

L^AT_EX 学习报告

Shiyuu

2023 年 4 月 12 日

摘要

这一篇文档算不上是什么报告, 其实只是拿浅薄的积累来练个手 (毕生所学), 以及把平常用的 T_EX 方法列个表出来方便查阅。最重要的还是把从数院学长那里薅来的我根本搞不懂的 T_EX 模板做个参考, 免得以后 Compile 了一大堆错误都不知道为什么。

现阶段没有太多时间去学 T_EX 的高端用法真是抱歉, 反正能记笔记、交作业就行, 后面有时间再补充。如果不耐烦了可以直接跳到??介绍部分喵。

目录

1 文本

那既然 \LaTeX 是用来写字的，那么文本编辑好像也没什么好说的，直接往 `\article` 里面敲字就行了。一篇最普通的文章模板（例如 `artical`、`ctexart`）理论上就只需要

```
1      \begin{dOCUMENT}  
2          .....  
3      \end{document}
```

这一段环境就可以了，在省略号省略的部分，直接往里敲文字就可以。

当然，只有文章题目是不够的，我们还需要更多的一级、二级、三级标题。只需要下面这几个命令（不用 `end`）

```
1      \section{Sec:A}  
2          \subsection{Sec:A-1}  
3              \subsubsection{Sec:A-1.1}  
4                  \sunsubsection*{notitle}
```

以此类推， \TeX 会自动帮你编号，`{}` 中的就是章节名称，一般情况下三级标题也够用了，如果还要加的话可以在里面内嵌 `\box`、`\framed` 之类的环境，然后使用 `\label` 和 `\ref` 交叉引用。

前面还有模板之类的问题，其实对于一二级标题的序号样式（`1`→`1.1`→`1.1.1`），字体，页码页眉之类的东西都可以自己打开模板文件调整，但是我没有那个闲工夫（悲），后面会尽量补上的。

关于转义。 \TeX 本质上是编译出来的，受指令关键字限制，一些字符或者排版意图就必须以转义的方式体现。例如，输入半个制表位要用指令 `\quad`，输入一个制表位则用 `\qqquad`，因为 \TeX 在字符连续时忽略空白符，在编辑时需要两个连续的“`\n`”换行符才能让文本另起一段，要么就直接使用 `\par` 指令。而如果在新的一段不想缩进则可以在前面加 `\\` 或者 `\noindent`，以及我在写这篇文档时疯狂复制粘贴的 `\textbackslash`。考虑以下的代码：

```
1      你说的对，  
2  
3      但是 \quad\TeX 是一款专门用于 \\ 文章
```

```

4      书籍排版的解释性语言
5
6      {\noindent} 玩家将在编译错误、红黄乱闪和频繁报错的编辑器中
7      逐步发掘“学术”的真相 {\par}6

```

注意在正文部分遇到需要转义输入的文本，要用 `{}` 括起来，否则 $\text{T}_{\text{E}}\text{X}$ 会认为反斜杠后面的都是转义的字符或者说指令。上面代码的效果如下

```

    你说的对，
    但是  $\text{T}_{\text{E}}\text{X}$  是一款专门用于
文章书籍排版的解释性语言
玩家将在编译错误、红黄乱闪和频繁报错的编辑器中逐步发掘“学术”的真相
6

```

好消息是对于转义字符，编辑器都有优化，例如我用的 VSCode 的 Latex Workshop 插件就在左边栏上加载了快速输入脚本，在后面输入数学公式和表格时也会相对方便。也可以通过各种公式生成器来生成复杂的 $\text{T}_{\text{E}}\text{X}$ 代码。对于一些常见的转义字符，详见速查手册。

2 环境

环境是 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 中很重要的一个抽象概念，它开始于一个 `\begin{}`，结束于一个 `\end{}`，其中间括起来的部分是环境的内容。环境的用法基于这样几个特性：

- 环境可以对内容实现模板化的处理。也即满足某种属性；
- 环境需要特定的格式，满足特定的条件。
- 环境可以相互嵌套，并可能产生相互影响。

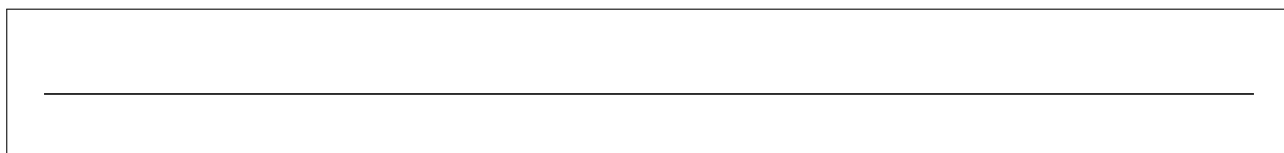
考虑以下例子：

```

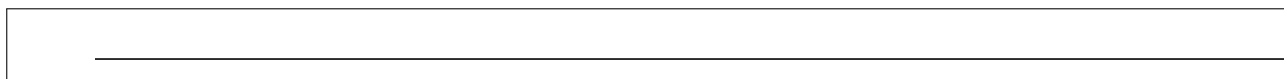
1      \begin{center}
2          \rule[-10pt]{16cm}{0.05em}
3      \end{center}%一条不错的加横线指令

```

这个环境中含有一个`\rule` 指令，创建了一条直线，效果如下：



但是，如果不将这条指令放在 `center` 环境中，就会变成这样：



这说明 `center` 环境起了将环境中的内容置于页面中间的作用。

另外，其实这个加直线的指令并不会带一个框，我在做这篇文档时的完整代码是：

```
1    \begin{framed}
2    \begin{center}
3        \rule[-10pt]{16cm}{0.05em}
4    \end{center}%一条不错的加横线指令
5    \end{framed}
```

哦，还有展示这段代码的代码：

```
1    \begin{framed}
2    \begin{lstlisting}[language=tex]
3        \begin{framed}
4        \begin{center}
5            \rule[-10pt]{16cm}{0.05em}
6        \end{center}%一条不错的加横线指令
7        \end{framed}
8    \end{lstlisting}
9    \end{framed}
10   %最后一行\后面加了空格
11   %否则编译不出来
12   %因为这里嵌套了两层 lstlisting
```

..... 开始套娃.....

上面已经用到了无序列表环境，与之对应的是有序列表：

```

1      \begin{multicols}{2}%本例还使用了多栏环境
2      \columnseprule 1pt   %中央分割线宽
3      \columnsep 35pt      %控制两栏之间间隔
4      无序列表
5      \begin{itemize}
6          \item 我是SB
7          \item SB is me.
8          \item Sb is xjtu.
9      \end{itemize}
10     有序列表
11     \begin{enumerate}
12         \item 主E20楼
13         \item 一跃解千愁{\label{sb}}
14     \end{enumerate}
15     \end{multicols}

```

效果如下：

无序列表

- 我是 SB
- SB is me.
- Sb is xjtu.

有序列表

1. 主 E20 楼
2. 一跃解千愁

另外比较常用的就是插入图片的环境，这通过 `graphicx` 宏包实现：

```

1      \usepackage{graphicx}
2      ...
3      \begin{figure}[htbp]%似乎 htbp 每个字母都有深意..
4          \centering
5          \includegraphics[width=0.65\textwidth]{../img/huoxing.png}

```

```

6      %文件路径
7      \caption*{火星上的伪卫星阵列}
8      \label{fig:sb}      %（伏笔
9      \end{figure}

```

效果如下：

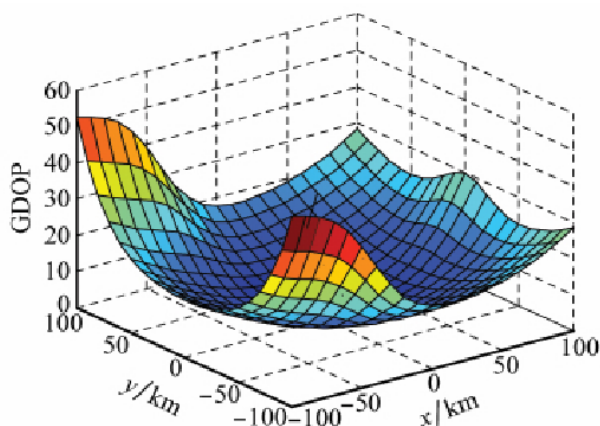


图 1: 某仿真图像

注意一下代码里的`\label{fig:sb}`，后面引用的时候来个前后呼应。

我们还可以自定义环境，用法是

```

1      \newenvironment{新环境名称}[参数个数][参数默认值]
2      {开始部分定义}{结束部分定义}

```

有点像 C/C++ 中的宏，新环境的开始部分和结束部分定义的内容在编译时被替换。如果要做一个“摘要”的新环境，可以这么设置并调用：

```

1      \newenvironment{Abstract}
2      {
3          \begin{center}\normalfont\bfseries Abstract\end{center}
4          \begin{quote}\par
5      }
6      {\end{quote}}

```

```

7      .....
8      \begin{Abstract}
9          This is an abstract Abstract.
10     \end{Abstract}

```

效果：（框是加的）

Abstract

This is an abstract Abstract.

再康康数院学长做的习题环境：

```

1      \definecolor{shadecolor}{RGB}{222, 227, 230}
2      \newcounter{problemname}%计数器，用来标号
3      \newenvironment{problem}
4      {\begin{shaded}\stepcounter{problemname}\par\noindent
5          \textbf{FRAME-\arabic{problemname}. }}
6      {\end{shaded}\par}
7      %shaded是framed包自带的底色环境

```

效果：

FRAME-1. 已有 3 个学生的 3 门课成绩，分别用函数实现以下功能：

- 1) 计算每个学生的总成绩
- 2) 按照学生总成绩从高到低进行排序

要求：

1) 在 main 函数中分别调用以上函数，按照学生三门课程总成绩从大到小输出学生的相关信息。

2) 函数自行定义。三个学生的信息按照如下直接赋值：

1001,11,zhang,99.5,88.5,89.5

```
1002,22,li,77.9,56.5,87.5
1003,11,wang,92.5,99.0,60.5
```

3 代码

当然对我来说， $\text{T}_{\text{E}}\text{X}$ 主要是用来记录 CS 学习的，所以不可避免地要插入很多代码。在 $\text{T}_{\text{E}}\text{X}$ 中，使用 `lstlisting` 宏包来进行代码插入。这个宏包通过 `lstset` 来设置样式，支持 100 多种计算机语言的关键字高亮，还有各种背景色的样式设置等等。上面的代码中，我只简单设置了一下样式，边框则是通过 `framed` 宏包添加的。

来试试看自己调一个：

```
1 \lstset
2 {%
3     language=C++, %语言种类
4     frame=shadowbox, %边框预设
5     xleftmargin=1.5em, %整体距左侧边线的距离为2em
6     rulesepcolor=\color{lightgray}, %阴影颜色
7     rulecolor=\color{black}, %框架颜色设置
8     aboveskip=30pt, %与整个代码环境上面距离
9     belowskip=10pt, %*下面
10    breaklines=true, %自动换行
11    basicstyle=\tt, %使用等宽字体，可调
12    keywordstyle=\bfseries\color{green!40!black}, %设置关键字
        颜色为绿色，转变成bold加粗序列
13    commentstyle=\itshape\color{purple!40!black}, %注释颜色设
        置为灰色
14    identifierstyle=\color{blue}, %标识符设置为蓝色
15    stringstyle=\color{orange}, %字符串设置为橙色
16    showstringspaces=false, %去掉空格时产生的下划的空格标志，
        设置为true则出现
17    numbers=left, %在左侧显示行数
18    numberstyle=\tiny\color{red}, %数字大小，颜色调整
```



```
19     numberstyle=\it, %数字字体设为罗马斜体
20     stepnumber=1 , %每行标号一次
21     numbersep=10pt, %数字水平距离代码10pt
22     tabsize=4, %制表符长度
23     framexleftmargin=0mm, %框架左边界延长
24     framexrightmargin=-1mm, %框架右边界延长
25     columns=fixed, %紧凑排列
26 }
27
28 \begin{framed}
29 \begin{lstlisting}[language=C++]
30     #include <stdio.h>
31
32     int main(void)
33     {
34         int* nums;
35         scanf("%d", nums);
36     }
37 \end{lstlisting}
38 \end{framed}
```

输出:

```
1     #include <stdio.h>
2
3     int main(void)
4     {
5         int* nums;
6         scanf("%d", nums);
7     }
```

2023.4.1 凌晨, 调了 3 个小时。这下小丑了。

4 公式

公式是 \LaTeX 中的一种特殊环境，分为行内和行间公式两种。利用 \LaTeX ，我们可以方便地对公式进行排版、对齐和交叉引用，唯一的不足是编辑起来实在是过于麻烦了，而且可能出现编译性的错误。好在 Code 给我们提供了一些脚本，也可以用前面提到的公式生成器。

行内公式的样式与行间略有不同。输入 $\text{\textbackslash lim}_{n\rightarrow\infty}$ 和 $\text{\textbackslash prod}_{n=1}^{\infty}$ ，效果如下：

行内公式： $\lim_{n\rightarrow\infty} \prod_{n=1}^{\infty}$

行间公式：
$$\lim_{n\rightarrow\infty} \prod_{n=1}^{\infty}$$

如果想让行内公式具有行间公式的样式，则加上 $\text{\textbackslash limits}$ 或 $\text{\textbackslash displaystyle}$ ，如

```
1 \lim\limits_{n\to\infty}
2 {\displaystyle\sum_{n=1}^{\infty}}
```

效果： $\lim_{n\rightarrow\infty} \sum_{n=1}^{\infty}$

关于括号的使用，直接使用 $()$, $[]$, $\{\}$ ，括号的高度不会随着括号中的内容高度而变化，如 $(\frac{3}{114})^2$ 、 $[\frac{1919^2}{810}]$ ，所以使用 $\text{\textbackslash left}(\dots\text{\textbackslash right})$ ，效果： $(\frac{3}{114})^2$ 。注意 $\text{\textbackslash left}$ (和 $\text{\textbackslash right}$) 必须成对出现，如果不想显示某一边则要把括号改成小数点，写成 $\text{\textbackslash left}[\dots\text{\textbackslash right}.$ 。效果： $[\frac{1919^2}{810}]$

行内公式调用如 $E = mc^2$ 只需要 $\text{\$}E=mc^2\text{\$}$ 即可。

行间公式调用有两种方法：

1. 使用 $\text{\textbackslash [...]}$ 或者 $\text{\$}\dots\text{\$}$ 这两种方法都只能输入单行公式， \textbackslash\ 在其中失效。

2. 使用环境，如：

- align
- alignat
- flalign
- equation
- gather

- multiline

使用环境公式的好处是可以自动编号并方便排版，打上 label 后也方便交叉引用。所有上面的环境都可以在环境名后面加 * 表示无序号。

先以 align 为例：

```

1 \begin{align}
2     &\ x^4+2x^3+11x^2+18x+18 \ \backslash\backslash
3     = &(x^2+2x+2)(x^2+9) \ \backslash\text{notag}\backslash\backslash
4     = &(x^2+x+3)^2+(2x+3)^2
5 \end{align}

```

环境中以 \$ 为对称处标识符，整体中心对称 $\backslash\backslash$ 表示换行，用 $\backslash\text{notag}$ 或 $\backslash\text{nonumber}$ 来隐藏任意一行公式的编号。效果：

$$x^4 + 2x^3 + 11x^2 + 18x + 18 \tag{1}$$

$$= (x^2 + 2x + 2)(x^2 + 9)$$

$$= (x^2 + x + 3)^2 + (2x + 3)^2 \tag{2}$$

equation 环境其实也只能插入一行公式，但是可以用 split 嵌套，好处是多行公式只显示一个编号。

```

1 \begin{equation}
2     \label{eq1}
3     \begin{split}
4         &\ x^4+2x^3+11x^2+18x+18 \ \backslash\backslash
5         = &(x^2+2x+2)(x^2+9) \ \backslash\backslash
6         = &(x^2+x+3)^2+(2x+3)^2
7     \end{split}
8 \end{equation}

```

效果：

$$x^4 + 2x^3 + 11x^2 + 18x + 18$$

$$= (x^2 + 2x + 2)(x^2 + 9) \tag{3}$$

$$= (x^2 + x + 3)^2 + (2x + 3)^2$$

除此之外，align 和 align 环境没有什么区别；gather 环境中不能出现对齐符号 &，公式全部居中。case 环境用于联立方程：

```

1      \begin{align*}
2          \boxed{ %给公式加边框
3              \begin{cases}
4                  2x + 3y = 7 \\
5                  3x + 5y = 8
6              \end{cases}
7          }
8      \end{align*}

```

效果：

$$\begin{cases} 2x + 3y = 7 \\ 3x + 5y = 8 \end{cases}$$

还有一种 multiline 环境，用的较少。

```

1      numberwithin{equation}{section} %编号带上section的序号
2      \begin{multiline}
3          \text{字字字字字字} \\ %汉字必须这样括起来
4          1-line \\
5          2-line \\
6          3-line \\
7          4-line
8      \end{multiline}

```

字字字字字字

1 – *line*

2 – *line*

3 – *line*

4 – *line* (4.4)

最后是矩阵和行列式的输入：

```

1      \begin{center}
2          $%注意要夹好
3          \begin{pmatrix}
4              a_{11} & a_{12} & \cdots & a_{1n} \\
5              a_{21} & a_{22} & & \vdots \\
6              \vdots & & \ddots & \vdots \\
7              a_{n1} & \cdots & \cdots & a_{nn}
8          \end{pmatrix}
9          $
10         $
11         \begin{bmatrix}
12             a_{11} & a_{12} \\
13             a_{21} & a_{22}
14         \end{bmatrix}
15         $
16         $\begin{vmatrix}
17             a_{11} & a_{12} \\
18             a_{21} & a_{22}
19         \end{vmatrix}
20         $
21         $
22         \begin{matrix}
23             a_{11} & a_{12}

```

```

24         a_{21} & a_{22} \\
25     \end{matrix}
26 $
27 $
28 \begin{Bmatrix}
29     a_{11} & a_{12} \\
30     a_{21} & a_{22}
31 \end{Bmatrix}
32 $
33 $
34 \begin{Vmatrix}
35     a_{11} & a_{12} \\
36     a_{21} & a_{22}
37 \end{Vmatrix}
38 $
39 \end{center}

```

效果：

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix} \left[\begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right] \left| \begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right| \begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \left\{ \begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right\} \left\| \begin{array}{cc} a_{11} & a_{12} \\ a_{21} & a_{22} \end{array} \right\|$$

5 表格

在写文章的时候表格和公式一样重要。当然，表格最重要的用途还是记录实验数据，这个时候我们就可以打个三线表（需要 booktabs 宏包）先：

```

1 \begin{table*}
2     \centering
3     \begin{tabular}{llllll}%控制表格格式
4         \toprule%一线

```

```

5          睡觉 & 1$( km/s^2 )$ & & & & Sb \\
6          \midrule%二线
7          $\mathbb{C}$ & Test \\
8          ? & & \\ %表格一定要在底线前换行
9          \bottomrule%三线
10         \end{tabular}
11         \caption{睡觉工程学导论}
12         \label{tbl:table1}
13     \end{table*}

```

效果:

睡觉	$1(km/s^2)$	Sb
\mathbb{C}	Test	
?		

表 1: 睡觉工程学导论

如果是需要竖线的表, 要合并单元格 (需要 multirow 宏包), 并使用创建框线的指令。

```

1     \multirow{NUMBER_OF_ROWS}{WIDTH}{CONTENT}
2     %NUMBER_OF_ROWS代表该表格单元占据的行数, WIDTH代表表格的宽
      度, 一般填 * 代表自动宽度, CONTENT则是表格单元里的内
      容。
3     \multicolumn{NUMBER_OF_COLUMNS}{ALIGNMENT}{CONTENT}
4     %NUMBER_OF_COLUMNS代表该表格单元占据的列数, ALIGNMENT代表
      表格内容的偏移 (填l,c或者r), CONTENT则是表格单元里的内
      容。
5
6
7     \begin{table}[htb]%浮动格式
8         \centering
9         \begin{tabular}{|c|c|c|c|c|c|c|c|}

```

```

10      %在这些竖线中间，c表示中间对称，r、l以此类推
11      \hline
12      title & title2 & title3 & title4 & \multicolumn{4}{c}
13          |}{title5} \\%向右合并，共4列，内容居中
14      \hline
15      \multirow{5}{*}{1} & \multirow{5}{column2} & \
16          multirow{5}{*}{clo3} & \multirow{5}{*}{clo4}
17      & f1 & f2 & f3 & f4 \\%向下合并，共5行，自动宽度
18      \cline{5-8}
19      & & & 1 & 2 & 3 & 4 \\
20      \cline{5-8}
21      & & & 5 & 6 & 7 & 8 \\
22      \cline{5-8}
23      & & & 1 & 2 & 3 & 4 \\
24      \cline{5-8}
25      & & & 5 & 6 & 7 & 8 \\
26      \hline
27      \end{tabular}
28      \caption{The title of the table}
29      \label{}
30      \end{table}

```

title	title2	title3	title4	title5			
1	column2	clo3	clo4	f1	f2	f3	f4
				1	2	3	4
				5	6	7	8
				1	2	3	4
				5	6	7	8

表 2: The title of the table

6 树状图

要画分叉树的话主要用 Qtree 或者 TikZ-qtree 宏包，后者的功能更强大，但是需要会 TiKZ（TikZ 是 T_EX 中画图表的非常强大的宏包，Code 的插件边栏中也有对应脚本），等后面用到的时候再做一篇吧。先介绍 qtree。

这个用法非常简单，输入 `\usepackage{ qtree}` 调用该拓展包。通过 `\Tree[]` 开始画树——“.”之后写标签，“`\qroof`”画三角形，比如：

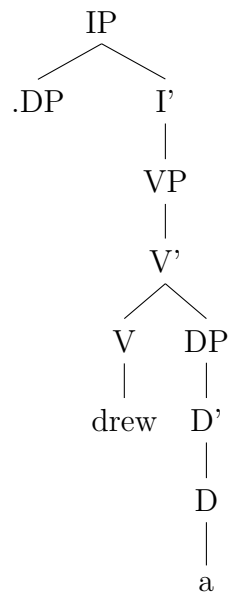
```

1      \centering
2      \Tree [ . IP
3          . DP
4          [ . I '
5              [ . VP
6                  [ . V '
7                      [ . V drew ]
8                      [ . DP
9                          [ . D ' [ . D a ] . NP ]
10                     ]
11                 ]
12             ]
13         ]
14     ]

```

注意：每个 “[” 前必须得有空格，不然系统就会报错。比如 “[.V drew]” 不能写成 “[.V drew]”。如果节点名称比较长，需要多行时，可以使用大括号包含来表示，否则会被当作子节点处理。

效果如下：



qtree-tikz 的用法大同小异，我们用它来尝试绘制一个二叉树。

```

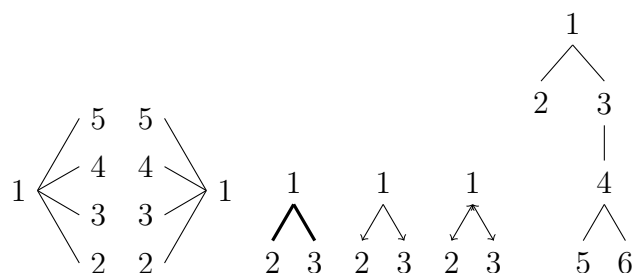
1      \begin{tikzpicture}[grow=right]%指定开口方向
2          \centering
3          \Tree
4              [.1
5                  [.2 ] [.3 ] [.4 ] [.5 ]
6              ]
7      \end{tikzpicture}
8      %%加百分号把中间的换行符注释掉
9      \begin{tikzpicture}[grow'=left]%grow'表示转置
10         \Tree
11             [.1
12                 [.2 ] [.3 ] [.4 ] [.5 ]
13             ]
14     \end{tikzpicture}
15     %
16     \begin{tikzpicture}
17         \tikzset{edge from parent/.append style={very thick}}%
            设置线的样式
18         \Tree
  
```

```

19      [.1
20      [.2 ] [.3 ]
21      ]
22 \end{tikzpicture}
23 %
24 \begin{tikzpicture}[semithick,->]%加箭头
25     \Tree
26     [.1
27     [.2 ] [.3 ]
28     ]
29 \end{tikzpicture}
30 \begin{tikzpicture}[semithick,<->]%加箭头
31     \Tree
32     [.1
33     [.2 ] [.3 ]
34     ]
35 \end{tikzpicture}
36 %
37 \begin{tikzpicture}[dashed]%虚线
38     \Tree
39     [.1
40     [.2 ] [.3 [.4 5 6 ] ]
41     ]
42 \end{tikzpicture}

```

效果:



7 页面设置

可以如此设置页边距和行距：

```
1 \linespread{0.5}      %设置{n}倍行距
2 \geometry{left=1.0cm, right=1.0cm, top=0.5cm, bottom=0.5cm
   }%设定页边距
```

以及前面提到的多栏环境：

```
1 \begin{multicols}{2}%2是栏数
2 \columnseprule 1pt %中央分割线宽
3 \columnsep 35pt %控制两栏之间间隔
4 无序列表
5 \begin{itemize}
6 \item 我是SB
7 \item SB is me.
8 \item Sb is xjtu.
9 \end{itemize}
10 有序列表
11 \begin{enumerate}
12 \item 主E20楼
13 \item 一跃解千愁{\label{sb}}
14 \end{enumerate}
15 \end{multicols}
```

现在会套模板已经够用了，最多就是加个多栏环境。

至于像学术期刊那样那么漂亮的排版，后面还会补的（吧？）。

8 引用

最后是 L^AT_EX 最为强大的功能之一：交叉引用。使用到了 hyperref 宏包。

在每个环境的 end 之前，通过`\label{ name}` 指令将环境的内部名称指定为 name，然后我们可以直接通过`\ref` 指令进行文内引用。比如上面提到的伏笔：??。

这个 1 是自动生成的，感觉有点大，我们自定义一个`\upref` 命令让数字显示在右上角??。

```
1 \newcommand{\upref}[1]{\textsuperscript{\ref{#1}}}
```

使用 hyperref 宏包还可以链接 外部网站和 本地文件，格式如下：

```
1 使用hyperref宏包还可以链接
2 \href{https://space.bilibili.com/16725323}{外部网站}和
3 \href{run:/sample.bib}{本地文件}
```

关于参考资料的外部引用，则是使用的`\cite` 指令，同样自定义一个`\upcite` 指令。当然我们要先创建参考资料环境：

```
1 \begin{thebibliography}{99}
2 \bibitem{a}作者. \emph{文献}[M]. 地点:出版社,年份.
3 \bibitem{b}作者. \emph{文献}[M]. 地点:出版社,年份.
4 \end{thebibliography}
```

```
1 \newcommand{\upcite}[1]{\textsuperscript{\cite{#1}}}
```

然后引用^[?]。注意这种方法需要两次以上编译。

另外一种方法是以 bibtex 来管理文献，这种方法很适合批量管理文献。为此，要创建一个sample.bib 文件，里面像 json 一样写入了文献的信息。例如：

```
1 @article{12,%12是这篇文献的label
2 title = {基于无迹卡尔曼滤波的室内定位系统},
3 author = {王袁雪;张前波;周媛媛;刘英明;李冰},
4 authoraddress = {河北师范大学中燃工学院},
5 journal = {物联网技术},
6 year = {2022},
```

```

7      volume          = {12},
8      number          = {07},
9      pages           = {18-19},
10     keywords         = {室内定位;超宽带;无迹卡尔曼滤波;非视距;
        chan算法;二维地图},
11     isbn/issn         = {2095-1302},
12     notes            = {61-1483/tp},
13     doi              = {10.16667/j.issn
        .2095-1302.2022.07.005},
14     databaseprovider = {cnki}
15 }

```

然后在文章末尾指定 bib 数据库文件和样式:

```

1      \bibliographystyle{IEEEtran}
2      \bibliography{sample.bib}

```

开始愉快地引用吧^[?]，至于排号和文献的先后顺序^[?]，bib 会帮你排好的^[?]。不过要同样注意: bibtex 需要 $\text{xe} \rightarrow \text{bib} \rightarrow \text{xe} \rightarrow \text{xe}$ 这样编译数次。以及这篇文档因为用了两种方式，所以出现了两个“参考文献”。

9 常用模板

最后终于来到了模板介绍时间。现在做了四个互相补充的模板:

- 平时作业
 - 用于平时选修课的作业
 - 使用 bib 管理参考文献，排版已经排好
- 数模论文
 - 大佬分享的建模论文模板
 - 在文件夹中附上了 mcode 宏包，支持 MATLAB 代码
 - 自定义了论文写作时要用的多种环境

- 数学习题
 - 提供了带有底色的问题-解答-评注环境
 - 可以按照自己意愿修改配色方案
- 代码
 - 调好了代码样式和配色
 - 主要支持 C/C++ 语言种类
 - 很适合学习 CS 时拿来记笔记

终于结束了，如果你是除了 Shiyuu 以外的阅读者，必须感谢你读到这里。这只是我个人到目前为之有需求的地方，不免有遗漏和偏颇，排版和屁话让这篇文档又臭又长且可能有点自说自话，但还是希望这篇自用的文档能有所帮助。

参考文献

- [1] <https://zhuanlan.zhihu.com/p/379321421>.