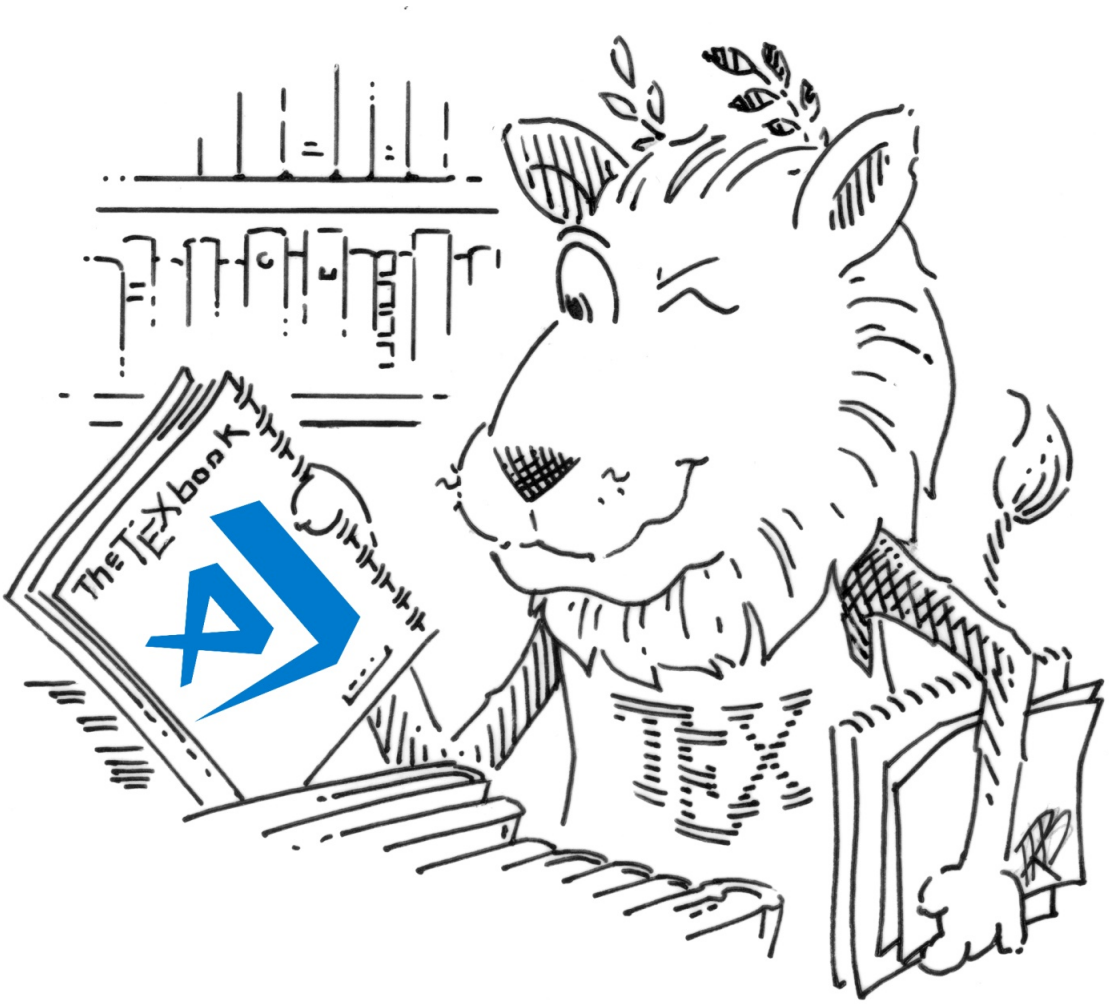
LaTeX 小白高效入门手册

Liuding.xin♡

2021-05-20

教你用 VSCode + LaTeX 搞定毕业论文



版权声明

目录

插图

表格

前言

无论你是在哪里看到我做个文档，其实我都不是跟你教学 LaTeX，我不是专业的 LaTeX 高手，我这个文档也不是为专业的用户而写。

在作为一个门外汉，我在接触 Markdown 写公式之后，感受到 LaTeX 对数学的强大地支持，就觉得有必要学习这个专业的排版系统，感受它的魅力。因此从零开始折腾，从入门知识，到开始写文档，添加各种内容，使用各种环境，开始对细节进行调整和把控。我这里只是给你一个最简单最基础的入门。让你能做出最简单的文档，让你能够在学习过程中，知道该如何入门学习，该如何去找资料看，该怎么样让环境工作起来，让你能够提笔写东西的时候，能生成一个最基础的文档，而不是这里查查，那儿看看，最后连环境都无法编译起来，甚至不知道该如何下手，进而导致放弃的念头。

要完成本文所需要的 LaTeX 的知识点其实在本文中已经完全包括，**也就是说只要本文作为参考，你就能写出和我这篇文章排版效果一致的文章**。我这里的基础知识很多东西都是非常简陋的，很多是摘抄自刘海洋老师的书籍 [1]。一来，我本身不是很专业的作者，二来是我就是为了入门，刚好也可以把自己的学习过程记录下来。本文记录了我的陡峭的学习过程和学习遇到的困难。但你在看到我这个文档时候，我已经完成了蜕变。恭喜我自己，也恭喜你，我们一直在进步。

整个过程，最终来看，我前前后后并没有花很长时间就入门了。所以加油吧。其实不难。当然完成此文章得梳理和编写，其实有点困难的，毕竟边学边需要双份脑子。但是如果你按照本文学习，真的不是很难。希望大家在看到这个文档的时候，点个赞！♡♡♡

本文托管在 Github 中，链接地址：https://github.com/heartacker/MyNotes，使用 VSCode [1] 作为编辑器，TexLive 2021 [2] 作为编译器进行编写。可以转到 release 页面获取此文档最新版本。

[1] 微软背书的开源全语言跨平台编辑器

[2] 建议使用清华镜像下载：https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/?C=M&O=A

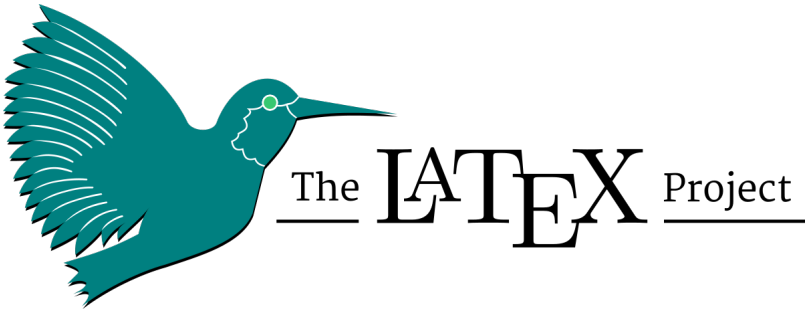
摘要

本文是记录在学习 LaTeX 过程中遇到的问题和解决办法，并用 LaTeX 写出来。本文所涉及到的内容足够你完成和本文一样的输出效果。

1 LaTeX 基础

1.1 什么是 LaTeX？

TeX 是高老师编写的非常流行的**排版软件**。LaTeX 是基于 TeX 之后，集合更多宏包的再发行版本。LaTeX 更加简单更流行，但是还是很难入门，因为他们都是使用 markup 语言进行编写，而不是类似 MicroSoft Word 那样所见即所得，所以门槛非常高。但是一旦学习并熟悉，对论文的编写是非常有用的。尤其是他的模板功能。而且任何东西都是可以细粒的控制的。正是因为门槛很高，导致学习非常陡峭，让很多人想入手都不知道如何入手。



本人也是折腾了很久之后，终于找到了学习方法和技巧。后续将会简单的给大家讲解如何上手。

这里将会讲解非常简单的基础。

当然入手难的问题实在是要靠自己去解决，所以不要认为最好地排版软件就是最适合自己的。你可以根据自己的需求选择不同的排版软件。

比如:

1. Microsoft Word.
2. LaTeX
3. Markdown

但是如何选择可以参考：

- ✓ 选择使用 TeX/LaTeX 的理由包括：
  - 免费软件；
  - 专业的排版效果；
  - 是事实上的专业数学排版标准；
  - 广泛的西文期刊接收甚或只接收 LaTeX 格式的投稿；
- ✓ 不选择使用 TeX/LaTeX 的理由包括：
  - 需要相当精力学习；
  - 图文混合排版能力弱；
  - 仅流行于数学、物理、计算机等领域；
  - 中文期刊的支持较差；

1.2 如何查询文档和学习？

对于这种非常陌生羞涩难懂的软件。你会觉得连查文档都很困难。不过如果你想对某个模块阅读文档，可以通过下面的命令打开英文文档。

```
> texdoc 模块的名字
> texdoc listings
```

比如如何设置代码显示的格式，我们都可以用哪个上面的方式。找到 listlings 的官方文档。但是要求你对英语比较熟练啦。

本文虽然只是给你讲解一些快速入门的东西。但是如果你想真正地学习好 LaTeX 那么，看一本相关书籍是必备的。推荐这本入门书籍 [1] 。建议先看: LaTeX 入门（刘海洋）.pdf

1.3 LaTeX 文类层级

LaTeX 有三种标准文类：book，report，article。中文是 ctexbook，ctexrep ，ctexart。

每种文类的章节命令和层次深度如下：

层次深度	层次名	book	report	article
-1	part	\part	\part	
0	chapter	\chapter	\chapter	\part
1	section	\section	\section	\section
2	subsection	\subsection	\subsection	\subsection
3	subsubsection	\subsubsection	\subsubsection	\subsubsection
4	paragraph	\paragraph	\paragraph	\paragraph
5	subparagraph	\subparagraph	\subparagraph	\subparagraph

表 1.1: 不同标准文类的章节命令与层次深度

1.3.1 article 渲染效果

使用 ctexart 文档类型进行渲染。

```
    \documentclass[UTF8]{ctexart}
```

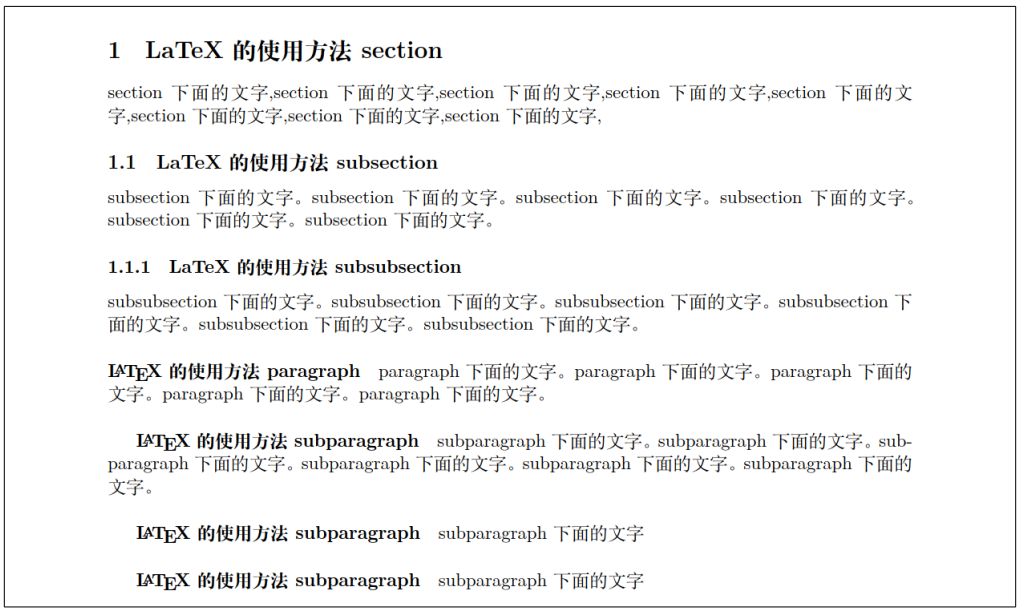


图 1.1: article 渲染效果

1.3.2 ctexart 渲染效果

使用 ctexart 文档类型进行渲染。

```
1 \documentclass[UTF8]{ctexart}
```

图 1.2: ctexart 渲染效果

对比两者可以看到。ctexart 针对缩进做了更改。而 article 只有 subparagraph 做了缩进，但 ctexart paragraph 竟然是不缩进的。需要注意。

1.3.3 段落渲染

段落 1

段落 1

子段落 1 子段落 1

子段落 2 子段落 2

段落 2

段落 2

子段落 2 子段落 2

1.4 LaTeX 编码问题

上述 层次结构 中，我们知道有三类 book，report，article（中文是 ctexbook，ctexrep ，ctexart）。但是如果我们需要在西文中插入中文要如何弄呢？这就需要引用 xeCKJ 包了。参考如何书写多国语言。

1. ✓ 引用包`usepackage{exCJK}`
2. ✓ 在需要的写入中文就好。

```
\documentclass{article}
\usepackage{xeCJK}
\begin{document}
    \title{这是使用 xeLaTeX 进行编写的的 英文article}
\maketitle

\section{测试}
这是中文
\end{document}
```

3. ☐ What is the difference between CJK and CJK-utf8?

1.5 TeX 源代码编码格式和 LaTeX 的编码

源代码的文件编码是指这个 tex 的源代码是用什么编码保存的。和你要输出什么文档是没有关系的。比如你可以完全用 ascii 的编码写 article 的源代码。但是你也可以用 utf-8 的编码进行编辑只有 ascii 字符的文章。

前者代表当前的文档是用什么格式保存源代码。后者是指生成的文档里面有什么编码的问题。一开始 TeX 是只支持 ascii 编码的原代码的。也就是说，你在原代码你面写一个中文的注释，TeX 编译器是无法识别这个中文注释的。

但是随着 pdfLaTeX 和 xeLaTeX 等采用 UTF-8 进行开发，对中文和其他多字节字符的编码已经完全支持。你能看到这个中文文档，说明这个文档的原代码里面肯定有中文。那么这个原代码肯定不是 ascii 的编码。编译器也需要认识我这个原代码的内容。（再次说明，原代码里面中文，不代表就是支持中文的。）

而上一章节中说的编码，是指文档内容的编码。默认情况下 book，report，article 是不支持 CJK 编码内容存在的。可以通过引用 xeCJK 进行扩充。

另外，随着文件编码的支持，其实我们完全可以在原代码中输入某个符号对应的字符。但是渲染出来不一定是一样的。

比如，这是 ⇌ 和 。大家看出区别了吗？大家可能看不到后面那个符号。

其实原代码是：`$\rightleftharpoons$ 和 ⇌` 。

1.6 LaTeX 文档结构

1.6.1 LaTeX 的编写思路

要想写一个 LaTeX 的文档，其实很简单.

LaTeX 主要分为三部分。可以理解为我要写什么，我要开始写文档，我正在写文档.

1. 说明我要写说明文档，就是我要写什么;

```
% cjk 使用utf-8，中文文章使用`ctexart`，article，report，...
% 大意就是说我要写什么
\documentclass[UTF8]{ctexart}
```

2. 说明我要用到的宏包，就是我要开始写;

```
% 宏包，意思是我要开始写文档了。我会用什么工具。只有加了宏包才能写公式，做表格，放图片
% 常用的有mathtools， amsmath， graphicx， array， geometry
\usepackage{mathtools,booktabs}
```

3. 说明我要用到的环境，就是我正在写。

```

% 这里是导言区
\begin{document}% 这是一个文档的环境
    document
    \begin{figure}% 这是一个图片的环境
        图片配置
    \end{figure}
    \begin{table}% 这是一个表格的环境
        表格代码
    \end{table}
\end{document}
```

1.6.2 LaTeX 的文件类型

参考：VSCode 的配置过程

tex: tex 文件是最常见的 latex 文件，也是平时编写文章的文件

cls: cls 文件是 latex 的格式文件，规定了 tex 源文件的排版格局，称为类文件（class）

一般使用`\documentclass{}`导入

sty: sty 文件是宏包文件（package）

一般使用`\usepackage{}`导入

bst: bst 文件是参考文件的格式文件

一般使用`\bibliographystyle{}`导入

bib: bib 文件是参考文献的库

一般使用`\bibliography{}`导入

bib 文件一般如下：

```
% 引用了一个文章
@article{XXX,
    title={ABC},
    author={A, B},
    journal={XX},
    year={20XX}
}
% 引用了一个正在编写的
@inproceedings{YYY,
    title={ABC},
    author={A, B, C},
    booktitle={YY},
    pages={a--b},
    year={20YY}
}
```

如何组织自己的源代码可以参考：如何组织 tex 的文档

1.6.3 LaTeX 文件语法结构

假设在当前目录下有下列文件：main.tex、A.cls、B.sty、C.bst、D.bib，按照我们的文件格式1.6.2。我们可以大致的了解真正的 LaTeX 的语法结构。

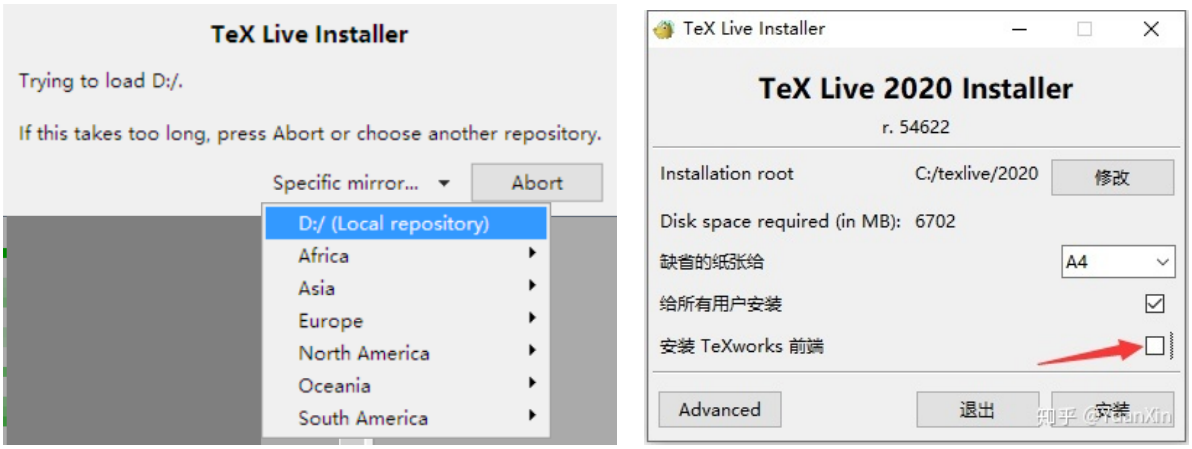
```
%main.tex文件
\documentclass{A}     % 或者不使用自定义的排版文件时，使用最普通的\documentclass{article}
\usepackage{B}        % 以及导入一些其他常用的宏文件，如amsmath、amssymb、amsthm等数学相关的宏文件
\begin{document}
this is document
this is document
this is document
this is document
% 正文结束
\bibliography{D}      % 导入正文中引入文献的数据
\bibliographystyle{C} % 导入参考文献的格式文件C.bst
\end{document}
```

1.7 常用的工具和资源链接

1.7.1 一些入门网站

入门书籍。建议先看: LaTeX 入门（刘海洋）.pdf

TexLive 安装: TexLive 2020 安装指南。可以下载离线版安装，同时不建议安装 TeXworks 前端，因为使用 VSCode 进行编辑了。



(a) 本地离线安装 (b) 取消安装 TeXworks

图 1.3: 安装勾选示意图

软件配置教程: 使用 VSCode 编写 LaTeX

软件配置教程 2： 用于 Visual Studio Code 的 LaTeX Workshop

手写识别： 手写体符号识别工具。

在线表格工具： 在线表格工具

在线表格工具： 在线表格工具 2

在线公式编辑器： 在线 LaTeX 公式编辑器, 支持导出多个格式。

插入代码： LaTeX 里「添加程序代码」的完美解决方案

学习入门总结： LaTeX 学习系列之—LaTeX 的总结

入门注意事项： LaTeX 实战经验：新手须知

论坛社区： LaTeX 工作室

1.7.2 编写、编译环境的安装准备步骤

这里就不做过多的描述了。无非就是按照以下面的步骤操作。相关的具体步骤，可以参考上上一节。

- 安装 TexLive [3]。
- 安装 VSCode。
- 在 VSCode 插件市场安装 LaTeX Workshop 插件。
- 按照 1.12 配置插件。
- Happy Writing

[3] 安装时间挺久的

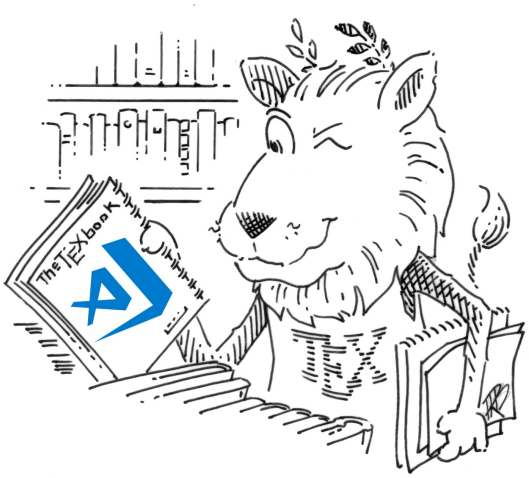


图 1.4: VSCode_LaTeX

1.7.3 常用宏包

常见的宏包推荐。来源：zhihu 孟晨

首先是国外用户写的一篇文档。9 essential LaTeX packages everyone should us，Elegant LaTeX Team 有翻译。

中文支持。

ctex - 封装好的中文支持和版式调整工具，定义了许多方便的工具

xeCJK - XeLaTeX 下的中文字体选择和禁则、压缩的处理

fontspec - XeLaTeX 下的西文字体选择

页面布局类。

geometry - 调整页面大小、页边距等尺

fancyhdr - 设计页眉页

titlesec - 设计章节标题格

titletoc - 设计目录格

超链接和 PDF 的各种功能。

hyperref - 超链接，PDF 书签，PDF 表单，PDF 元信息……

media9 - 插入 Flash、视频等

图表和浮动体。

graphicx - 插图

xcolor - 颜色

tikz - 绘图

booktabs - 三线表

multirow - 列合并单元格（cell）

makecell - 在单元格内手动换行

longtable - 换页表格

tabu - 封装了各种接口的表格宏包，制表强烈推荐

threeparttable - 在表格中使用脚标，和 tabu 有冲突，补丁：https://gist.github.com/157a703e0ed8804e7696

diagbox - 斜线表头（作者：@ 刘海洋）

float - 提供了 H 选项，禁止浮动体浮动（除非必要，不建议这么干）

placeins - 提供了 `\FloatBarrier` 命令，限制浮动体浮动范围

列表环境。

enumitem - 修改列表环境的各种间距、label 样式等的不二法门

scrextend - 提供特殊的清单

其他一些工具。

nag - 检查你是否使用了过时的宏包和命令的宏包

etoolbox - 主要是针对宏包和文档类开发者，不过提供了一些对环境的钩子，有时候很有用

xpatch - 修补命令用的

environ - 增强了 LaTeX 本来的 `\newenvironment`的功能，解决了一些花括号不匹配导致的问题

1.8 TeX 的环境的命令环境和参数

1.8.1 什么是 TeX 的环境？

以常见的文章结构为例:

```
\begin{document}
% 这里是 document 的环境

    \begin{abstract}
    % 这里是 abstract 的环境
    \end{abstract}

    \begin{section}
    % 这里是 section 的环境
    \end{section}

% 这里依旧是 document 环境。
\end{document}
```

1.8.2 什么是命令？

上面以 \ 开头的代码。就是 TeX 命令。

一个 TeX 命令（也就是所谓的宏）的格式为：

- 无参数: **command**
- 有参数: **command**{<arg1><arg2>...<argn>}
- 可选参数: **command**[<arg_{opt} >] {<arg1><arg2>...<argn>}

1.8.3 命令的参数？

先看一个

这是一个 **加粗**效果。

```
1  这是一个 \textbf{加粗}效果。
```

先看这个代码：这是一个引用，但是你会发现字号等等看起来和正文没有什么区别和变化。

孔子曰：人之初，性本善。

```
1  \begin{quote}
2     孔子曰：人之初，性本善。
3  \end{quote}
```

再来看看这个代码。可以看到，我们添加 \zihao{-5}\kaishu 改变了这个 quote 内部的字体大小和字体。

孔子曰：人之初，性本善。

```
1  \begin{quote}
2     \zihao{-5}\kaishu 孔子曰：人之初，性本善。
3  \end{quote}
```

上面我们知道。\textbf{加粗} 只改变了“加粗”，但是为什么 \zihao{-5} 改变了 quote 的所有内容呢？

我们将类似\zihao{-5} 这类命名称作声明 (declaration)。其中 **分组**限定了声明的范围。

1.8.4 什么是分组？

一个 LaTeX 环境自然就是一个分组（Group），因此，前面的 \zihao{-5} 会影响 quote 缩进。其中，最大分组就是 document 了。不过也可以用 { } 产生一个分组。

LaTeX 的环境一般格式为：

```
\begin{<环境的名字>}
    环境的里面的内容。
\end{<环境的名字>}

% 带参数的环境
\begin{<环境的名字>}[<可选参数>]<参数>
    环境的里面的内容。
\end{<环境的名字>}

\textit{花括号里面这里也是一个环境}
```

quote 是无参数的环境。

下面是一个自定义分组。这个分组我们叫做【我的定理】

在导言区域做定义：

```
\newtheorem{thm}{我的定理}
\begin{document}
```

在需要的地方使用这个环境。

```
\begin{thm}[勾股定义]
    直角三角形的斜边的平方等于两腰的平方和。
    可以用符号语言表述为...
\end{thm}
```

效果：

我的定理 1 (勾股定义) 直角三角形的斜边的平方等于两腰的平方和。可以用符号语言表述为...

我的定理 2 (等边三角形状) 等边三角形的三边边长是相等。

可以看到，文本不但添加了“我的定理”，还自动编号。对名字添加了括号。

这就是分组和环境的效果。

1.9 文章结构设计及模板

如果我们按照基本格式。按部就班的写文档。这是没有问题的。

但是，TEX 的精神是精益求精的。也不要枉费高老师[4] 为大家做出来的努力。

TEX 毕竟是标记语言的排版系统。因此，就像软件工程一样，都需要精益求精。优秀的结构设计，让文档排版事半功倍。

1.9.1 导言区

绝大多数排版是是到导言区之前使用命令定义和参数设定来对排版进行调整的。比如页边距。

```
\usepackage{geometry}
\geometry{a4paper, scale=0.8}

% 增加目录的项目，使用tocbibind 讲目录添加到目录本身。
\usepackage[nottoc]{tocbibind}
```

1.9.2 自定义环境

对于大部分同类的块，我们一般都有固定的格式。就以 quote 为例，我们在每个 quote 里面都进行调整格式是非常不方便的。我们可以自定义个环境。

```
%将这个放在导言区之前
\newenvironment{myquote}
    {begin{quote}\zihao{5}\kaishu}
    {\end{quote}}
```

之后我们就可以使用这个环境进行编写了。

> 这是使用 myquote 自定义环境的一个引用。我在这里设置了字号和字体。

```
\begin{myquote}
    这是使用myquote 自定义环境的一个引用。我在这里设置了 字号和字体。
\end{myquote}
```

这是渲染效果:

这是使用 myquote 自定义环境的一个引用。我在这里设置了字号和字体。

之后只需要对 myquote 进行设置就可以设置 quote 的格式了。

[4] 高老师为了有一个美好地排版，中间花费了很多时间开发排版软件，中止写书。直到接近 10 年之后，才回来重新写书

1.9.3 自定义命令

类似的，比如公式的角度，我们输出比较麻烦，`^\circ` 也很不直观。我们就可以使用 `\newcommand{}{}` 进行定义。

```
%新建命令
\newcommand\degree{^\circ}
%
\begin{docCommand}
    $\degree$
    .
    .
\end{docCommand}
```

之后就可以使用。替代。了。

```
之后就可以使用 $\degree$ 替代 $^\circ$ 了。
```

自定义环境在一两页的文档中，可能大家看不出效果和必要性，但是当你在维护几十到几百页的文章的时候，就可以看到这个代码的优越性。

1.9.4 模板

另外，我们经常说用模板，其实很多也是这样做的。限定各种命令和环境。你只要套用就好了。做到格式和风格统一。之前所有的努力都能得到回报。

> Tip: 理解这个环境，我们应该有所了解，我们不应该在正文中过多使用字体，字号，对齐，缩进等等。让这些排版交给环境就好。我们只需要在正确的环境中编写内容，并对环境进行配置来调整格式，做到让编写文档和格式调整分离。这样不但让自己的文档结构清晰。还能提高效率。这种我们叫做“What you think is what you want ”。

1.10 TeX，LaTeX，pdflatex，xelatex，xetex 等的区别和关系

TeX 一种宏语言。

Plain TeX
TeX 中的一个最基本的宏集合与 TeX 的基础语言构成的一种格式。

LaTeX TeX 中的一个宏集合，构成一种与 Plain TeX 不一样的格式。

TeX 程序 把 TeX 语言转换为排版的程序，也叫 TeX。为区别，称这个 TeX 程序叫 Knuth TeX。

TeX 命令 TeX 程序中的编译命令。TeX 命令默认用 Plain TeX 格式进行排版。也就是说 TeX 命令后面默认跟的 TeX 文件应该是用 Plain TeX 格式写的。

LaTeX 命令
LaTeX 命令加上某一个选项使用，就会用 LaTeX 格式进行排版，也就是说此时后面跟的 TeX 文件应该是用 LaTeX 格式写的。为方便，就把 TeX 命令与对应编译选项合成为一个命令，叫 LaTeX 命令。

ε-TeX 程序
Knuth TeX 程序的一个扩展，也是一个程序，一般写成 eTeX。增加了少量的几个命令，但一般来说是与 Knuth TeX 程序没有太多区别的。

pdfTeX 程序

TeX 语言的又一个实现，也就是把 TeX 语言转换为排版的又一个程序。它会把 TeX 语言写的代码直接编译成 PDF 文件。

pdftex 命令

pdfTeX 程序中的命令，用来编译用 Plain TeX 格式写的 TeX 文件。

pdflatex 命令

pdfTeX 程序中的命令，用来编译用 LaTeX 格式写的 TeX 文件。

XeTeX 程序

TeX 语言的新的实现，即把 TeX 语言转换为排版的一个新程序。支持 Unicode 编码和直接访问操作系统字体。

xetex 命令

LaTeX 2_ε 程序中的命令，用来编译用 Plain TeX 格式写的 TeX 文件。

xelatex 命令

XeTeX 程序中的命令，用来编译用 LaTeX 格式写的 TeX 文件。

> 实现：在文中的意思就是指“程序”的意思。如文中：eTeX 程序和 Knuth TeX 都是 TeX 语言的一个实现（也就是说，eTeX 程序和 Knuth TeX 都是把 TeX 语言转换为排版的程序。程序作用于 TeX 文本文件，把 TeX 文件编译成 dvi 文件）。

其中“实现”这个概念比较别扭，不知是不是计算机中的概念，反正非计算机专业的人读起来不知道“实现”是什么意思，不知道“TeX 语言的一个实现”是什么意思。如果把“TeX 语言的一个实现”写成是把 TeX 语言转换为排版的一个程序，这个程序作用于 tex 文本文件，把 tex 文件编译成某些文件，如 dvi，pdf 文件（比如 pdfTex 程序把 tex 文件编译成 pdf 文件）。那就好理解多了。

不知道上述理解对不对。如有错误，还请各位指点。

另外，我觉得，介绍概念时可以采用数学上的定义的方法，单独列出来每个概念的定义，每个概念的定义中不能含有前面没有定义的概念。这样就会让人好理解的多。

> PDFLaTeX 和 XeLaTeX 有什么区别？两者最大的区别是：XeLaTeX 对应的 XeTeX 对字体的支持更好，允许用户使用操作系统字体来代替 TeX 的标准字体，而且对非拉丁字体的支持更好。详情请查看这里的解释：http://stackoverflow.com/questions/15796519/difference-between-xelatex-and-pdflatex 。但是凡事有利就有弊，我在实际使用过程中发现，使用 XeLaTeX 编译，如果说论文中有很图片或者其他元素没有嵌入字体的话，生成的 PDF 文件也会有些字体没有嵌入。相反，由于 PDFLaTeX 使用的是 TeX 的标准字体，所以生成 PDF 时，会将所有的非 TeX 标准字体进行替换。所以，使用 PDFLaTeX 生成的 PDF 文件默认嵌入所有字体，这给我们的论文排版带来了极大地方便。建议大家根据自己的需要选择合适的程序。

1.11 LaTeX 编译过程

生成一个文档，需要使用排版软件进行编译生成，这就要涉及不同的步骤。

附带参考文献的整个编译需要四步。

```
1 (xe/pdf)latex main.tex   # 表示使用 latex， pdflatex 或 xelatex 编译，下同
2 bibtex main.aux
3 (xe/pdf)latex main.tex
4 (xe/pdf)latex main.tex
```

第一步后生成 main.aux、main.log 和 main.pdf 文件。其中 aux 是引用标记记录文件，用于再次编译时生成参考文献和超链接。此时的 pdf 文件中没有包含参考文件，在正文中的引用后为 [?]。

第二步后生成 main.bbl 和 main.blg 文件。blg 为 bibtex 处理过程记录文件。

第三步后更新了 main.aux、main.log 和 main.pdf 文件。此时的 pdf 文件的末尾已经有了参考文献列表，但是在正文中的引用后仍然为 [?]。

第四步同样更新了 main.aux、main.log 和 main.pdf 文件。并生成最终的 pdf 文件，此时正文中的引用后已经标记好了引用文献的序号 [1]、[2] 等。

如果对这个编译流程非常熟悉的话，就可以自己编写脚本快速编译了。当然还是推荐使用 VSCode1.12。

```
@compile.bat文件

@echo off
set CompileName="pdflatex"
for %%F in (*.tex) do (
set FileName=%%~nF
)
if not exist "./Tmp" (
    md Tmp
    )

%CompileName% -output-directory=Tmp %FileName%
bibtex ./Tmp/%FileName%
%CompileName% -output-directory=Tmp %FileName%
%CompileName% -output-directory=Tmp %FileName%

echo ------------------------------------------------
echo Compile  finished.
echo ------------------------------------------------
copy /Y ".\Tmp\%FileName%.pdf" ".\%FileName%.pdf"
start  " " /max "./%FileName%.pdf"
```

1.12 使用 VSCode+LaTeX WorkShop 进行编译

LaTeX（LaTeX 2ε）写的文档如何一键转为 Word?

通过上述编译流程1.11，对于简单的 xeLaTeX 进行编译，只需要编译一次就够了。但是如果需要生成目录等文档，就需要使用 xeLaTeX 进行编译两次。如果还要使用 bibTeX 进行引用的话，还需要使用 bibTeX 进行编译，之后在用 xeLaTeX 进行编译两次。

对于不同的编译流程，就是所谓的 recipe。不同的文档类型有不同的编译流程。

因此我们需要根据自己的需求，自己定义这个编译流程。这也是 LaTeX 入手很难的原因之一。

1.12.1 什么是 Recipe?

一个 LaTeX recipe 是指在构建 LaTeX 项目时，LaTeX Workshop 按顺序执行的一系列命令。

它是由 **latex-workshop.latex.recipes** 定义的。

latex-workshop.latex.recipes

用来定义编译流程。您可以使用不同的工具创建多个 recipes。每个 recipe 都是配置列表中的一个对象，包含一个 name 字段和一个要在 recipe 中调用的 tools 列表。

latex-workshop.latex.tools

用来定义每个编译的参数。recipes 中的 tools 可以在 latex-workshop.latex.tools 中定义，其中每个命令都是一个 tool。每个工具都是一个对象，包括一个 name、要生成的一个 command、参数 (args) 和一些特定的环境变量 (env)。env 条目是一个字典。假设您想要在您的 home 项目中使用一个 texmf 子目录，只需编写：

```
"env": {
    "TEXMFHOME": "%DIR%/texmf"
}
```

1.12.2 VSCode LaTeX Workshop 插件关键两个配置

对于中文而言，大部分使用的是 xeLaTeX，但是这个插件默认没有把它添加到工具链中。对于大部分中文用户。则使用这两个居多。

第一步，在工具链添加这个工具（latex-workshop.latex.tools）[5]

```
{
    "name": "xelatex",
    "command": "xelatex",
    "args": [
      "-synctex=1",
      "-interaction=nonstopmode",
      "-file-line-error",  //*针对 调用外部脚本的功能，比如pygment
      "-shell-escape",
      "%DOC%"
    ],
    "env": {}
}
```

第二步，在 recipes 添加这个组合（latex-workshop.latex.recipes）

```
// 针对有目录没bit引用的
{
    "name": "xelatex`×2",
    "tools": [
    "xelatex",
    "xelatex"
    ]
},
// 针对有引用的
{
"name": "xelatex → bibtex → xelatex`×2",
"tools": [
    "xelatex",
    "bibtex",
    "xelatex",
    "xelatex"
    ]
}
```

[5]第二步中会调用这个命令。

LaTeX Workshop 参数

对于命令里面的参数，可以查看下表1.2:

占位符	替换为
%DOC%	LaTeX 根文件路径和名称，不带.tex 扩展名
%DOCFILE%	不带.tex 扩展名的 LaTeX 根文件名
%DIR%	LaTeX 根文件路径
%TMPDIR%	存储辅助文件的临时文件夹
%OUTDIR%	在 latex-workshop.latex.outDir 中配置的输出目录

表 1.2: LaTeX Workshop tool 参数占位符

1.12.3 最终效果

理解了编译过程，其实很好理解上面是干嘛的了。

安装好了 TexLive 并配置好 VSCode 之后，结合上面几个描述，很快你就可以使用 VSCode 进行编译了。下图就是配置好之后最终的效果。

只要打开 TEX 源代码，就可以在左边看到相关的 Build 的操作。

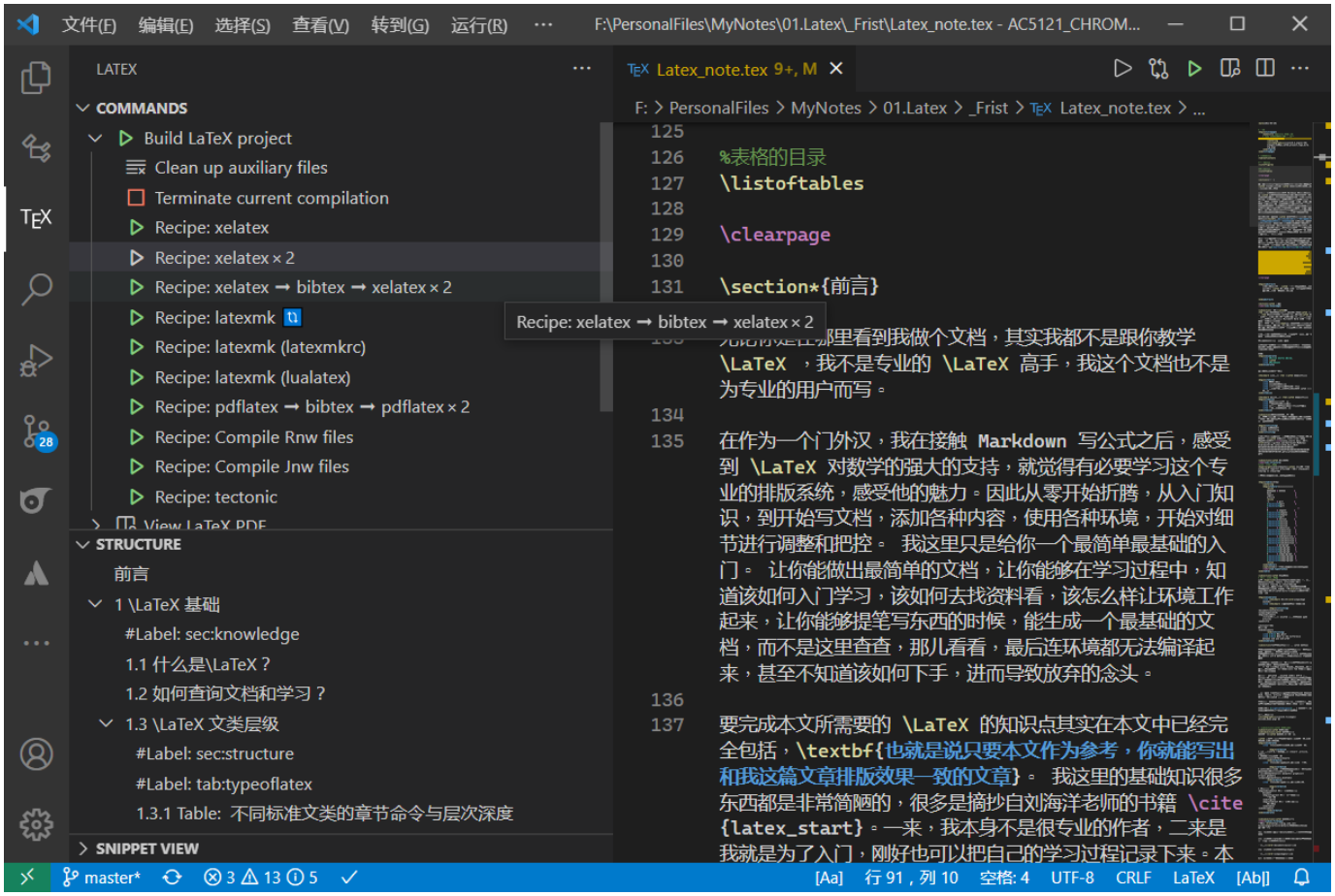


图 1.5: VSCode Latex WorkShop Demo

可以看到，这里支持多种 build 的形式。

- 简单的中文可以直接使用 xelatex。

- 如果包含目录，可以选择 xelate × 2。
- 如果包含目录、参考文档，就需要使用 xelatex → bibtex ⇝ xelatex × 2。

2 LaTeX 学习笔记

学习完基础知识1 之后，相信对 LaTeX 有一定的了解。相信你也成功的使用 VSCode + TeXLive + LaTeX Workshop 完成了一个最基本的编译。很快就可以开始下面的学习啦。接下来，只要按照每个章节，按部就班的操作一遍，就可以完成学习啦。**注意，一定要将每个步骤都用代码写一遍，才能理解，否则，仅仅是看看这个文档，是无法入门的。切记。**

2.1 如何生成目录?

制作目录其实非常简单，只需要一个命令，就是 `\tableofcontents` 。这个命令放在哪里，目录就会出现在哪里。和交叉引用相同的一个特点是，目录的排版也需要两次编译。一方面是因为其中涉及到页码，另一方面是涉及到各个章节的标题。另外，如果需要对图片插入章节信息，可以在参考章节 2.17 。

2.1.1 内置目录

Tip: 只要生成了目录，之后生成的 PDF 就会包含书签。

目录对于图表而言也是可以用的。如果你的文档中有很多图表，也可以专门为它们建目录。对应的命令是 `\listoffigures` 和 `\listoftables` 。它会收集对应图表中的标题来产生图表的目录。图表的插入我们将在章节2.20 和章节 2.21 中介绍。

注意，这会在目录下生成`.toc`，`.lof`，`.lot` 文件，分别对应【目录】、【图像】、【表格】。

1. 目录

   ```
   \tableofcontents
   ```

2. 图像

   ```
   \listoffigures
   ```

3. 表格

   ```
   \listoftables
   ```

不过，需要注意的是，在编译的时候，需要使用编译工具进行编译两次。第一次是初步编译，生成目录等临时文件，之后在将临时文件编译进去。比如第一次编译之后，生成的 PDF 文件是看不到目录的，之后再次编译，就可以把.toc、.lof、.lot 编译进去。

2.1.2 如何对某个定义对象进行编号并生成目录?

除了表格、图片之外，如何制作目录，请参考 3.5。

2.2 如何使标签、目录、引用可以跳转？

一般情况下，我们希望目录是可以跳转的，图片的引用是可以鼠标点击跳转的。

2.2.1 操作

为了实现这个功能。我们可以使用`hyperref`。操作方式：

1. 引用宏包，`\usepackage{hyperref}`，这样会默认在链接上显示红框。
2. 引用宏包，`\usepackage[colorlinks=true]{hyperref}`，这样会在链接上字体显示为红色。

目录的内容显示为红色，是因为 hyperref 宏包的 colorlinks 选项。我们以后将默认载入这个宏包，告诉大家这些红色的文字都是可以点击跳转的，这也是我非常喜欢的一个特性。

当然也可以在配置里面进行配置。

```
\hypersetup{
    colorlinks  = true,
    linkcolor   = blue,
    filecolor   = magenta,
    urlcolor    = cyan,
    % pdftitle    = {Overleaf Example},
    % pdfpagemode = FullScreen,
}
```

2.2.2 注意事项

0. 图、表建议添加“\label{标签}”。
1. label 图表建议添加 label 到环境的后面，也就是 caption 后面，否则标号不对。
2. 在需要添加引用的地方，输入“\ref{标签}”。
3. 之后就可以看到红色的引用标签了。
4. 如：\ref{sec:hyperreflink}：2.2
5. 如果只想引用不想点击，可以 \ref*{sec:hyperreflink}：2.2

2.3 如何让标题单独一个封面？

我们要将 title 显示出来的话，可以直接用一句话解决。\maketitle。

有些书籍封面是单独一个页面的，还配了图片。看起来非常舒服，这个该如何处理呢？可以参考以下代码，使用 titlepage 独立为一页:

```
%标题
\begin{titlepage}
    \maketitle %将title 编译进来
    \centering
    \includegraphics[scale=1.5]{simaqian.png} % 顺便插入图片
\end{titlepage}
```

2.4 如何创建代码和渲染效果可对比的“example”环境？

作为一个教学文章，那么有例子和渲染对比，可以更加清晰明了的向读者阐明要点，让读者一目了然的的知道代码和渲染效果对比。这也是我一开始就说这个很有难度的环境的原因之一。

先来看看效果。

$$E = MC^2$$

$$E = MC^2 \tag{1}$$

```
\begin{equation}
    % notag 代码不添加tag
    E=MC^2\notag
\end{equation}
\begin{equation}
    E=MC^2
\end{equation}
```

2.4.1 使用 showexpl

通过上面的例子，我们可以看到代码和渲染的效果。其中左边是渲染，右边是源代码。如果有 LaTeX “命令和环境”的使用示例，那么 LaTeX 宏包的文档就更容易读懂。最好的方法是对 LaTeX 代码和格式化输出进行比较。

showexpl 是一个用于进行比较的包，它基于包列表，该列表提供了良好的排版源代码，强调了关键字等等。使用 **showexpl** 首先需要先引用这个宏。之后，在需要演示的地方。并且使用 **LTXexample** 包围。

```
\begin{LTXexample}[pos=r]
    \usepackage{showexpl}
    % 源代码
    \[
        E=MC^2
    \]
\end{LTXexample}
```

之后，我们就可以得到一个渲染图。这个图包含渲染效果和源代码。读者很清晰的就可以看到对比。清晰地理解作者的用意。

```
\usepackage{showexpl}
% 源代码
\[
    E=MC^2
\]
```

$$E = MC^2$$

2.4.2 showexpl 渲染图位置设置

想要设置渲染图和代码之间关系可以通过 pos 进行配置。具体可以参考 showexpl 的文档。命令行敲打 texdoc showexpl 打开 showexpl 的文档。我们可以看到，这一次，渲染图在右边。而之前的在左边。那是因为我们使用了 [pos=r]。pos 是可选参数，参数可为 lrtb，分别代表 l 左边，r 右边，t 顶部和 b 底部。

2.5 基础字体样式调节

西文中。预定义的字体族有 3 种：罗马字体族（roman family，rmfamily），无衬字体族（sans serif，sffamily），打字机字体族（typewriter family，ttfamily）。

> Tips: 如果设置了全局字体，那么下表的渲染就是按照全局字体渲染的。比如 typewrite 的渲染就是等宽字体。所以可能看到的和默认是有差别的

字体族	带参数命令	声明命令	效果
罗马	\textrm{罗马}	\rmfamily	roman 罗马
无衬线	\textsf{无衬线}	\sffamily	sans serif 无衬线
打字机	\texttt{打字机}	\ttfamily	typewrite 打字机

表 2.1: 基本字体的渲染

2.5.1 常用样式

```
\textup{up shape} 直立（roman shape）使用（upshape）\\
\textsl{Italicized shape} 斜体使用（slshape）\\
\textit{Italic shape} 意大利体 （itshape） \\
\textsc{Small capital} 小型大写（scshape） \\ \\
\textbf{Bold text} 粗体使用（bfseries） \\
\textmd{Medium series} 中等加宽（mdseries） \\
\underline{Underlined text} 下划线使用 \\
\emph{Emphasising} 强调使用 （就是和当前的字体表现得不一样）\\
\textbf{\textit{Combine text} \underline{当然也可以组合起来}}\\
{\bfseries this is \textnormal{normal} text}\\
{\ttfamily this is \normalfont{normal} font}
```

up shape 直立（roman shape）使用（upshape）
Italicized shape 斜体使用（slshape）
Italic shape 意大利体（itshape）
SMALL CAPITAL 小型大写（scshape）

Bold text 粗体使用（bfseries）
Medium series 中等加宽（mdseries）
Underlined text 下划线使用
Emphasising 强调使用（就是和当前的字体表现得不一样）
Combine text **当然也可以组合起来**
this is normal **text**
`this is` normal font

其中 textnormal 和 normalfont 可以恢复默认的字体和样式。

这是黑体，恢复默认。
这是楷书，恢复默认。

```
{\heiti 这是黑体，\normalfont{恢复默认}}。 \\
{\kaishu 这是楷书，{\normalfont{恢复默认}}}。
```

中文没有过多复杂的变体。主要是不同的字体族进行区分。

中文的字体族根据不同的系统各有不同。Windows Ctex 默认情况下有四种字体族。分别是 song 宋体，hei 黑体，kai 楷书，fs 仿宋。

song 宋体、hei 黑体，kai 楷书，fs 仿宋。

```
song{\songti 宋体}、hei{\heiti 黑体}，kai{\
    kaishu 楷书}， fs{\fangsong 仿宋}。
```

另外，

西文的 rmfamily (罗马体家族) 正常对应的分别是宋体: 宋体，加粗则是黑体：黑体，意大利则对应 楷书：楷书。

```
\rmfamily 西文的 rmfamily (罗马体家族) 正常对应的
    分别是\textnormal{宋体}:{\songti 宋体}，加粗
    则是\textbf{黑体}：{\heiti 黑体}， 意大利则对
    应 \textit{楷书}：{\kaishu 楷书}。
```

2.5.2 中文的加粗

也不知道什么原因，参考，xelatex 编译加粗楷体为什么会失败？ 默认情况下，使用 `\textbf{加粗}` 是不会加粗的。处理方法是，在启用 `AutoFakeBold`。比如:

```
% 在class 里面启用 AutoFakeBold
\documentclass[UTF8,AutoFakeBold]{ctexart}
```

2.5.3 字体大小

一般情况下，我们很少进行独立的字体配置。文档的总体字体风格和大小在一开始就设置了全局的字体。

但是如果还是需要独立调节字体的大小，可以参考以下的设置和命令。

西文中

LaTeX 提供了 10 个简单的声明式调整文字大小的命令：

命令	意义
\tiny	tiny
\scriptsize	scriptszie
\footnotesize	footnotesize
\small	small
\normalsize	normalsize
\large	large
\Large	Large
\LARGE	LARGE
\huge	huge
\Huge	Huge

表 2.2: 西文字体大小的命令

实例

```
This is \small{small 小} and \Large{Large 大} and
    \Huge{Huge 巨大}!
```

This is small 小 and Large 大 and Huge 巨大!

中文中

可以使用中文的字号进行设置。

命令	大小（bp）	意义
\zihao{0}	42	初号
\zihao{-0}	36	小初号
\zihao{1}	26	一号
\zihao{-1}	24	小一号
\zihao{2}	22	二号
\zihao{-2}	18	小二号
\zihao{3}	16	三号
\zihao{-3}	15	小三号
\zihao{4}	14	四号
\zihao{-4}	12	小四号
\zihao{5}	10.5	五号
\zihao{-5}	9	小五号
\zihao{6}	7.5	六号
\zihao{-6}	6.5	小六号
\zihao{7}	5.5	七号
\zihao{8}	5	八号

表 2.3: 中文字号

实例

```
This is {\zihao{6} 六号 small} and {\zihao{0} 初号
    Large}!
```

This is 六号 small and 初 号 Large!

Tip: 从两个渲染图看。我们可以看到西文的大小和中文的字号是可以**同时**作用中文和西文的。

2.5.4 删除线

删除线比较复杂，但是需要引用 ulem 的宏包。

1. `\usepackage{ulem}`;
2. 使用方法和显示效果

~~删除线~~
波浪线
~~斜删除线~~
双下划线

```
\sout{删除线}\\ %删除线
\uwave{波浪线}\\ %波浪线
\xout{斜删除线}\\ %斜删除线
\uuline{双下划线}\\ %双下划线
```

2.6 全局字体设置

上面我们知道，系统自带三个字体族，分别是衬线字体，无衬线字体，打字机（等宽）字体。我们可以通过以下的方式，在导言区，设置全局字体。

```

%%%%%%中文字体设置%%%%%%
\setCJKmainfont{方正书宋简体}%设置CJK衬线字体设置
\setCJKsansfont{方正黑体简体}%设置CJK无衬线字体设置
\setCJKmonofont{方正仿宋简体}%设置CJK等宽字体设置(对应的是西文的打字机字体)

%%%%%%英文字体设置%%%%%%
\setmainfont{Times New Roman}%设置英文衬线字体
\setsansfont{Calibri}%备选Arial，设置英文无衬线字体
\setmonofont{Courier}%设置英文打印机字体
```

2.7 段落缩进

在中文文章中。使用 ctexart 是默认首行缩进的。可以通过 `\noindent` 取消缩进。但是，在英文下，是不会缩进的。就需要使用宏包进行解决。

```
\usepackage{indentfirst}    %引用宏包
\setlength{\parindent}{2em} %2em代表首行缩进两个字符
```

2.8 如何设置参考文档（论文必备）?

用 Word 文档写论文的时候，最烦的是不是做论文参考啊。哈哈哈。做论文参考就真的好烦，要不停的进行编号，进行调整。如果顺序搞错了，烦恼的不得了，当然，如果你对 Word 文档很熟悉的话，你其实也可以轻易地做到自动编排论文，但是，Word 这种所见即所得的理念，会让写作者放弃探究背后机理和秘密的念头。可以这么说，那时候作为学生对 Word 本身也不是很熟悉，而且也没有模板。缺乏对排版系统的研究。另外就是有模板了，其实我们看不到模板后面的规则，所以也很容易把格式搞错而不自知。这也是 Word 文档进行排版很烦恼的因素之一。

言归正传，我们在这里说论文参考文档的的事情。其实很简单。我们可以这样理解：

1. 我们要引用参考文章。
2. 我已经准备好了参考文档的信息。
3. 我要如何显示参考信息。
4. 我要在什么地方显示参考信息。
5. 好了，我都准备好了，请编译到文章里面去。

2.8.1 如何设置？

按照这个步骤，我们可以开始去做，步骤可以参考 [1]。

第一步，创建一个 frist.bib 的文件。内容如下：

```
% encoding: UTF8

% editor    = {高洪霞},
@book{latex_start,
    author    = {刘海洋},
    title     = {{\LaTeX} 入门},
    publisher = {电子工业出版社},
    year      = {2013-6}
}
```

第二步，引用在文档末尾设置引用论文的格式和论文的数据库[6]。

```
% 如何隐藏格式 设置
% plain 格式按照作者 日期 标题排序
% unset 不排序
% alpha 使用一种三字母的方式编号并按照作者排序
% abbrv 与alpha 基本相同，知识定义了一些缩写
\bibliographystyle{plain}

% 使用那个数据库
\bibliography{frist.bib}

% 注意是文档末尾
\end{document}
```

第三步，在需要与引用参考文章的地方。添加 \cite{数据编号}，其中数据名字就是第一步创建的数据库里面的项目名字 *latex_start*。

第四步，使用 **bibtex** 进行编译。1.11。一般情况下。中文文档需要使用 xelate → bibtex → xelatex → xelatex 就可以编译完成。也就是比其他的多了一个 xelatex 和 bibtex。

第五步，显示效果，直接看论文最后的参考文献 3.8。

2.9 如何使设置页眉页脚？

页眉页脚也是论文编写操作比较多的一点。

2.10 如何插入标签、超链接、文内链接？

超链接分好几种。比如目录，比如图片引用等2.2。

这里着重讨论其他外部连接。

label 标签

使用 `\label{text}` 。

Linking web addresses 超链接

使用 `\href{url}{text}` 或者 `\url{URL}`

[6]第一步创建的 frist.bib 的文件

For further references see Something Linky or go to the next url: http://www.overleaf.com

```
For further references see \href{http://www.
    overleaf.com}{Something Linky}
or go to the next url: \url{http://www.
    overleaf.com}
```

Linking local files 超链接

使用 `\href{run:url}{text}`

For further references see Something Linky or go to the next url: http://www.overleaf.com or open the next file File.txt

```
For further references see \href{http://www.
    overleaf.com}{Something Linky}
or go to the next url: \url{http://www.
    overleaf.com} or open the next
file \href{run:./file.txt}{File.txt}
```

Linking context addresses 文内连接

这是一个文内的跳转方式。

使用 `\hypertarget{Target}{Text}` 结合 `\hyperlink{Target}`

It's also possible to link directly any word or any sentence in you document.
If you read this text, you will get no information. Really? Is there no information?
For instance this sentence.

```
It's also possible to link directly any word
or \hyperlink{thesentence}{any sentence} in
    you document.

If you read this text, you will get no
    information. Really?
Is there no information?

For instance \hypertarget{thesentence}{this
    sentence}.
```

2.11 空格、换行、换段、换页、首行缩进？

2.11.1 空格

通常汉字后面的空格会被忽略，其他的符号后的面的空格则保留。

所以，“左 右”会被渲染为“左右”，“zuo you”则依旧正常显示“zuo you”。同时，单个换行相当于一个西文的空格。（汉字除外）(hanzi chuwai)。

```
% 上述最后的原始代码，可以看出对比。
单个换行相当于一个西文的空格。（汉字
除外）(hanzi
chuwai)
```

使用 xelatex 编译文档的时候，ctexart 文档类会调用 xeCJK 宏包，自动处理汉字于其他符号之间的距离，无论你有没有在他上面加上正确的空格。这是非常方便的。

使用一个“反斜杠 + 一个空格”（“\ ”）表示 1 个空 格，space space。多个同理，但是不推荐这样使用。表2.4 是空格大小的调整方式。

描述	符号	渲染效果	宽度
quad 空格	a \qquad b	a b	两个 m 的宽度
quad 空格	a \quad b	a b	一个 m 的宽度
大空格	a\ b	a b	1/3m 宽度
中等空格	a\;b	a b	2/7m 宽度
小空格	a\,b	a b	1/6m 宽度
没有空格	ab	ab	
紧贴	a\!b	ab	缩进 1/6m 宽度

表 2.4: 空格大小的调整参考表

Tip:

西文单词用空格隔开，LaTeX 中的连续多个空格在编译排版时被看做一个空格。但是“\ ”一定会是认为一个空格。

中文 (cjk)（ctexart）中，空格不算空格。很奇怪！你 好 会被渲染为“你好”，而不是“你 好”。另外，中英文混编 (大家可以用光标移动看看是不是):

1. 中文紧随着英文会自动添加一个空格。如你好 hello！“你好”和“hello”之间是有空格的。
2. 西文 xiwen 紧随中文也会添加空格。
3. 中文标点符号紧随的应为不会自动添加空格。如你好，hello！“你好”和“hello”之间是没有空格的。
4. 英文之间符合西文。就是一个或者多个算一个空格。hi hello 显示为 hi hello，hi 和 hello 之间是有一个空格的。
5. **任何时候，中西文混编都推荐在中西文之间添加空格，这样是符合规范的**，“space 你好 space 你好 space”。

2.11.2 句号后面的空白

句号圆点，可以表示句子结束，也可以表示缩写。在小写字母后面圆点表示句子结束，但是为了表示缩写，需在圆点后空格加上倒斜线\，表示缩写后面可以分行; 在 LaTeX 中，大写字母后的圆点看做是一个缩写而不是句号。有时大写字母后面圆点表示成句号，可以在圆点前加上”\@.”，即显示为“.”。

2.11.3 换行命令

Tip: 换行和段落是有本质的区别的。换行只是当前段落换行而不是新起一个段落，但段落是有独立的排版特性的！连续两个回车被看做是段落结束，可以使用\\强制换行而不是段落。

\\ : 换行。

\\[offset] : 换行，并且与下一行的行间距为原来行间距 + offset。如何 \\[20pt] 显示为

这个样子，中间间距很大。

\newline : 与\\相同。

\linebreak : 强制换行，与\newline的区别为:\linebreak 的当前行分散对齐。(就是说上一行会自动占满整整一行，但是没有更换段落，看看这行的效果)

2.11.4 分段命令

\par：分段。

2.11.5 分页命令

\newpage: 换一页。
\clearpage：和 \newpage 类似。我们在使用 CJK 环境时会加入 \clearpage 在环境末尾。

2.12 如何输出反斜杠?

要想打印“\”比较麻烦。参考链接:https://blog.csdn.net/xovee/article/details/106728213

1. 文本中
   (a) 使用“\verb!\!”，注意是“!!”之间进行截止
   (b) 使用$\backslash$
   (c) 使用代码

```
\usepackage{verbatim}
\begin{verbatim} % 使用代码的的形式显示
    \
\end{verbatim}
```

2. 标题中 (不建议使用这个方法。会搞乱目录和书签)
   (a) 引用 cprotect 宏

```
\usepackage{cprotect}
% 在标题中显示原始字符
```

   (b) 在标题中插入

```
\cprotect\section{This is a section heading with a verbatim \verb!\frac{1}{2}!}
% 这是在标题中使用\verb
```

2.13 其他特殊符号

2.13.1 常见特殊符号

除了 \ 之外，含有很多符号，其他特殊符号:

```
#  ——  \#
$  ——  \$
%  ——  \%
{  ——  \{
}  ——  \}
~  ——  \~{}
^  ——  \^{}
\  ——  $\backslasb$
```

可以看到，最后三个是比较特殊的。

不过中文还是建议使用中文符号。

逗号，	省略号……	花括号 { }
句号。	分号；	方头括号 【】
顿号、	双引号 “	
问号？	单引号 ‘	空心方头括号 〖〗
感叹号！	双书名 《》	六角括号 〔〕
波浪号 ～	破折号——	
冒号：	方括号 [ ][7]	圆括号 ( )

2.13.2 引号

西文中，引号分别是' 和 ' 表示。单引号用一遍，双引号用两遍。如'nihao' 和"nihao" ~~和"nihao"~~。
西文中，如果出现连续引用。则用 `\,`（反斜杠 + 逗号）隔开。如`''\,'a'\,''` 显示为："'a'" 。
中文中，引号分别是 ‘和 ’ 表示。单引号用一遍，双引号直接用中文的双引号，“nihao”。

Tips:

英文的双引号极少使用。常常用作其他用处。

另外，中文文章中显示的英文' ' 都是逗号方向的样子，而不是左右对称的。这个看起来不是很舒服。所以中文文章中建议使用中文的 “，而不是英文的' '。

2.13.3 省略号

使用`\ldots`。

英文文中。使用 i...you.
英文句尾，使用 i amd you ….

```
英文文中。使用 i$\ldots$you. \\
英文句尾，使用 i amd you \ldots.
```

2.13.4 连体号

在 LaTeX 中，把 fi 这样的字母组合，当成一个连体号排版出来的，而不是分开来的。如要将两者分开，应在两者中间加入 “\/”，即 “`f\/i`”，就可以显示 “fi”，而不是 “fi”。

不同的字体不一样。有些专业的字体会有很多，如 `fb,fh,fj,ffb,ffh,ffj,ffk,ct,st,sp,Th,fi,fl` 等。具体看当时的编排效果。

2.13.5 连字号、破折号

除了在数学模式中 – 表示减号之外，符号 “-” 在 LaTeX 有多中用途。

- 一个连字号 “-” 表示一个连字号 “-” 。如 latex-A。
- 连续用两个连字号 “--” 表示数字范围符号–(~:`sim$!)`。不过中文喜欢用中文符号 verb! “ ”！表示范围。
- 连续用三个连字号 “---” 表示—破折号。

2.14 颜色

多彩的世界，需要五彩斑斓的渲染。tcolorbox 教学。参考链接：xcolor 的教学

[7] 参考 https://www.zhihu.com/question/19710761

2.15 注脚

2.15.1 使用方法

注脚[8]是很常用的功能[1]。

```
%代码
注脚\footnote{这是注脚的注脚}是很常用的
功能\footnote[1]{这是第二个带号码的注脚}。
```

2.15.2 自定义注脚的样式

Todo

[8] 这是注脚的注脚

[1] 这是第二个带号码的注脚

2.16 有序和无序列表

2.16.1 无序列表

使用 itemize 可以生成序列表，当然也可以和 enumerate 嵌套使用。代码：

- 第一个
- second
  - a
  - b
- 无序中有序
  一、 中文
  二、 编码
  (a) 有序
  (b) 嵌套

```
\begin{itemize}
  \item 第一个
  \item second
    \begin{itemize}
       \item a
       \item b
    \end{itemize}
  \item 无序中有序
    \begin{enumerate}[label = \chinese*、]
       \item 中文
       \item 编码
          \begin{enumerate}
             \item 有序
             \item 嵌套
          \end{enumerate}
    \end{enumerate}
\end{itemize}
```

大家看到没，其实还有**中文编码** 的。中文编码之后的嵌套系列还是会会切换成英文。

2.16.2 有序列表

使用 enumerate 可以生成有序列表，但是也可以和 itemize 嵌套使用。代码：

1. 第一个
2. 嵌套
   一: 中文
      i. 编号 i
      ii. 编号 ii
   二: 编号
3. 有序中无序
   - 第一个
   - second
   - 第三个

```
\begin{enumerate}
  \item 第一个
  \item 嵌套
    \begin{enumerate}[label = \chinese*:]
       \item 中文
          \begin{enumerate}
             \item 编号i
             \item 编号ii
          \end{enumerate}
       \item 编号
    \end{enumerate}
  \item 有序中无序
    \begin{itemize}
       \item 第一个
       \item second
       \item 第三个
    \end{itemize}
\end{enumerate}
```

2.16.3 如何自定义编号或者使用中文编号?

这属于高级技巧。参考章节 3.6

2.16.4 特殊清单

参考链接：http://softlab.sdut.edu.cn/blog/xuqianhui/2017/06/20/latex□□□□□□□□/

这属于高级技巧。参考章节 3.6

2.16.5 自定义列表清单

这属于高级技巧。参考章节 3.6

2.17 如何在表格和图表之间插入章节信息？

参考链接：LaTeX 技巧：在图表序号中加入章节号（实现诸如“图 1.1.2”这样的图表序号） 。

2.17.1 插入章节编号

如何插入表格和图片还没有讲解。但是这里先教一下如何插入章节信息在图表中。

Latex 在默认情况下，Figure 图像的编号是从 Figure 1，Figure 2，Figure 3，…，Figure N 顺序递增的。虽然在文章只有一个章节的情况下，这种简单的顺序递增的 Figure 图像编号没有问题。但是当文章涉及到多个章节的时候，这种顺序递增的 Figure 图像编号就不符合文章排版规范。举个例子，第三章的第一个 Figure 图按照排版规范，应该编号为 Figure 3.1，但是默认的 Latex 的 Figure 图像编号却将它编号为 Figure 1。

1. 引用宏包 `\usepackage{amsmath}`
2. 在 document 之前添加这行代码。

```
% https://blog.csdn.net/Canhui_WANG/article/details/87364800
\numberwithin{figure}{section}    %对应图
\numberwithin{table}{section}     %对应表
```

2.17.2 如何自定义系列数字为中文？

参考LaTeX 中 enumerate 环境的使用技巧

2.17.3 自动插入表图到引用？

todo

2.18 如何插入数学公式？

LaTeX 对数学和科学支持最好的排版系统。这也是为什么科技工作者、科研人员、计算机科学家等非常喜欢使用 LaTeX 写文章的原因。https://www.zhihu.com/question/38306880

2.18.1 行内公式

行内公式叫做 inline formula，只需要 $ $ 括起来，就是一个公式环境，比如`$a+b$` 会得到 $a+b$，而不是 a+b。

2.18.2 单块列表公式

较长的公式或者大块公式，一般单独居中编写，为了方便引用，还要进行编号。我们一般叫 displayed formula。使用 equation （单行）和 gather[9] （多行）等环境方便输出。

[9] 需要引用 amsmath 宏包

1. 环境 1

   单块行公式使用\[\] 进行包围。如\[\frac{a}{b}=c/d\]:

$$\frac{a}{b}=c/d$$

   这是单块行公式。

2. 环境 2

   可以使用 $$ $$ 双美元进行包围。如$$\frac{a}{b}=c/d$$:

$$\frac{a}{b}=c/d$$

3. 环境 3

   也可以使用 equation 。但是他会自动带 tag

```
\begin{equation}
    \angle CAB = 90^\circ =90\degree=\pi / 2
\end{equation}
```

   如:

$$\angle CAB = 90^\circ = 90^\circ = \pi/2 \tag{2}$$

   插播一下，还记得之前定义的 degree 1.9.3 吗？这里用上了。

   需要注意的是：不想加 tag 可以添加 \notag 的标记。当然。任何加 * 的环境是不会添加 tag 的。这个和 LaTeX 的其他环境一样。

2.18.3 列表公式

多行公式是必备的。以下是常用的例子。参考：公式对齐设置。

2.18.4 写不完的多行

一行写不下的，要写多行的，用 multiline 。

$$\begin{aligned} p = 4x^5y + 590x^4y^2 + 19x^3y^3 \\ + \sin x + \cos y + \tan a + e^{x+y} \\ - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \end{aligned} \tag{3}$$

```
\begin{multline}
    p = 4x^5y + 590x^4y^2 + 19x^3y^3 \\
    + \sin{x} +\cos{y} +\tan{a} +e^{x+y}\\
    - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3
\end{multline}
```

2.18.5 有对齐的的多行

多行的公式。而不是一行写不下的公式，使用 align 环境以 & 进行对齐。

$$\begin{aligned} X^{ABC} &= \int_a^b x\mathrm{d}x \\ 1\times 2 &= 2 = 4/2 = 8/4 = 16/8 \end{aligned}$$

$$\begin{aligned} a &= b + c \\ &= d + e \end{aligned} \tag{4}$$

```
\begin{align*}
    X^{ABC}          & = \int_a^b x \mathrm{d
        }x \\
    1\times 2=2=4/2=8/4 & =16/8
\end{align*}

\begin{align}
    a & = b + c     \\
      & = d + e \notag
\end{align}
```

$$a = b + c \tag{5}$$
$$= d + e \tag{6}$$
$$= d + e$$

分别交叉引用，依旧是使用 label。式 (5) 和式 (6) 采用 align 对齐环境。使用 notag 取消编号。

2.18.6 多列对齐 align

$$a = 1 \qquad b = 2 \qquad c = 3$$
$$d = -1 \qquad e = -2 \qquad f = -5$$

```
\begin{align*}
    a & =1 & b & =2 & c & =3 \\
    d & =-1 & e & =-2 & f & =-5
\end{align*}
```

2.18.7 不对齐 gather

$$a = b + c \tag{7}$$
$$d = e + f + g \tag{8}$$
$$h + i = j + k$$
$$l + m = n \tag{9}$$

```
\begin{gather}
    a = b + c \\
    d = e + f + g \\
    h + i = j + k \notag \\
    l + m = n
\end{gather}
```

2.18.8 统一编号 equation

使用 aligned / gathered 环境，并且依赖 `\begin{equation} \end{equation}`，
若不须加编号则使用`\begin{equation*} \end{equation*}`，或者`\[ \]`包裹。

1. equation

$$\begin{aligned} a &= b + c \\ d &= e + f + g \\ h + i &= j + k \\ l + m &= n \end{aligned} \tag{10}$$

```
\begin{equation}
    \begin{aligned}
        a    & = b + c   \\
        d    & = e + f + g \\
        h + i & = j + k \\
        l + m & = n
    \end{aligned}
\end{equation}
```

2. equation*

$$\begin{aligned} a &= b + c \\ d &= e + f + g \\ h + i &= j + k \\ l + m &= n \end{aligned}$$

```
\begin{equation*}
    \begin{aligned}
        a    & = b + c   \\
        d    & = e + f + g \\
        h + i & = j + k \\
        l + m & = n
    \end{aligned}
\end{equation*}
```

3. \[

$$
\begin{gathered}
a = b + c \\
d = e + f + g \\
h + i = j + k \\
l + m = n
\end{gathered}
$$

```
\[
    \begin{gathered}
        a = b + c \\
        d = e + f + g \\
        h + i = j + k \\
        l + m = n
    \end{gathered}
\]
```

2.18.9 限定符

$$
\left\{
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_2 &= b_1 \\
a_{21}x_1 + a_{22}x_3 + a_{23}x_3 &= b_2
\end{aligned}
\right. \tag{11}
$$

$$
\left\{
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_2 &= b_1 \\
a_{22}x_3 + a_{23}x_3 &= b_2
\end{aligned}
\right.
$$

```
\begin{equation}
    \left\{
    \begin{gathered}
        a_{11} x_{1} + a_{12} x_{2} + a_{13}
            x_{2} = b_{1}
        \\[3pt]
        a_{21} x_{1} + a_{22} x_{3} + a_{23}
            x_{3} = b_{2}
    \end{gathered}
    \right.
\end{equation}

\begin{equation*}
    \left\{
    \begin{aligned}
        a_{11} x_{1} + a_{12} x_{2} + a_{13}
            x_{2} = b_{1}
        \\[3pt]
        a_{22} x_{3} + a_{23} x_{3} = b_{2}
    \end{aligned}
    \right.
\end{equation*}
```

2.18.10 array 环境

参考：A new implementation of LaTeX's tabular and array environment

$$
|x| = \begin{cases}
-x & \text{if } x < 0, \\
0 & \text{if } x = 0, \\
x & \text{if } x > 0.
\end{cases}
$$

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

2.18.11 case 环境

统一编号：

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{12}$$

```
\begin{equation} |x| =
   \begin{cases}
      -x & \mbox{if } x < 0, \\
      0 & \mbox{if } x = 0, \\
      x & \mbox{if } x > 0.
   \end{cases}
\end{equation}
```

分别编号：需要在导言区载入 cases 宏包 usepackagecases，并且放在 amsmath 之后。

$$|x| = \begin{cases} -x & \text{if } x < 0\ , \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{13}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{14}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{15}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{16a}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{16b}$$

$$|x| = \begin{cases} -x & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ x & \text{if } x > 0. \end{cases} \tag{16c}$$

```
\begin{numcases} {|x| =}
   -x & \mbox{if } x < 0 \label{eq:eq41},\\
   0 & \mbox{if } x = 0,\\
   x & \mbox{if } x > 0.
\end{numcases}

\begin{subnumcases} {\label{eq:eq42} |x| =}
   -x & \mbox{if } x < 0,\\
   0 & \mbox{if } x = 0\label{eq:eq43},\\
   x & \mbox{if } x > 0.
\end{subnumcases}
```

2.18.12 empheq 环境

其与之前的都不同，它可以从整个公式 (组) 四周加载定界符。从左端、从右端，甚至是用一个盒子包住整个公式 (组)。

首先在导言区载入 usepackageempheq。使用它的框架是这样:

```
\usepackage{empheq}
\begin{empheq}[markup instructions]{AMS env name}
  < content AMS environment >
\end{empheq}
```

$$\begin{cases} E = mc^2 \\ Y = \sum_{n=1}^{\infty} \frac{1}{n^2} \end{cases}$$

```
\begin{empheq}[left=\empheqlbrace]{align*}
   E & =mc^2
   \\[3pt]
   Y & = \sum_{n=1}^\infty \frac{1}{n^2}
\end{empheq}
```

2.18.13 常用数学宏 amsmath 推荐

数学公式的编辑，一定离不开这个宏。我这里找了一个关于 amsmath 的PPT 介绍。[10]

amsmath 这个宏包的作用如下。

1. 提供更多的数学符号和数学函数
2. 提供了除了 equation 之外的其他几个常用的数据环境。gather、align、flalign、alignat、multiline、subsequent。

   Split 和 gathered 、aligned、alignated 则不会进入数学环境。
3. 提供各种数学矩阵，如 matrix ， pmatrix ， bmatrix ， Bmatrix ， Vmatrix ， smallmatrix。

引用参数：

[10] 来自华东师范大学数学系潘建瑜老师的课程。

```
\usepackage[选项]{amsmath}
% 选项：其中前者是默认的参数
% centertags 或者 tabtags， split 环境种公式的编号的位置，左首右末
% sumlimits 或者 nosumlimits，求和符号上下限位置
% nointlimits 或者 inlimits，积分符号上下限位置
% namelimits 或者 nonameinlimits，积分符号上下限位置
% reqno 或者leqno。公式编号位置
% flenq， 行间公式居左对齐，默认是居中对其。
```

参考下面的代码：编号：

$$a+b=b+a \tag{17}$$
$$ab=ba \tag{18}$$

$$a\times b=b\times a$$
$$ab=ba$$

$$a+b=b+a$$
$$ab=ba$$

```
\usepackage{amsmath}
\usepackage{amssymb}%花体字符


\begin{gather}%会产生编号
   a+b=b+a\\
   ab=ba
\end{gather}

\begin{gather*}%不会产生编号
   a \times b=b \times a\\
   ab=ba
\end{gather*}

\begin{gather}%会编号
   a+b=b+a \notag \\%\notag阻止编号
   ab=ba  \notag %\notag阻止编号
\end{gather}
```

对齐：

$$x=t+\cos t+1 \tag{19}$$
$$y=2\sin t \tag{20}$$

```
   %align和align*环境（用$对齐）
   \begin{align}
      x & = t + \cos t + 1 \\
      y & = 2\sin t
   \end{align}
```

多行对齐：

$$\begin{aligned}\cos 2x &= \cos^2 x-\sin^2 x\\ &= 2\cos^2 x-1\end{aligned} \tag{21}$$

```
   %split环境（用$对齐）(一个公式分为多行排版)
   \begin{equation}
      \begin{split}
         \cos 2x &= \cos^2 x - \sin^2 x\\
         &= 2\cos^2 x - 1
      \end{split}
   \end{equation}
```

$$D(x) = \begin{cases} 1, & \text{如果} x \in \mathbb{Q} \\ 0, & \text{如果} x \in \mathbb{R} \setminus \mathbb{Q} \end{cases} \tag{22}$$

```
%case环境
%每行公式使用&分割成两部分
%通常表示值和后面的条件
\begin{equation}
    D(x) = \begin{cases}
        1, & \text{如果} x \in \mathbb{Q}
                    \\%mathbb花体字符
        0, & \text{如果} x \in \mathbb{R}\
            setminus\mathbb{Q}
    \end{cases}%\text是为了在数学公式中处理中文
\end{equation}
```

2.19 如何插入代码？

就像 markdown 一样，

1. 可以通过简单的`code` 标记来标识行内代码。
2. 通过进行多行显示原始代码。

```
code
code
```

2.19.1 简单代码 (适合非常简单的原始代码)

LaTeX 可使用原始显示的方式进行快速标识。虽然效果看不出来是行内代码，但是效果还是不错的，主要是字体稍微有点差别：这是code~。

1. 使用代码的 \verb!code!，进行简单的标识。但是中间不要太多“!”号
2. 使用 verbatim

```
%this is code
%very simple code
```

```
\begin{verbatim}
%this is code
%very simple code
\end{verbatim}
```

2.19.2 初级的多行代码

1. usepackage Listings：\usepackage{listingsutf8}
2. 设置代码的格式

```
%*******************代码设置*****************************************
% 用来设置附录中代码的样式

% step 1： 引用颜色宏
\usepackage{xcolor}

% step 2： 设置宏的参数
\lstset{
    backgroundcolor = \color[RGB]{240,240,240},
    breaklines      = true,                 % 自动换行
```

```
    numbers         = left,                 % 行号的位置在左边
    basicstyle      = \zihao{-5}\ttfamily, % sf/tt基本代码风格
    frame           = LtBr,                    % 边框。lrbt，大写的则是双边框
    framesep        = 1pt,                  % 代码和边框之间的margin
    rulesep         = 4pt, %双框距离
}
```

3. 在指定的位置插入例子

```
\begin{lstlisting}[language=C,frame=single]
int main() {
    printf("hello, world");
    return 0;
}
\end{lstlisting}
```

4. 可以看到有行号，有方框，有背景颜色。(看到下面的框框是单框的, 是因为设置了 ***frame=single***)

```
int main() {
    printf("hello, world");
    return 0;
}
```

当然，大家看到的本文很多代码是双边框的。为什么呢？因为有一个设置是 frame ，有效值是 lrtb 和 LRTB，single 等。大写则为双边框。大家还记得 1.8.2 吗？可选参数是如何配置的？可选参数是 [] 方括号，即 `\begin{lstlisting}`后面可以写`[frame = xxxx]`。

2.19.3 解决 listing 嵌套的问题

参考链接：解决 listing 嵌套的问题

实现原理

如果使用两个 lstlisting 环境嵌套，会使得第一个 `\begin{lstlisting}` 匹配到第一个 `\end{lstlisting}` 而使得第二个报错。比较好的解决方法就是把外层的摘录环境换成 verbatim。

~~不过由于个人比较喜欢 listings 包的风格，所以~~这里采用 tcolorbox 包的 listings 库提供的 tcblisting 环境。tcblisting 环境其实就是调用的 listings 包，它一方面接受 listings 包的设置，另一方面还接受 tcolorbox 包的设置，简直一举两得。默认情况下，摘录的同时还执行摘录的代码[11]，所以这里添加了 *listing only* 选项，仅仅作摘录使用。[12]

实现步骤

宏包引用 tcolorbox 代码:

```
    \usepackage[listings]{tcolorbox}
    % 或者 \tcbuselibrary{listings}
```

[11] 就是说会同时把代码也渲染出来当作例子

[12] https://yuxtech.club/2019/07/30/listings/

使用 tcblisting 包围代码

```
\begin{tcblisting}{listing only,boxrule=-1pt,colback=white}
    \begin{lstlisting}[language=Fortran,frame=single]
int main() {
    printf("hello, world");
    return 0;
}
    \end{lstlisting}
\end{tcblisting}
```

这里需要注意的是，一定要使用 *listling only* 参数。同时为了防止边框重叠。添加一些边框参数。总体要添加 `{listing only,boxrule=-1pt,colback=white}`。显示效果参考 3

2.20 如何在中插入图片？

2.20.1 行内图片

一般情况下，很少在行内插入图片。行内插入图片会和文字混 

合在一起。

上面的代码：

```
行内插入图片会和文字混
\includegraphics[width=5cm]{images/pic_inline.png}
合在一起。
```

2.20.2 图片环境

插入图片是比较麻烦的时候。不像 Word 文档那样所见所得。更多更详细的可以参考 CTex 链接：LaTeX 2ε 插图指南

另外一个参考链接：LaTeX 中插图总结

1. 插入代码，参考代码

```
\begin{figure}[H] %H为当前位置，!htb为忽略美学标准，htbp为浮动图形
    \centering %图片居中
    % \setlength{\fboxrule}{1pt}
    % \setlength{\fboxsep}{0.5cm}
    \fbox{
        \includegraphics[]{images/insert-png.png} %插入图片，[]中设置图片大小，{}中是图片文件名
    }
    \caption{插入图片的demo} %最终文档中希望显示的图片标题
    \label{Fig.insert-one-png} %用于文内引用的标签
\end{figure}
```

2. 最终效果，图2.1

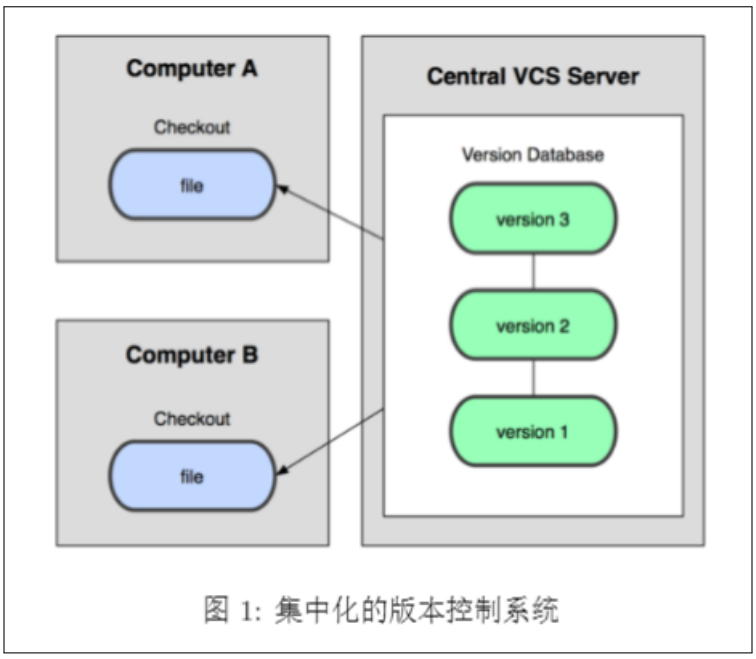


图 2.1: 插入图片的 demo

3. 双排设置

```
Figure \ref{Fig.main} has two sub figures, fig. \ref{Fig.sub.1} is the travel demand of
    driving auto, and fig. \ref{Fig.sub.2} is the travel demand of park-and-ride.

\begin{figure}[H]
\centering  %图片全局居中
\subfigure[name1]
{
    \label{Fig.sub.1}
    \includegraphics[width=0.45\textwidth]{DV_demand}
}
\subfigure[name2]
{
    \label{Fig.sub.2}
    \includegraphics[width=0.45\textwidth]{P+R_demand}
}
\caption{Main name}
\label{Fig.main}
\end{figure}
```

4. 双排效果显示，图 2.2

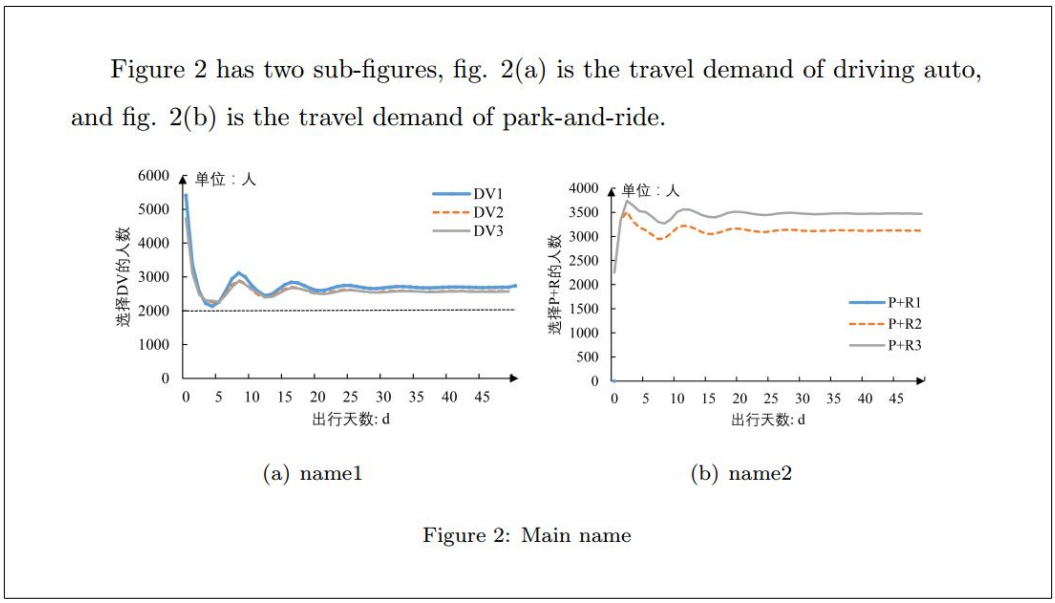


图 2.2: 并排插入两个图片的效果

5. 引用图片

   在文档中使用 \ref{标签}，如图`\ref{Fig.Insert_two_pic_demo}` 显示为：图2.2。

6. begin{figure}[H] 这个 H 参数，可以设置为以下几个，H 为当前位置，!htb 为忽略美学标准，htbp 为浮动图形。这个参数具体是怎么回事，什么原理，请看章节 2.23 讲解。

2.20.3 图片过大如何超过边距进行居中？

参考链接：LaTeX 入门 (八)——图片

有时候我们的图片会显示的过大。但是图片会按照左边距进行开始排版，这会导致排版偏右，甚至超出界面。

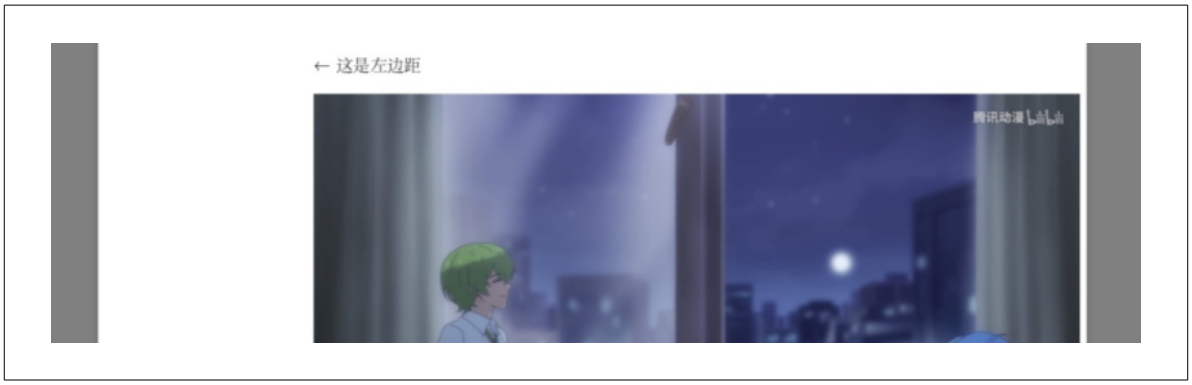


图 2.3: 图片过大超出版面

图 2.3 是一个图片超出缩进范围的渲染图，导致图片偏向页面右边。参考图1.5，我们可以看到这个图占用比较多的空间，但居中了，看起来会好很多。

要想解决这个问题，可以使用 adjustbox 解决。首先需要在导言区域添加宏包引用。

```
% 宏包
\usepackage[export]{adjustbox}
```

之后，在图片的代码中添加 **center**。

```
% 将 \includegraphics[xxx]{pic.png}
% 改成:
\includegraphics[center,xxxx]{pic.png}
```

2.20.4 如何设置图片缩放等于行宽?

参考: LaTeX 中的浮动体：处理超宽问题

图2.1，我们可以看到他只是占用了一部分。

假如我想让他占用整个列宽，可以把大小的设置设置为行宽。还可以设置为行宽的倍数。

```
% 也可以设置为行宽的倍数。比如[width = 0.2\linewidth]代表只有行的0.2行宽
\includegraphics[width = 0.2\linewidth]{pic.png}
% 设置宽度为行宽
\includegraphics[width = \linewidth]{pic.png}
```

重新排版后的图2.1：

只有行的 0.2 行宽的图片：

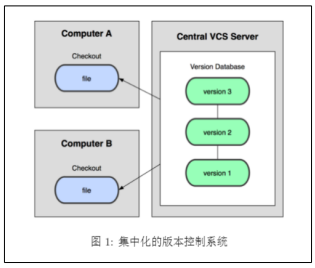


图 2.4: 图片缩放 0.2 行宽

下图是刚好占用行宽的图片效果，可以看出，整个图片占用整个段落的行宽:

图 2.5: 图片缩放等于行宽

2.21 如何插入表格?

参考连接: https://blog.csdn.net/yhl_leo/article/details/50066137

tabular 环境提供了最简单的表格功能。它用\hline命令表示横线，| 表示竖线；用 & 来分列，用\\来换行；每列可以采用居中、居左、居右等横向对齐方式，分别用 l、c、r 来表示。

一般情况下，tabular 会嵌入到 table 的环境中。

table 环境和 figure 环境类似。也是提供一个浮动的 box 供给 tabular 进行绘制。

2.21.1 操作步骤

使用以下的代码。显示效果参考表2.5

```
\begin{table}[H]
    \centering
    \begin{tabular}{|c|lr|} % 这里三个字母clr 代表三列。竖线代表画出竖边框
        \hline
        操作系统    & 发行版    & 编辑器     \\
        \hline
        Windows     & MikTeX    & TexMakerX \\
        Unix/Linux & teTeX     & Kile       \\
        Mac OS      & MacTeX    & TeXShop    \\
        通用        & TeX Live & TeXworks   \\
        \hline
    \end{tabular}
    \qquad
    不同的操作系统和编辑器之间很多是通用的。
    推荐使用 TeX Live.
    \caption{不同操作系统对应的发行版和编辑器}
    \label{Table.Demo0}
\end{table}
```

2.21.2 效果显示

显示效果:

操作系统	发行版	编辑器
Windows	MikTeX	TexMakerX
Unix/Linux	teTeX	Kile
Mac OS	MacTeX	TeXShop
通用	TeX Live	TeXworks

不同的操作系统和编辑器之间很多是通用的。

推荐使用 TeX Live.

表 2.5: 不同操作系统对应的发行版和编辑器

当然这样写表格肯定很麻烦。也可以参考链接1.7.1 的在线表格生成器。

Tips:

表格和图一样是浮动环境。所以图片要在 figure 里面引用。而 table 会浮动到其他地去。在\begin{table}后面加入 [H]，代表不允许浮动，其中 [H] 是 float 宏包提供的。(注意请导入:\usepackage{float})参考章节 2.23。

2.21.3 居中

有时候，用 \begin{center}-\end{center} 来将 table 整体居中，但它的居中只是居中了\begin{table}-\end{table} 这个环境（把 table 想象成为盒子，相当于居中盒子，但不是将盒子里面的内容居中）;

如果\begin{table}-\end{table} 里面嵌套了 \begin{tabular} - \end{tabular} ，那么表格还是不会居中。要使得表格居中，则需要在 \begin{table} 后面附加 \centering，才能使得表格整体居中。

2.21.4 合并表格如何设置？

属于高级技巧，参考 3.7。

2.22 如何设置边距？

2.22.1 操作步骤

1. 引用 geo 设置宏包 \usepackage{geometry}
2. 设置页边距
   (a) 设置纸张大小，也可以设置常用的 a4paper 之类的。
   ```
   \geometry{papersize={20cm,15cm}}
   ```
   (b) 设置为缩放的形式
   ```
   \geometry{a4paper,scale=0.8}
   ```
   (c) 设置为绝对距离
   ```
   \geometry{a4paper,left=2cm,right=2cm,top=2cm,bottom=2cm}
   ```
3. 效果不错

2.22.2 页面边距设置概念

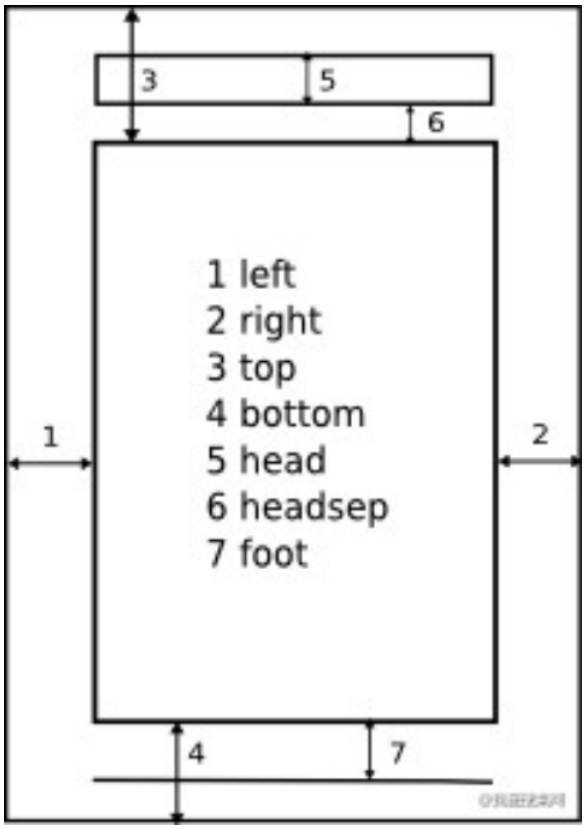


图 2.6: 页边距概念

2.23 显示盒（box）的浮动美学设置

很明显，LaTeX 把很多内容当作一个 Box 进行显示，也是就是划定一个区域（Box）出来，你可以在这个区域里面进行绘制，但是里面如何绘制是这个盒子的事情，那么这个盒子又要放在哪里呢？

参数	内容
h	当前位置。放置在正文文本中给出该图形环境的地方。如果本页所剩的页面不够，此参数不起作用。
t	顶部。将图形放置在页面的顶部。
b	底部。将图形放置在页面的底部 16.1。
p	浮动页。将图形放置在一只允许有浮动对象的页面上。

表 2.6: 浮动的位置参数

我们可以图片和表格等内容当作为一个 Box，那么我们就可以按照 LaTeX 对盒子的美学排版进行设置这个盒子的配置。看表 2.6，你会很好奇，为什么这个表跑到上面去了，因为我在绘制表的时候，没有设置参数，那么就是默认的 [tbp]，也就是顶部，所以他跑到顶部去了。

参考 CTex 链接：图形的放置

我是在这里放置表 2.6的。但是他在本页的最上面显示了。

默认情况下是 tbp, 同时需要注意的是，LaTeX 总是尝试以 **h-t-b-p** 的顺序来确定图形的位置。当然，加一个“!”（感叹号），可以强制按照指定的顺序来执行。如“!bpt”则会先考虑 b，在考虑 p，之后考虑 t，而不默认的 htbp。

3 LaTeX 高级技巧

3.1 我们如何知道需要引用什么宏呢？

当我们需要用什么东西的时候，我们该如何知道要引用什么？比如我们要用超链接，我们就需要引用 hyperref 这个宏包？那我该如何知道呢？

知乎讨论: 用 TeX 编辑论文时，如何选择合适的 Packages ？

CTAN（Comprehensive TeX Archive Network）是干什么的，各位从它的名字应该能看出来。一般来说，如果有作者写了一个宏包，自觉还挺有价值，测试之后与常用宏包没有发现冲突，就会考虑上传到 CTAN 这个网站。CTAN 网站经过收集、整合、分类，给广大 TeX 用户提供了方便地检索系统：CTAN: Search。具体的用法可以参见页面当中的 Search Tips。

3.2 如何对宏进行配置？

当我们引用一个宏的时候，我们如何知道怎么配置这个宏呢？比如我们引用 hyperref 宏的时候。我是如何知道我可以配置一下的这些东西呢？

```
\hypersetup{
    colorlinks  = true,
    linkcolor   = blue,
    filecolor   = magenta,
    urlcolor    = cyan,
    % pdftitle    = {Overleaf Example},
    % pdfpagemode = FullScreen,
}
```

3.3 如何设置多栏并列显示？

参考：latex 设置多栏正文

很多科技文章，整体都是双栏进行显示的。这样做出来的效果会比较专业和上档次。

为什么会这样？看过这类文章之后，你会发现排版比较密集，内容密度高，信息关联清晰。那么该如何设置这个呢？

3.3.1 模板

一般情况下，会在 document 环境里面，套一层分栏来进行全局的分栏设置。

首先需要引用宏 `\usepackage{multicol}`，之后我们就可以使用 multicol 的环境了。此环境有必要参数，默认需要传入一个分栏的个数，如需要分两栏就传入 2，需要分三栏，就传入 3。如下面的例子：

```
\documentclass[UTF8]{ctexart}
\usepackage{multicol}

%文档
\begin{document}

```

```
    % 开始全局的分栏模式。
    \begin{multicols}{2}
    正文。
    \end{multicols}

\end{document}
```

3.3.2 文内

如果不是全局的分栏，也可以在指定的地方分栏，使用方法和上面的一模一样。

这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，**这是大概是第二栏的内容**，这是第二栏的内容，这是第二栏的内容，

这是第二栏里面的分栏。栏内栏。栏内栏。栏内栏。栏内栏。栏内栏。

这是第二栏的内容，这是第二栏的内容，这是第二栏的内容，

```
\begin{multicols}{2}
这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，这是第一栏的内容，\textbf{这是大概是第二栏的内容}，这是第二栏的内容，这是第二栏的内容，
\begin{multicols} {2}
这是第二栏里面的分栏。栏内栏。栏内栏。栏内栏。栏内栏。栏内栏。
\end{multicols}
这是第二栏的内容，
这是第二栏的内容，
这是第二栏的内容，
\end{multicols}
```

可以看到，通过这个方式，我们可以将这个任务栏分成两栏。当内容超过一页的时候，他会在下一页继续分栏。另外还可以在分栏里面继续分栏。

3.3.3 按照指定的位置分栏

当我们想在指定的页面分栏的时候，我们可以添加标记 \columnbreak 强制分栏。这样，标记之前的就会占满左边的页面，即使左边的内容或者右边的内容没有占满页面，也会进行分栏处理。那么可以在指定的位置添加 \columnbreak。

现在我添加了\columnbreak，现在已经换栏了，而且这里的内容比左边少。

现在我又添加了\columnbreak。**需要注意的是，分栏一定是要在段落之间，而不能在段落内，否则无法在指定的位置分栏。**这里是第三栏。内容很少哦。

3.3.4 分栏的一些设置

分栏的间距

分栏之间的间距可以通过以下进行配置。

```
% 设置分栏之间的宽度。
\setlength{\columnsep}{0.5cm}
```

分栏分割线宽度

分栏线的宽度可以通过以下进行配置。

```
% 设置分割线的宽度
\setlength{\columnseprule
    }{0.5pt}
```

分栏分割线的颜色

分栏线的颜色可以通过以下进行配置。

```
%定义分割线的颜色
\def\columnseprulecolor{\
    color{gray}}
```

3.4 双栏模板下插入通栏公式和图片等

参考：双栏模板下插入通栏公式和图片

3.5 如何添加自定义类型目录?

参考：在 LaTeX 中自定义新目录

在 LaTeX 中，有三个命令\tableofcontents，\listoftables，\listoffigures，别用来生成章节目录，表格目录和插图目录。但有时候我们需要生成其他目录，比如书籍中的例题目录，这个时候我们可以利用 tocloft 包自定义一个例题目录。tocloft 包提供了\newlistof 命令，其格式如下：

```
1    \newlistof[已有计数器count1]{计数器count2}{扩展名}{目录名称}
```

3.6 如何自定义系列标头?

编号是需要计数器的，在标准的计数器中，只有\roman、\Roman、\arabic、\alph、\Alph以及\fnsymbol，其输出格式分别为 i、ii、I、II、1、2、a、b、A、B、花体符号，现在如果需要输出一、二这种类型的编号，可以使用 enumitem 包提供的\AddEnumerateCounter命令，同时利用 ctex 包提供的\chinese 命令即可，一个 minimal working example 如下[13]：

```
1    \documentclass{ctexart}
2    \usepackage[showframe]{geometry}
3    \usepackage{enumitem}
4    \AddEnumerateCounter{\chinese}{\chinese}{}
5
6    % \begin{document}
7    \begin{enumerate}[label = \chinese*、]
8        \item 第一行
9        \item 第二行
10   \end{enumerate}
11   % \end{document}
```

一、 第一行

二、 第二行

中文嵌套效果，可以看2.16.2。

3.7 如何合并表格的单元格?

参考连接：https://blog.csdn.net/u011439796/article/details/78539678

参考连接：https://blog.csdn.net/yhl_leo/article/details/50066137

合并单元格也是比较常见的。这里简单的介绍一下横项合并，纵向合并和多行多列合并。

先看看下面的例子，我们可以看到，我们有两行一列的合并，有一行两列的合并，有两行两列的合并。感觉看起来非常好麻烦。

[13] https://yuxtech.club/2021/01/18/enumerate/

```
\begin{table}
   \centering
   \begin{tabular}{|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}
      \hline
      I & II & III & IIII \\
      \hline
      \multirow{2}*{合并两行一列} & a & 三 & 四 \\
      \cline{2-4}
      ~ & b & \multicolumn{2}{|c|}{合并两列} \\
      \hline
      1 & \multicolumn{2}{|c|}{\multirow{2}*{合并两行两列}} & 4 \\
      \cline{1-1}    \cline{4-4} % 这里进行格列划线。
      1 & \multicolumn{2}{|c|}{~} & 4 \\
      \hline
   \end{tabular}
   % \caption[]{合并单元格和划线}
   \label{tab:combin_cell_of_table}
\end{table}
```

I	II	III	IIII
合并两行一列	a	三	四
	b	合并两列	
1	合并两行两列		4
1			4

3.7.1 纵向多行合并

纵向多行合并，我们使用 multirow 进行合并。

当我们需要占用几行的时候。我们就用这个 multirow 先占用合并的首行，然后其他行用“~”替代。如下面的代码：

```
1 \multirow{3}*{合并三行}&....\\
2 ~                      &....\
3 ~                      &....\
```

3.7.2 横向多列合并

表格们是使用 & 进行分列的。当我们需要占用多列的时候。我们可以用 multicolumn 占用要合并的列。就不用在添加 & 号了。如下面的代码：

```
1 \multicolumn{3}*{合并三列} &....\\
2 1    &    2    &    3     &....\\
3 1    &    2    &    3     &....\\
```

3.7.3 多行多列合并

同理，我们需要多行列的时候，我们可以先用 multicolumn 占用要合并的列。同时嵌套 multirow ，占用需要合并的行，同样，其他行用“~”替代。当然，因为这里 multirow 占用了两行，所以。第二行，还是要把 multicolumn 加上，同时里面的内容为“~”。如下面的代码：

```
\multicolumn{3}*{ \multirow{2}*{合并两行} } &....\\
\multicolumn{3}*{          ~          }     &....\\
1         &            2         &      3    &....\\
```

Tips: 这里在第二行采用 multicolumn 来进行空白占位，这样可以避免一些奇怪的划线行为，如果直接采用 ~ & ~ & ... 的方式来占位，会受到表格划线方式 {|c|c|c|c|} 的影响而多划一条竖线。

3.7.4 多行合并后的横线 hline 问题

行合并之后，我们的划线就不能直接穿越合并的单元格了，所以是用 hline 会在合并的单元格添加横向线条，我们可以用 cline 进行替代。

cline 支持传入列的范围进行划线。如 `\cline{1-2}`，代表第一列和第二列单元格进行划线。`\cline{5-5}` 代表只划第五个列。这样连起来，就相当于不画第三个列和第四个列。

如下面的代码，显示效果参考 **??**

```
1 & \multicolumn{2}{|c|}{\multirow{2}*{合并两行两列}} & 4 \\
\cline{1-1}        \cline{4-4}  % 这里进行格列划线。
1 & \multicolumn{2}{|c|}{~}                          & 4 \\
```

3.8 如何在表头单元格里面添加斜线？

如何在单元格里面添加斜线？如何分割单元格？我想很多使用 Excel 或者 Word 的人都不太会。结果我们在这里使用 LaTeX 啃这个硬骨头。废话不多说，我们来看看代码和效果：

```
\usepackage{diagbox}
\begin{table}
    \centering
    \begin{tabular}{|l|c|c|c|}
        \hline
        \diagbox{Time}{Room}{Day} & Mon & Tue & Wed \\
        \hline
        Morning & used & used & \\
        \hline
        Afternoon & & used & used \\
        \hline
    \end{tabular}
    \caption{表头分格，画斜线}
    \label{tab:div_cell}
\end{table}
```

Room \ Day / Time	Mon	Tue	Wed
Morning	used	used	
Afternoon		used	used

表 3.1: 表头分格，画斜线

我们使用宏 diagbox 进行分割。只要用 diagbox 占用某个单元格即可。分割几个，我们就传入几个表头。顺序按照顺时针进行添加。如实例 3.1 中的 `\diagbox{Time}{Room}{Day}`。

参考文献

[1] 刘海洋. LaTeX 入门. 电子工业出版社, 2013-6.