑器支持; 一般仅在使用 `uplatex+dvipdfmx` 时须进行此标注, 否则会有宏包各种报错

- 基于此宏包, 规定附录第一页左上角出现“附录”字样, 默认值为不出现, 设为出现时则是出现“Appendices”。
- (h) 调用颜色宏包`xcolor`: 但其实 \LaTeX 本身也提供一些默认颜色, 但它可以指定 RGB 色号。
 - 定义全局背景色为RGB值为 (46, 46, 46) 的颜色⁴;
 - 定义全局文字颜色为RGB值为 (248, 248, 248) 的颜色⁵。
- (i) 调用行距调整宏包`setspace`: 基于此宏包, 可以部分调整行距;
- (j) 调用文本框生成宏包`framed`: 基于此宏包, 可在被选择的文本外围生成文本框;
- (k) 调用表格样式宏包`tabularx`
- (l) 调用表格背景色包`colortbl`: 基于此宏包, 可表示Excel中的背景色
- (m) 调用参考文献样式包`biblatex`
 - i. 设置`hyperref=true`: 令其意识到 `hyperref` 已启用, 须生成带超链接的引用标识;
 - ii. 设置`backend=biber`: 使之选定参考文献管理引擎为`biber`. 选用`biber`的原因是它在完全兼容`bibtex`的基础上提供了更多的文献出处类型以及自定义选项;
 - iii. 设置`url=true`: 有些`.bib`代码会附上超链接, 可以按照不同需求选择表示与否;
 - iv. 设置`doi=true`: 类似于超链接, 可选择表示DOI号
 - v. 设置`dateabbrev=true`: 出版日期中的月份是否简写, 类似于November简写为Nov.;
 - vi. 设置`backref=true`: 可选择在参考文献条目后表示此文献在哪几页被引用了, 并将此处的页码做成超链接;
 - vii. 设置`style=numeric-comp, maxnames=15`

3. 插件调用区:

- (a) 调用图片插入宏包`graphics`;
- (b) 调用子图插入宏包`subfigure`;
- (c) 调用浮动元素宏包`float`: 多数时候没什么用, 原因后述;
- (d) 调用脚注样式宏包`footmisc`: 脚注功能基于 \LaTeX 本身, 此宏包并非必要;
- (e) 调用超链接宏包`hyperref`: 它的作用不只是超链接, 也可以编辑生成的`.pdf`的属性; 它会“污染”一些别的宏包的命令, 故而其最好是作为最后一个调用的宏包, 易于覆盖别的宏包的命令; 易知仅需打印的文档完全不需要此宏包;
- (f) 对超链接宏包的设置项:
 - i. 在 pdf 阅读器识别到的书签里, 为各个书签前加上章节号;
 - ii. 对超链接文字进行染色;
 - iii. 规定文内跳转链接为蓝色;
 - iv. 规定 url 超链接为蓝色;
 - v. 规定文献引用标签为黑色;
 - vi. 规定`.pdf`属性中的文章作者为 `xyx`;
 - vii. 规定`.pdf`属性中的文章主题为 \LaTeX 简明手册;
 - viii. 规定`.pdf`属性中的关键词为 `X\LaTeX`, `p \LaTeX` , `ctex`, `marco`, `xeCJK`.

4. 文章元素规定区⁶

- (a) 规定文章标题为《README》;
- (b) 规定文章作者为 `xyx`;
- (c) 规定发行日期为`\today`⁷.

以上为导言区的大体设定, 除了基于对宏包的设定需要在其对应宏包之后, 而其余导言区代码则无顺序以及缩进需求。

⁴这是我的编辑界面背景色, 不然左边黑边白太扎眼, 文件正式发布前删掉再跑通就行了

⁵底色偏黑, 那我就随便选了一个偏白的颜色当文字颜色

⁶此处还可以有 `institute` 等其它项, 但暂时不需要

⁷此为自动获取今天日期的命令, 但是不能像 `python` 那样可以 `print(1+1)` 得到输出 2; 可在正文中使用

2.2 正文区

2.2.1 顶级结构, 目录与标题

在`\begin{document}`后, 即为文章内容的开始, 标题与目录自然也算是文章的内容. 而整个文章 (代码) 的结束标志则是`\end{document}`

紧随`\begin{document}`, 我们能看到如下的代码:

```
\maketitle
\tableofcontents
\setcounter{page}{0}
\clearpage
```

它们的含义分别是:

1. 生成标题

- 若在导言区未用`\title{...}`设定标题, 则会报错;
- 若未在[此处](#)标明`titlepage`选项, 则仅会在对应位置的顶部生成标题, 故极度推荐使用`titlepage`选项, 你好我好大家好.

2. 生成目录;

3. 强行设定此页的页码为0;

4. 此页余下留白, 即之后的内容新起一页.

2.2.2 行文操作

在写作时, 时常需要进行切换自然段, 顶格另起一行, 强行空行等操作. 以下为其对应样式

• 切换自然段

在原生文档类 (`article`等) 中, 也可以使用`\paragraph{自然段名}`来新起一段. 此文件基于`ctexart`生成, 故无法演示. 两种方式英语文章中可以使用, 但是中文日文则不推荐有自然段名;

`\paragraph{自然段名}`会拉大和上一段的间距, 哪怕不写自然段名. 如果想特意拉大段间距, 则可以全部使用`\paragraph{}`;

`\par`没有自然段名, 而且貌似没有哪里人喜欢用自然段名, 故而这个足够;

以下为效果实例.

这是一段
这是另一段

自然段名 这是新一段

没有自然段名
这是又一段

• 顶格另起一行

以下为效果实例.

这是一段
顶格新起一行
这是另一段
顶格再新起一行
顶格又新起一行

- 插入空行

\LaTeX 貌似原则上不允许在没有内容的一行文字下再空一行, 也就是禁制连续空行. 但是在 \ctex 文档类下是可以的. 最保险的办法是在新开的一行里加一个空白元素, 然后再空行, 便是符合规则的;

以下为效果实例.

这是一段
顶格新起一行

空两行然后顶格又新起一行
这是另一段

- 特殊符号

一些特殊符号需要在公式环境内使用. 考虑到在文字间的使用情景, 那么我推荐调用行内模式的公式环境, 或者是抄写环境

在全局命令中被广泛使用的 $\%$, $\{$, $\}$, \backslash 四种符号可以通过在代码前面加一个 \backslash 来解决 (\backslash 需要公式环境).

- 新起一页

类似于插入空行, \LaTeX 不允许直接插入空白页, 但可以在一个页面内插入空白元素, 然后再新起一页;

插入的空白页也会带有页眉页脚装饰. 若想清除 (变成纯白页), 则可以使用 $\backslash\text{\thispagestyle}\{\text{\empty}\}$ 命令清除其所在页的所有装饰元素⁸;

强行改变某个页的页码可以用 $\backslash\text{\setcounter}\{\text{\page}\}\{\text{整数}\}$. 同理, 它也可以改变一些其他元素的序号 (section, figure 等). 当然, 此命令也会影响后续的页码自动增加;

以下为效果实例.

⁸页面计数不会停

2.2.3 插入元素

以下为一些常用示例. 更加完整的宏包功能与调用代码可在[CTAN](#)上搜索包名并查看其官方文档.

- 插入图片:

1回目	2回目	3回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

图 1: 这是题注

- 此处“图片设定”为`width=0.5\textwidth`, 即一半栏宽; 也可用具体的毫米, 厘米, 英尺等单位: `width=2 cm`; 可指定的数字可以精确到不低于 20 位小数⁹;
- `[H]` 表示作为浮动图形¹⁰强行摆放于此处 (Here), 也可使用 `[p]`, `[t]`, `[b]`, `[h]`, 它们分别浮动位置的 Page of its own(自成一页并摆放在下一页), 表示固定位置的 Here(此处), 固定位置的 Top(此页顶端), 表示固定位置的 Bottom(此页底部). 不标注则摆放位置默认为 `[t]`. 也可使用填入多个字母来表示你可接受的摆放方式. 须注意的是, 尽管一般不使用 `[p]`, 但它会令此后的内容强制出现在此图片页的后一页: 下一页的正文有且仅有此图片. 若图片以当前设定摆放失败则会自动切换到 `[p]` 模式;
- 另有一些其它的图片设定, 例如 `angle=30`, 表示图片顺时针旋转 30°. 此处支持填入大于 360 的数字;
- \LaTeX 支持导入 `.png`, `.pdf`, `.eps` 等格式的图片. 尽管一般可不备注文件后缀名, 但若存在同一文件 + 多种后缀名的情况, 则会在可导入的文件类型中以拓展名字母顺序 (A→Z) 决定导入优先顺序¹¹. 故而此处推荐在代码中标明后缀名, 以避免潜在的报错;
- 若即将被导入的文件与文档在同一个文件夹内, 则不需额外注明路径; 相应地, 注明路径便能导入电脑里任意位置的图片.

- 子图

效果如下图所示

⁹至少 20 位还没报错, 多了我懒得试, 千分位再往后也没有意义, 因为人看不出来

¹⁰没有 `[T]`, `[B]`, `[P]`

¹¹不知为何, `png` 的优先级貌似远高于 `jpg`

1回目 k/T	2回目 k/T	3回目 k/T	1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05	6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208	0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973	0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353	0.000179538	0.000130363	0.000131353

(a) 子图 1

1回目 k/T	2回目 k/T	3回目 k/T	1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05	6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208	0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973	0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353	0.000179538	0.000130363	0.000131353

(b) 子图 2

1回目 k/T	2回目 k/T	3回目 k/T	1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05	6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208	0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973	0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353	0.000179538	0.000130363	0.000131353

(c) 子图 3

1回目 k/T	2回目 k/T	3回目 k/T	1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05	6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208	0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973	0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353	0.000179538	0.000130363	0.000131353

(d) 子图 4

图 2: 子图示例

子图的摆放默认为先横后竖, 且会自动换行, 亦可手动换行.

注意到:

- `\hfill`命令表示令其前后两元素 (即图片) 在文字栏内水平等距摆放, 且自动取到最大间距, 亦可在数个元素间加入此命令. 其虽自动, 但是两元素的宽度和超过了`\textwidth`, 则将会失效, 两者将在`\centering`命令下变成页面居中并竖直摆放;
- 亦可用`\hspace{长度}`命令指定两图间距;
- 同一横排的各元素宽度和不应超过 $0.9910443115 \times \text{width}$, 否则最右端的元素会被摆放到下一行. 此处指定了每行两图间距均为 $0.03 \times \text{width}$; 图2(a)和图2(b)宽度均为 $0.48052215576 \times \text{width}$, 而图2(c)和图2(d)宽度均为 $0.48052215577 \times \text{width}$;
- 若在两`\subfigure[...]{...}`间插入`\hfill \vrule \hfill`命令, 则可令两图中央出现一条竖分隔线;
- 同理可得: 还有`\vfill`和`\vspace{长度}`命令 (V for Vertical, h for Horizontal), 只是一般不用; 亦有`\hrule`.

• 公式

L^AT_EX 内自带 3 种公式环境¹²如下:

- 行内模式 (不支持公式内换行), 其效果为如红色文字所示: $E = mc^2$
- 行间模式 (不支持公式内换行), 其效果为如蓝色文字所示:

$$E = mc^2$$

¹²此处是举例环境内, 故而行间公式 \longleftrightarrow 文字间距会很怪, 也受环境影响不会严格居中, 平常不这样

- 带编号的行间模式 (不支持公式内换行), 其效果为如绿色文字所示:

$$E = mc^2 \quad (1)$$

而`amsmath`引入了更多的样式环境如下:

1. `equation*`¹³: 等效于行间模式;
2. `align`: 可标注行内对齐点的`equation`. 其效果如下灰字所示:

$$\begin{array}{rcl} 1 + 1 & = & 2 \\ 10 + 10 & = & 20 \end{array} \qquad \begin{array}{rcl} 2 + 2 & = & 4 \\ 20 + 20 & = & 40 \end{array} \qquad (2)$$

它使得不同行的第奇数个 `&` 后的那个字符进行中线对齐, 而第偶数个 `&` 则表示拉开两侧元素的距离; 若不填入任何 `&` 则默认将最长的公式居中, 再令其余向其右对齐.

3. `align*`: 不带编号的`align`.

据前, 易知:

- 公式中的字符间距不受源代码的空格影响, 而是自动识别各个字符与命令并摆放
- 若在断行前插入命令可让此行被跳过计数

字符间距自然也可以手动调整, 共有如下 6 种可选:

1. `\negthinspace` 负空格
2. `\noindent` 无空格
3. `\medspace` 窄空格
4. `\normalindent` 中等空格
5. `\textwidth` 词间空格
6. `\quad` 四倍空格
7. `\qquad` 八倍空格

其中的`\!`, `\,`, `\:`, `_`, `\quad`, `\qquad`即为字符间距设置命令.

注意到:

- 此处的`_`也被识别为一个命令;
- 可以用`\text{...}`命令在公式中插入”文本”;
- 命令间可以紧密相连.

• 抄写

抄写环境分为行内抄写与行间抄写, 类比与行内公示与行间公式, 此处不再赘述. 每种又分为表示空格与否两种

行内抄写 (不表示空格) 效果下: `$E=mc^2$` %这是注释

¹³很多时候可用加星号的方式令一个本被计数的环境 (section, subsection 等) 被跳过并不在目录中显示

行内抄写 (表示空格) 效果如下: $E=mc^2$ 这是注释

行间抄写 (不表示空格) 效果如下:

```
\begin{equation} %这是注释
E=mc^2
\end{equation} %这是注释
```

行间抄写 (表示空格) 效果如下:

```
\begin{equation} %这是注释
E=mc^2
\end{equation} %这是注释
```

注意到:

- 两个 | 之间的内容将忽略行首行尾缩进的前提下, 忽略区域内一切指令的前提下, 将代码作为文本绝对还原地转写;
- 英文与半角符号 (ASCII 字符) 转写得到 **Courier New** 字体, 即老式打印机字体;
- 汉字转写得到楷体;
- 转写将无法被附加颜色;
- 每一个 符号表示一个半角空格, 而日语中常用的全角空格将会被忽视.

• 列表

列表环境分为带序号与否 2 种: `itemize`(无序) 和 `enumerate`(有序).

两种列表环境可相互混合嵌套: 每一种列表环境最大可以达到 4 层¹⁴, 因为只做了 4 种不同层级的标识符.

每一个例子还可以分为 `item`, `subitem`, `subsubitem` 三层, 故逻辑上最高能套到 24 层.

以下为不完全示例:

- 第一级
 - 第二级
 - 第三级
 - 1. 第四级
 - 第五级
 - 第六级
 - (a) 第七级
 - 第八级
 - 第九级

¹⁴4 层 `itemize` 加 4 层 `enumerate` 后, 就不能再向其中套任何新的这两种环境了. 实际上只允许套到总计第 6 层, 但是套到第 7 和第 8 也只是报一个非致命错误, 输出文件没问题

注意到:

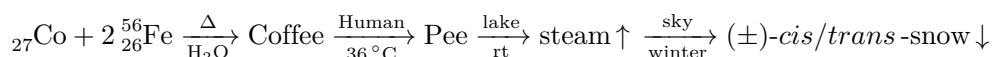
不可在没有`\item`的地方使用`\subitem`; 同理对`\subsubitem`成立;

仅`item`前有标识符;

在 \LaTeX 内部的缩进处理中, 无法做到视觉效果 24 层, 故而在重复嵌套举例环境中不推荐使用严格的`item`, `subitem`, `subsubitem`.

• 化学 (反应) 式

输出效果如下:



注意到:

- 化学 (反应) 式可以出现在公式环境内, 也可以出现在正文中, 因为它被别的包调用;
- 输入化学 (反应) 式代码时, 很多命令都跟空格有较强的联系, 故须格外关注;
- 输入化学 (反应) 式代码时, 在本该有空格的地方选择断行代码则可以不输入空格, 因为断行自带切断作用; 相应地, 乱断行也会造成不必要的麻烦.

• 表格

基于Excel的[宏包¹⁵](#), 可以将Excel的表格导入至 \LaTeX . 安装完之后点击顶端“加载项”按钮即可看到. 选定想要的区域然后点击[不带闪电标](#)的按钮即可读取此区域的Excel布局元素¹⁶与内容, 并生成代码. 就目前来看, 如果是把代码粘贴到主文件里, 则必然会无尽报错, 但是如果用`\input{文件名}`命令, 则丝般顺滑:

将生成的表格源代码粘贴进一个空白的`.tex`文件内, 然后放到主文件的同一个文件夹内, 再用

```
\begin{centering}
\input{文件}
\end{centering}
```

命令即可成功导入如图1所示的表格.

带闪电标的按钮说是将所有暂存的源代码输出到 \LaTeX , 但实际上这玩意是针对Office2010开发的, 目前仅能输出选定区域;

被`\input`的文件若标注路径则可实现从电脑的任意处导入代码; 相应地, 若被导入的文件和文档在同一文件夹下, 则不需注明路径.

表 1: 带背景颜色与文字颜色的表格示例

1 回目	2 回目	3 回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

¹⁵ 估计是不支持 WPS, 但是正经笔记本都应该是自带了永久 Office 套装

¹⁶ Excel 的框内文字左/右对齐, 外框线, 合并单元格, 各处颜色等

- 超链接

常用超链接主要分为以下四种:

1. 文内指定位置跳转
2. 获取文内元素代号的跳转
3. 普通url跳转
4. 带文字的url跳转

超链接跳转基于`hyperref`宏包, 这个宏包的功能包括但不限于生成超链接, 甚至可以更改纸张尺寸和更改`.pdf`文件的属性, 是一个极其强大但混乱的宏包: 它重新定义了许多默认的指令, 生成需要复杂排版的纸质文件时极其推荐不要调用它.

效果如下:

1. 跳转至[这里](#)
2. 跳转至1
3. <https://www.google.com>
4. [PornHub](#)

注意到:

- `\hyperref[落点名]{呈现的元素}`命令中的呈现的元素可以是整整一章, 也可以是一整张图, 也可以是一张子图, 也可以是表 (具体效果详见附录);
- 需要用`\label{名字}`命令来标识超链接的落点位置;
- 点击文内跳转超链接后, 落点会在当前显示区域最上方;
- `\ref{名字}`命令仅可获取目标的序号, 而不能获取其元素类型 (section, figure 等), 需要自行注明, 可以参看[此处](#)源代码;
- 对于图片和表格等元素, 若要作为落点, 则 `\label{名字}`须紧贴相当于`\end{...}`的命令前才可被`\ref{名字}`命令正常识别出其序号, 可见图2(a)和表1的源代码;

- 慎点超链接, 作者都可能不知道它会跳到哪里

- 附录

附录的设置同时需要在[导言区](#)和附录的代码部分前进行命令。

附录代码处需要用以下命令环绕

```
\begin{appendices}
  设定
  \section{...}
  ...
  \section{...}
  ...
\end{appendices}
```

此处的设定为:

```
\def\thesection{附录\Alph{section}}
\appendixpage
\noappendicestocpagenum
\addappheadtotoc
```

其作用分别为:

1. 令附录章节名样式变为“附录 A”，“附录 B”...;
2. 令附录第一页的左上角出现标识字样，字样设定则是在导言区;
3. 上一条命令定义的标识字样其实也被算作一种有序章节，故而会在目录中出现页号，此命令会抹去其页号;
4. 令“附录”字样出现在目录中。

易知: `\def\thesection{附录\Alph{section}}`可用于自定义各种元素的序号样式:

- `\thesection`中的`section`可以换成`figure`或是`table`等元素名;
- 后面的大括号即为样式设定:

假如我在这里写

```
\def\thesection{\textit{电子烟假尼古丁真}\alph{section}}
```

则之后的附录章节名会变成“电子烟假尼古丁真 a”，“电子烟假尼古丁真 b”，...

各部分具体含义如下:

1. `\def`表示对此后的元素重新定义;
2. `\thesection`表示章节序号样式;
3. 大括号内即为指定的样式:

* 在`ctex`系列下，系统会自动在ASCII字符和全角字符 (汉字) 间加入一个半角空格;

相应地，作为处理中文的格式，汉字 (Chinese characters) 间没有空格，故而在源代码汉字间的空格与断行与不会被识别，需要用`_`命令强加进去;

由于字母语言的特性，需要用半角空格断字，若在源代码文字部分的字母间连续输入不少于 1 个的半角空格或者是断行，最终也仅会输出一个半角空格;

* 可以对此处的内容用命令进行添加下划线, 改变颜色或是变为斜体等操作;

* `\alph`表示小写英文字母用作序号, 类似地, 还有:

`\arabic`表示阿拉伯数字;

附录默认的`\Alph`表示大写英文字母;

`\Roman`表示大写罗马数字;

`\roman`表示小写罗马数字.

仅有阿拉伯数字序号可以表示第负数个元素, 其余均不支持非正数序号, 但仅有大/小写英文字母会严重报错, 罗马数字仅会消失;

当且仅当序号始终是不大于 26 且不小于 1 的整数时, 才能选择英文字母用作序号, 否则严重报错; 换言之, 附录, 第二层与第四层`enumerate`环境内的条目数不得大于 26 个.

• 参考文献

参考文献的引用与参考文献列表的引入由以下设定实现:

1. 到期刊官网上, 往往能在引用界面看到一个BibTeX的功能, 点击后会生成一段代码或是下载一个`.bib`文件, 亦或是`.ris`文件. 这就是 L^AT_EX 的参考文献数据文件.

`.ris`文件用于在文献管理软件中导出BibTeX代码, 此处不再赘述.

2. `.bib`文件可以直接在大部分的 L^AT_EX 编辑器中直接打开, 也可作为`.txt`文件打开. 然后会发现其内是形似

```
@article{cite-key,
  annote = {doi: 10.1021/jacs.3c11569},
  author = {Li, Jing-Chang and Tang, Jiayi and Tian, Jiaming and Cheng, Chen and Liao,
  date = {2024/03/20},
  date-added = {2024-03-25 19:26:10 +0900},
  date-modified = {2024-03-25 19:26:10 +0900},
  doi = {10.1021/jacs.3c11569},
  isbn = {0002-7863},
  journal = {Journal of the American Chemical Society},
  journal1 = {Journal of the American Chemical Society},
  journal2 = {J. Am. Chem. Soc.},
  month = {03},
  number = {11},
  pages = {7274--7287},
  publisher = {American Chemical Society},
  title = {From Oxygen Redox to Sulfur Redox: A Paradigm for Li-Rich Layered Cathodes},
  type = {doi: 10.1021/jacs.3c11569},
  url = {https://doi.org/10.1021/jacs.3c11569},
  volume = {146},
  year = {2024},
  year1 = {2024},
  bdsk-url-1 = {https://doi.org/10.1021/jacs.3c11569}}
```

的BibTeX代码.

3. 此时, 需要在 \LaTeX 文档内创建一个**.bib**文件, 然后将这段代码粘贴进去; 一个**.bib**文件可以容纳多篇文献的数据;
4. 注意到, BibTeX代码第一行的**cite-key**为此文献在被引用时的代号, 可以任意编辑, 但须保证唯一性;
5. 在想生成参考文献列表的地方¹⁷插入如下两行代码即可:

```
\bibliographystyle{unsrt}
\bibliography{reference.bib}
```

其中第一行规定了文献列表的样式¹⁸, 此处选择的样式**unsrt**会按照在文章中被引用的顺序摆放, 并且显示各名词全称;

第二行则规定了将被导入的**.bib**文件的路径和文件名. 同前, 若该文件和 \LaTeX 文档在同一文件夹则不需要注明路径.

\LaTeX 的优势之一便是参考文献列表的格式绝对正确;

使用效果如下:

电子 [ConcreteMath][Er01][Knuth92, ChenXiangDong2015] 烟假 [Simpson] 尼古 [alum]
丁真 [cite-key]
电 [cuth] 子 [greenwade93] 烟 [highschool] 假 [hy] 尼古丁真 [machine][miss][spec][utf8].

点击以上两行文字的任意一处引用标便可跳转至参考文献列表.

¹⁷习惯上, 参考文献列表一般在正文和附录之间, 是否要新起一页则是看具体情况下的要求

¹⁸另有 **abbrev** 等

3 注意事项

3.1 编译一篇文章

中文环境下推荐使用 $\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$ 编译。

因为 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 的特性, 一个 .tex 文件在第一次编译时仅会导入元素加初步排版. 一些基于文章结构的精细操作, 例如超链接与目录的生成, 则不会完成, 须编译第二次。

若一个文章有参考文献, 则须在第一次编译后再使用 $\text{Bib}\text{Te}\text{X}$ 引擎编译其参考文献列表: 编译 .tex 文件会生成生成记载了结构信息的 .aux 文件, $\text{Bib}\text{Te}\text{X}$ 引擎会对其内容做出反应, 去找到参考文献数据, 再提取其中的数据. 现在有了新的数据, 按照刚刚所说的, 我们又需要编译 2 次这个 .tex 文件, 才能把参考文献数据导入并排列好。

也就是总计要编译四次: $\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X} \rightarrow \text{Bib}\text{Te}\text{X} \rightarrow \text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X} \rightarrow \text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X} \rightarrow$ 我们所需的 .pdf 文件

3.2 处理日语

处理日语时, 可以用日本人在维护的, 原生支持日语的 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 引擎编译. 而因为其中只有 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 原生支持 UTF-8 , 故而也可以用它编译 ctex 文档类。

日本人最主流的标准文档格式, js 系列, 仅可被 $\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 识别. 而这两个引擎有别于中文最常使用的 $\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$, 它们并不直接生成 .pdf , 而是生成一个作用类似于“pdf 生成指导”的 dvi 文件, 然后再跑一遍 dvi2pdf 命令, 才能得到 .pdf 。

$\text{p}\text{L}\text{A}\text{T}\text{E}\text{X}$ 和 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$ 对内容的导入与排列逻辑与别的引擎相同, 故其也须多次编译才能得到完成数据导入再排版. 但是, 它需要在最后再加上一次 dvi2pdf . 目前主要使用 dvipdfmx 作为 .pdf 生成器. 这其实是一种相对传统的 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 编译方式, 但是过于传统。

目前的国际主流是 $\text{L}\text{A}\text{T}\text{E}\text{X}$, $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ 与 $\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$ 这三个引擎¹⁹, 故而对于极其小众的日本引擎, 多数宏包是默认不按照其特性工作的: 需要在文档类处进行标注:

- 中文使用 $\text{\documentclass[a4paper,titlepage]{ctexart}}$, 其中:
 1. 指定纸张尺寸为 A4 , 尽管这是默认值;
 2. 指定生成标题页: 自动整体居中且不计页码.
 另有:
 - twocolumn , 生成双栏文档, 也就是常见的期刊格式那种两竖排文字;
 - 10pt , 11pt , 12pt 三个全局字号²⁰选项 (三选一);
- 日文使用 $\text{\documentclass[a4paper,titlepage,uplatex,dvipdfmx]{jsarticle}}$, 其中:
 1. 指定纸张尺寸为 A4 , 尽管这是默认值;
 2. 指定生成标题页: 自动整体居中且不计页码;
 3. 标明 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 引擎为 $\text{up}\text{L}\text{A}\text{T}\text{E}\text{X}$;
 4. 标明 dvi2pdf 引擎为 dvipdfmx .

js 系列中自带此处的命令, 故应将其删去. 另外, 导言区的字样设置也应更改为日语汉字。

¹⁹最新的 $\text{Lua}\text{L}\text{A}\text{T}\text{E}\text{X}$ 还很慢, 故而“不受待见”

²⁰基于 article , 不同文档类支持的全局字号不同

附录

附录 A 举例环境外的行间公式与正文的间距示例

电子烟假尼古丁真 RELX5

$$E = mc^2 \tag{3}$$

电子烟假尼古丁真 RELX5

附录 B 列表环境的层级与标识符示例

- 第 1 层 **itemize**

- subitem 样式 (itemize-subitem)

- subsubitem 样式 (itemize-subitem-subsubitem)

- 第 2 层 **itemize**

- subitem 样式 (itemize × 2 + subitem)

- subsubitem 样式 (itemize × 2 + subitem-subsubitem)

- * 第 3 层 **itemize**

- subitem 样式 (itemize × 3 + subitem)

- subsubitem 样式 (itemize × 3 + subitem-subsubitem)

- 第 4 层 **itemize**

- subitem 样式 (itemize × 4 + subitem)

- subsubitem 样式 (itemize × 4 + subitem-subsubitem)

1. 第 1 层 **enumerate**

- subitem 样式 (itemize × 4 + enumerate + subitem)

- subsubitem 样式 (itemize × 4 + enumerate + subitem-subsubitem)

- (a) 第 2 层 **enumerate**

- subitem 样式 (itemize × 4 + enumerate × 2 + subitem)

- subsubitem 样式 (itemize × 4 + enumerate × 2 + subitem-subsubitem)

- i. 第 3 层 **enumerate**

- subitem 样式 (itemize × 4 + enumerate × 3 + subitem)

- subsubitem 样式 (itemize × 4 + enumerate × 3 + subitem-subsubitem)

- A. 第 4 层 **enumerate**

- subitem 样式 (itemize × 4 + enumerate × 4 + subitem)

- subsubitem 样式 (itemize × 4 + enumerate × 4 + subitem-subsubitem)

其实能输出这么多层 itemize 和 enumerate, 但是第 7 层和第 8 层 (带 begin 和 end 的) 都会报错: 可能是学的 html 的 h1 ~ h6. 但是并不会不输出, 只会说 **Too deeply nested.**

而且, 里面还可以疯狂套 subitem 和 subsubitem.

但是, 须注意, subitem 和 subsubitem 行首是没有标识符的, 想要标识符就只能用 begin 和 end 新开一层. 正常人也开不了这么多层, 这里只是示范极限.

附录 C Excel 表格导出示例

1回目	2回目	3回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

表 2: 表格示例

1 回目	2 回目	3 回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

图 3: Execl 截图

在这里可以做个实验：如果源代码中`\hfill`的上下有空行，则两者马上会错开到上下布局。可见虽说 \LaTeX 整体对缩进和空格不敏感²¹，但还是尽量标准一些写代码比较好。

`\hfill`上空行:

1回目	2回目	3回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

图 4: Execl 截图

表 3: 表格示例

1 回目	2 回目	3 回目
k/T	k/T	k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

²¹比 Python 强

\hfill下空行 (新摆一页是因为前面放不下了):

1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

图 5: Execl 截图

表 4: 表格示例

1 回目 k/T	2 回目 k/T	3 回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

\hfill上下空行:

1回目 k/T	2回目 k/T	3回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353

图 6: Execl 截图

表 5: 表格示例

1 回目 k/T	2 回目 k/T	3 回目 k/T
6.45833E-05	6.18056E-05	8.40278E-05
0.000139249	0.000146075	0.000223208
0.000106711	0.000281208	0.000195973
0.000179538	0.000130363	0.000131353