

这是一篇用 LaTeX 写的 LaTeX 教程

IT 工匠

May 2019

摘要

很多人习惯使用 word 写文档，这对普通人来说足够用了，但是对于需要撰写专业文档的工程师来说不仅操作太过复杂，而且很多需求无法满足（比如对公式和代码的支持），一部分对公式要求不是很严格的工程师会选择使用 markdown 写文档，特点是轻量级、对代码支持好、上手难度小、满足不那么正规的文档交流，而如果需要书写更专业的文档，比如简历、论文等，就需要更正规的工具了——LaTeX，LaTeX 的特点是完美支持各种公式和符号、排版简约美观，但是上手难度相比 markdown 略大，本文是笔者在使用 LaTeX 的一些总结，可以满足初学者的日常需求。接下来是广告时间：本文使用纯 LaTeX 编写，关注微信公众号“IT 工匠”，后台回复“O3”（注意是大写字母欧不是零）可获取本文 LaTeX 源代码（强烈建议大家研读一下本文源码，因为很多东西由于时间原因都没介绍到，比如如何插入代码、如何生成目录等，但是在本文均有使用，相信大家看一遍源码就都懂了）以及本文 pdf，另外还附赠一份我个人和女朋友（代号小草莓，公众号“烂草莓”）一直在用 **LaTeX 中文简历模板**，转载请注明出处。

目录

1	OverLeaf 入门	3
1.1	OverLeaf 概述	3
1.2	OverLeaf 使用	3
2	LateX 基本使用	7
2.1	文本 & 排版	7
2.1.1	定版型	7
2.1.2	加标题/日期/作者	7
2.1.3	修改页边距	7
2.1.4	文本加粗	7
2.1.5	左对齐	7
2.1.6	换行	8
2.1.7	空格	8
2.1.8	居中/左对齐/右对齐	8
2.2	公式编辑	8
2.2.1	插入公式	8
2.2.2	粗体	8
2.2.3	上下标	8
2.2.4	分数	8
2.2.5	累和、累积	9
2.2.6	常用符号表	9
2.3	插入图片	11
2.3.1	将本地图片上传到 OverLeaf 的项目中	11
2.3.2	编写代码将图片插入到正文中	12
2.4	插入表格	13
2.4.1	粗线（表格的第一根线和最后一根线比表格中的横线更粗一些）	13
2.4.2	调整位置	13
2.4.3	居中	14
2.4.4	制表	14
2.4.5	列的合并（合并一行多列）	14
2.4.6	行的合并（合并一列多行）	16

2.4.7	合并行的同时合并列	18
2.4.8	标签和名称	19
2.4.9	普通线	19
2.5	参考文献的书写	19

1 OverLeaf 入门

1.1 OverLeaf 概述

Overleaf 是一个使用 LaTeX 进行多人协同编辑的平台，可以免费注册和使用，不用下载 LaTeX 软件，是最为著名的 LaTeX 在线协作系统。主要特色是有 LaTeX 插件，编辑功能十分完善，有实时预览（即编即看，无需手动编译）的功能。科研工作者可以在各大期刊的网站上下载到其 Overleaf 模板，进行论文写作（不同模板的排版方式（比如“作者”和“地址”）的格式不同）。平台是纯英文的，需要使用者适应，这也是优秀的工程师应该具备的基本素养。

1.2 OverLeaf 使用

OverLeaf 的网址是：<https://www.overleaf.com>，进入网站后，首页如图 1所示，大家可以根据自己的情况使用第三方账号进行登录。

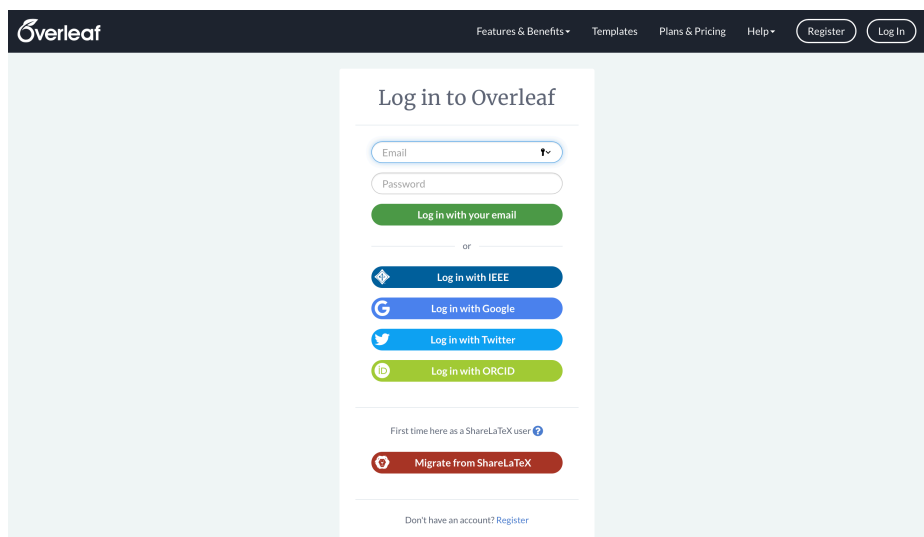


图 1: OverLeaf 首页

登录成功后就可以创建项目，如图 2所示，直接选择 Blank Project 即可。

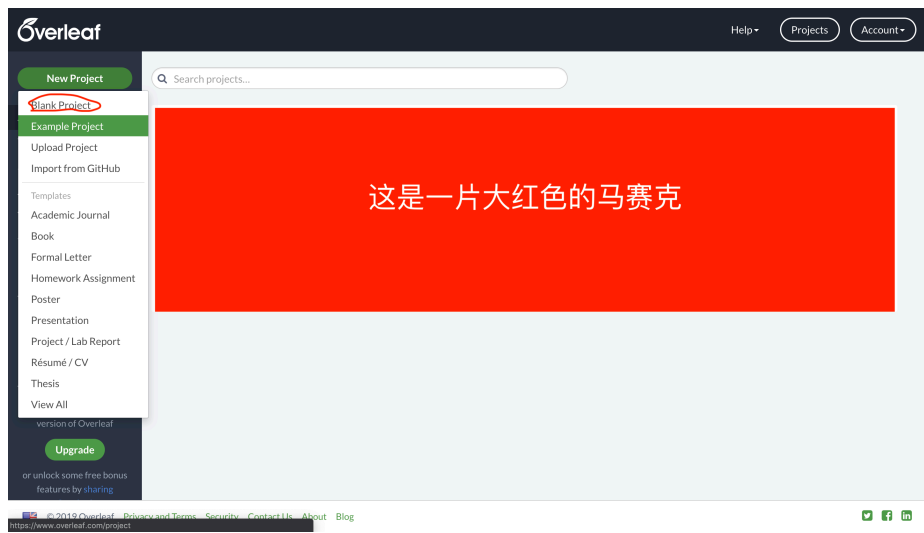


图 2: 创建项目

项目新建完成后初始页面如 3所示:

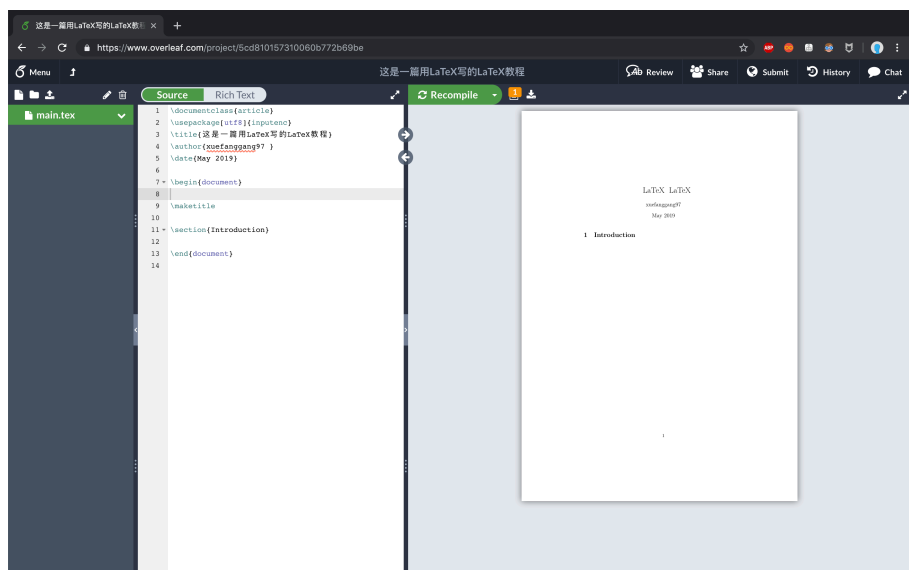


图 3: 新项目初始页面

编辑界面分为三栏，从左往右看，第一栏为工具栏，主要用作对项目的配置和管理，第二栏是编辑界面，主要用来编写代码，第三栏是预览栏，主要是预览作用。可能细心的同学已经发现，我这里新建完项目之后右边预览栏中的标题是没有正确显示的（正常应该显示“这是一片用 LaTeX 编写的 LaTeX 教程”，此处显示为“latex...”），这是因为 overleaf 默认是不支持中文的，所以我们需要配置一下，让当前项目支持中文，当然如果你是准备写英文就可以不用配置。具体的配置分为 2 步：

1: 添加 ctex 宏包

将代码 `\usepackage{ctex}` 添加到你代码的顶部即可。

2: 配置使用 xelatex 引擎编译

如图 4所示，在左边第一栏中点击“Menu”，然后设置 Compiler 为 xelatex。

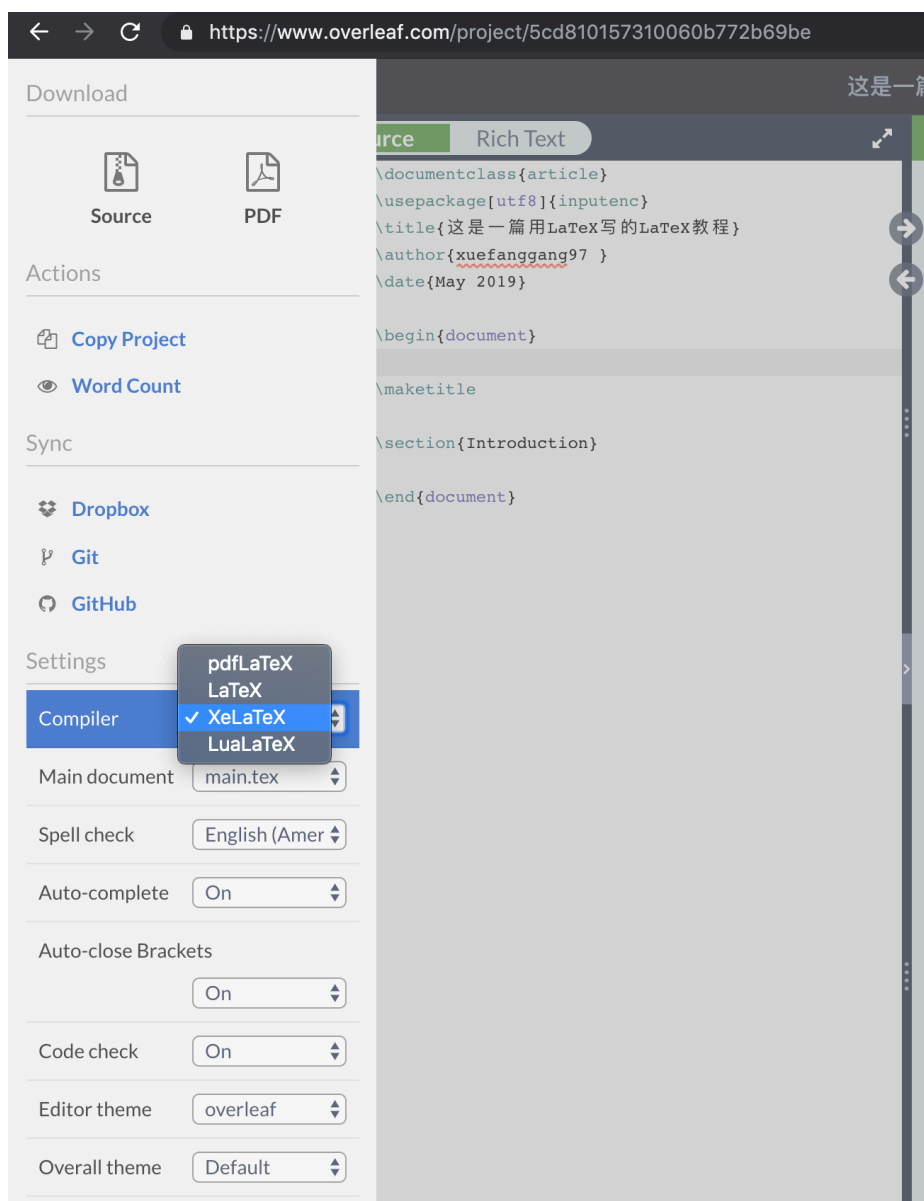


图 4: 设置编译器为 xelatex

完成了以上两步之后，你的当前项目就可以完美支持中文了，你就可以开始你愉快的写作了。

接下来我来介绍一下 LaTeX 常用的一些操作。

2 LateX 基本使用

2.1 文本 & 排版

2.1.1 定版型

`\documentclass [参数 A] 参数 B`

- 参数 A
- 1 字体 10pt(默认值), 11pt, 12pt, 例子: `\documentclass[11pt]article;`
 - 2 纸张大小有几个, 最常见的就是 a4paper, letterpaper(默认值), 例子: `\documentclass[a4paper]{article};`
 - 3 单双面 oneside(article, report 默认值), twoside(book 默认值), 例子: `\documentclass[twoside]{article};`
 - 4 组合实现: `\documentclass[a4paper,twoside,11pt]{article}` 顺序随意;

- 参数 B
- 1 常用: Article (英文科研文章) /report/book;
 - 2 ctex 文档类 (支持中文): ctexart/ctexrep/ctexbook;

2.1.2 加标题/日期/作者

在 `\begin{document}` 之前输入: `\title 标题\author 作者\date 日期`, 输入空格即为空

在 `\begin{document}` 之前输入: `\maketitle`, 输入后, 前三者才生效

2.1.3 修改页边距

首先需要导入宏包: `\usepackage{geometry}`

然后使用: `\geometry(left=2.54cm, right=2.54cm, top=3.09cm, bottom=3.09cm)`, 表示 A4 版上下为 2.54 厘米, 左右为 3.09 厘米

2.1.4 文本加粗

`\textbf{ 需要加粗的内容 }`

2.1.5 左对齐

`\noindent`, 表示本行左对齐不缩进

2.1.6 换行

`\newline` 或者 `\\`

2.1.7 空格

单空格 `\quad` , 双空格 `\quad\quad`

2.1.8 居中/左对齐/右对齐

- 部分居中
`\centering`, 小范围内 (比如表格) 居中后面部分内容
- 全部居中/左对齐/右对齐
`\begincenter/flushleft/flushright`
要居中的内容
`\endcenter/flushleft/flushright` ;

2.2 公式编辑

2.2.1 插入公式

- 行内插入公式: `$ 公式 $`
- 行间插入公式 (自动为公式编号): `$$ 公式 $$`

2.2.2 粗体

粗体主要是写向量或矩阵的时候会需要加粗, 用法: `\mathbf{}`

2.2.3 上下标

- 上标: `^`, 比如: `a^2` 表示 a^2
- 下标: `_`, 比如: `a_2` 表示 a_2

2.2.4 分数

`\frac{分子}{分母}`

2.2.5 累和、累积

- 累和： $\sum_{\text{下标}}^{\text{上标}}$ ，比如 $\sum_{n=1}^{100}$ 的结果为 $\sum_{n=1}^{100} 100$
- 累积： $\prod_{\text{下标}}^{\text{上标}}$ ，比如 $\prod_{n=1}^{100}$ 的结果为 $\prod_{n=1}^{100} 100$

2.2.6 常用符号表

常用的符号表如下面几张图所示：

命令	显示	命令	显示
<code>\alpha</code>	α	<code>A</code>	A
<code>\beta</code>	β	<code>B</code>	B
<code>\gamma</code>	γ	<code>\Gamma</code> <code>\varGamma</code>	Γ \varGamma
<code>\delta</code>	δ	<code>\Delta</code> <code>\varDelta</code>	Δ \varDelta
<code>\epsilon</code>	ϵ	<code>E</code>	E
<code>\eta</code>	η	<code>H</code>	H
<code>\theta</code>	θ	<code>\Theta</code> <code>\varTheta</code>	Θ \varTheta
<code>\kappa</code>	κ	<code>K</code>	K
<code>\lambda</code>	λ	<code>\Lambda</code> <code>\varLambda</code>	Λ \varLambda
<code>\mu</code>	μ	<code>M</code>	M
<code>\nu</code>	ν	<code>N</code>	N
<code>\pi</code>	π	<code>\Pi</code> <code>\varPi</code>	Π \varPi
<code>\rho</code>	ρ	<code>P</code>	P
<code>\sigma</code>	σ	<code>\Sigma</code> <code>\varSigma</code>	Σ \varSigma
<code>\tau</code>	τ	<code>T</code>	T
<code>\phi</code> <code>\varphi</code>	ϕ φ	<code>\Phi</code> <code>\varPhi</code>	Φ \varPhi
<code>\omega</code>	ω	<code>\Omega</code> <code>\varOmega</code>	Ω \varOmega

图 5: 希腊字母

命令	显示	命令	显示
<code>\pm</code>	\pm	<code>\mp</code>	\mp
<code>\times</code>	\times	<code>\div</code>	\div
<code>\circ</code>	\circ	<code>\bullet</code>	\bullet
<code>\cdot</code>	\cdot	<code>\cup</code>	\cup
<code>\cap</code>	\cap	<code>\subset</code>	\subset
<code>\supset</code>	\supset	<code>\subseteq</code>	\subseteq
<code>\supseteq</code>	\supseteq	<code>\leq</code>	\leq
<code>\geq</code>	\geq	<code>\propto</code>	\propto
<code>\in</code>	\in		

图 6: 基本运算符

命令	显示	命令	显示
<code>\sum</code>	Σ	<code>\int</code>	\int
<code>\sum_{i=1}^N</code>	$\sum_{i=1}^N$	<code>\int_a^b</code>	\int_a^b
<code>\prod</code>	Π	<code>\iint</code>	\iint
<code>\prod_{i=1}^N</code>	$\prod_{i=1}^N$	<code>\iint_a^b</code>	\iint_a^b
<code>\bigcup_{i=1}^N</code>	$\bigcup_{i=1}^N$	<code>\bigcap_{i=1}^N</code>	$\bigcap_{i=1}^N$

图 7: 积分运算符

命令	显示	命令	显示
<code>\cdotp</code>	\cdot	<code>\cdots</code>	\dots
<code>\ddots</code>	\ddots	<code>\infty</code>	∞
<code>\partial</code>	∂	<code>\bot</code>	\perp
<code>\hat{a}</code>	\hat{a}	<code>\tilde{a}</code>	\tilde{a}
<code>\bar{a}</code>	\bar{a}	<code>\vec{a}</code>	\vec{a}
<code>\dot{a}</code>	\dot{a}	<code>\sqrt{a}</code>	\sqrt{a}
<code>\sqrt[3]{2}</code>	$\sqrt[3]{2}$	<code>a^{3}</code>	a^3
<code>\frac{1}{a}</code>	$\frac{1}{a}$	<code>\lim_{x \rightarrow 0}</code>	$\lim_{x \rightarrow 0}$

图 8: 其他符号

2.3 插入图片

插入图片需要 2 步:

2.3.1 将本地图片上传到 OverLeaf 的项目中

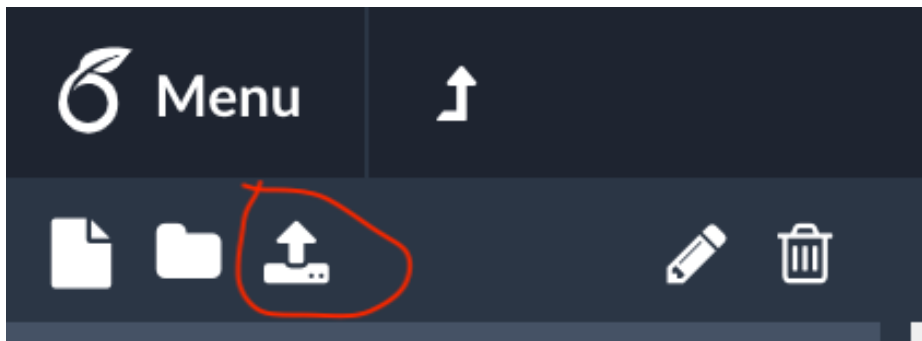


图 9: 上传图片示意图-1

如图 9 所示, 点击 menu 栏的上传图标, 然后就可以看到如图 10 所示的选择界面, 然后点击那个绿色按钮就可以选择你本地的图片进行上传, 当然你也可以直接将本地的图片拖拽到相应的区域上传。

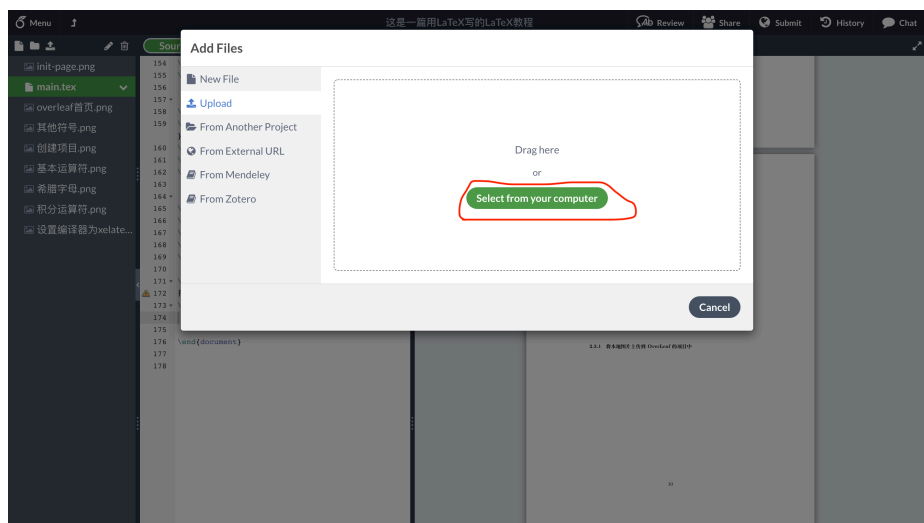


图 10: 上传图片示意图-2

2.3.2 编写代码将图片插入到正文中

将图片成功上传到 OverLeaf 的项目中后就可以编写代码在正文中插入相应的图片了，代码的编写规则如下：

```
\usepackage{graphicx} %加入头文件\\
\begin{figure}[! htpb]/[H] %[htbp] 是自动排版；[H]
    固定位置
\centering %图片居中
\includegraphics[ scale=0.3]/[ width=4.5 in]{DPBS
    .png} %设置大小和名称
\caption{DPBS}
\label{1} %图片名称和图片标号
\end{figure} %结束
```

其中，figure 的可选参数 [!htbp]：；

- 『h』 当前位置。将图形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，这一参数将不起作用。
- 『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。
- 『p』浮动页。将图形放置在一只允许有浮动对象的页面上。

如果想让图片出现在你编写代码的对应位置（默认情况下图片是自定根据美观调整位置的）一般使用 [htb] 这样的组合，只用 [h] 是没有用的。这样组合的意思就是 latex 会尽量满足排在前面的浮动格式，就是 h-t-b 这个顺序，让排版的效果尽量好。

!h 只是试图放在当前位置。如果页面剩下的部分放不下，还是会跑到下一页的。一般言，用 [!h] 选项经常会出现不能正确放置的问题，所以常用 [ht]、[htbp] 等。

如果你一定需要把图片放在当前位置，不容改变，可以用 float 宏包的 [H] 选项。不过如果这样做，出现放不下的问题时需要手工调整。使用格式如下：`\usepackagefloat`

2.4 插入表格

2.4.1 粗线（表格的第一根线和最后一根线比表格中的横线更粗一些）

导入宏包：`\usepackage{booktabs}`
`\toprule`，第一根线
`\midrule`，中间的线
`\bottomrule`，最后一根线

2.4.2 调整位置

`\begin{table}[!htbp]`
`\end{table}` 其中，`{table}` 有若干可选参数 [!htbp]

- h 代表 here, 将表格排在当前文字位置;
- t 表示将表格放在下一页的 top (页首);
- p 表示 p-page-of-its-own;
- b 表示将表格放在当前页的 bottom (底部);
- ! 表示忽略美观因素，尽可能按照参数指定的方式来处理表格浮动位置;

2.4.3 居中

长度不长时:\centering

长度过长时:\centerline{}，把 tabular 的所有内容放进去

2.4.4 制表

`\begin{tabular}{|l|c|r|}`，“|”表示竖线，“l/c/r”表示格内文字居左/中/右显示

```
A & B & C \
```

```
E & F & G \
```

```
\end{tabular}
```

“&”分隔不同列内的内容，“\\”表示换行

2.4.5 列的合并（合并一行多列）

我们使用 `\multicolumn{项数}{新列格式}{内容}` 命令将一行中的几个不同的表格合并为一项。

比如原本有一个表格是这样的：

```
\begin{tabular}{|c|c|c|}
```

```
\hline
```

```
成绩1 & 成绩2 & 总计 \\
```

```
\hline
```

```
语文 & 数学 & \\
```

```
\hline
```

```
87 & 100 & 187 \\
```

```
\hline
```

```
\end{tabular}
```

绘制出来效果如表 1所示.

成绩 1	成绩 2	总计
语文	数学	
87	100	187

表 1: 合并列之前的表格

现在我们需要将 1 的第 1 行的两列（成绩 1、成绩 2）合并成为 1 列（成绩），我们应该使用 `\multicolumn{合并的列数}{新列格式}{合并后行的内容}` 命令：

```

\begin{tabular}{|c|c|c|}
\hline

\multicolumn{2}{|c|}
{成绩}&总计 \\
%合并本行的2项，新列的格式是,2项合并后的本
列的格式是|c|,合并后本格的内容是{成绩}&
总计

\hline

语文 & 数学 & \quad \\
\hline

87 & 100 & 187 \\
\hline

\end{tabular}

```

绘制出来的表格如表 2 所示

成绩		总计
语文	数学	
87	100	187

表 2: 合并列以后的表格

在这里需要注意的一个地方就是当我们使用了 `\multicolumn{合并的列数}` 的时候这个时候我们在表头定义的格式就会完全对我们重新合并的单元格失去作用, 所以我们自己必须重新定义新的格式。也就是我们例子中的这个样子 `\multicolumn{2}{|c|}`

2.4.6 行的合并 (合并一行多列)

合并多行 1 列单元格可以用 `multirow` 包中的 `\multirow{合并的行数}{宽度}{合并后列中的内容}` 来实现, 注意和 `\multicolumn` 命令的 2 点不同:

- `\multicolumn` 不需要导入任何宏包可以直接使用, `\multirow` 必须使用 `\usepackage{multirow}` 导入 `multirow` 这个宏包
- `\multirow{合并的行数}{宽度}{合并后列中的内容}` 中的第 2 个参数是 {宽度}, 与 `\multicolumn` 第 2 个参数不同。如果不确定 {宽度} 需要填什么, 就将其替换为 `*`, 如下面代码块中所示

例如如下表格:

```
\begin{tabular}{|c|c|c|}
\hline

成绩1 & 语文 & 87\\
\hline

成绩2 & 数学 & 100 \\
\hline

总计 & \quad & 187 \\
\hline

\end{tabular}
```

其绘制出来如表 3:

成绩 1	语文	87
成绩 2	数学	100
总计		187

表 3: 合并行之前的表格

现在我们想合并第 1 列的前 2 行，应该使用如下代码：

```
\begin{tabular}{|c|c|c|}
\hline
\multirow{2}{*}{成绩}

~ & 语文 & 87\\
\cline{2-3}

~ & 数学 & 100 \\
\hline

总计 & \quad & 187 \\
\hline

\end{tabular}
```

绘制完成之后的效果如表 4所示：

成绩	语文	87
	数学	100
总计		187

表 4: 合并行之后的表格

这里需要注意 2 点：

- 第 3 行的 `\multirow` 的第 2 个参数使用了，注意不是 `{}` 哦
- 第 5 行的 “`& 语文 & 87\`” 和第 8 行的 “`& 数学 & 100 \`” “中都有个其 实这个不 是必须要的，但是 后面紧跟的那个 `&` 是必须的，它

起到一个占位作用，不然编译器怎么知道你要合并的是哪个位置的 2 行（是 1-2 行还是 2-3 行），我这里为了更加显眼写了个`~`，大家在实际使用的时候可以省略，但是后面的那个 `&` 是绝对不能省的

- 第 6 行使用了 `\cline{2-3}` 来画这一行的直线，没有使用 `\hline`，因为 `\hline` 绘制的是从表格开始到表格结束的一行直线，很明显我们这里只需要在第 2 列到第 3 列的当前位置绘制一条横线即可，所以使用 `\cline{2-3}`

2.4.7 合并行的同时合并列

其实只要学会了合并行和合并列，这里只要将二者同时使用即可，这里给出一个示例：

```

\begin{tabular}{|c|c|c|}
\hline
\multicolumn{2}{|c|}
{姓名}&小草莓\\

\hline

\multirow{2}{*}{成绩}
& 语文  & 87\\
\cline{2-3}

& 数学  & 100 \\

\hline

总计      & \quad & 187  \\
\hline

```

`\end{tabular}`

绘制完成后如表 5 所示:

姓名		小草莓
成绩	语文	87
	数学	100
总计		187

表 5: 合并行和列

2.4.8 标签和名称

`\label{label}\caption{name}`

2.4.9 普通线

`\hline` 或 `\cline{2-5}`, 两者都是用来画水平的表格线, 但是 `\cline` 可以用来指定画线的起始和终止位置

`\subsubsection{行高}` `\renewcommand\arraystretch{2}`, 表格行高设置为默认的 2 倍

2.5 参考文献的书写

```
\begin{thebibliography}{}
  \bibitem{ref label}
  内容
  {\em 要斜体的内容} {\bf 要加粗的内容}
\end{thebibliography}
```

参考文献

[1] 微信公众号”IT 工匠”