

系统开发工具基础第一周报告

韩昊鑫 24020007039

August 31, 2025

Contents

0.1	LaTeX 初学内容及简单演示	1
0.1.1	发出指令	1
0.1.2	文章类型与格式	1
0.1.3	注释与换行	1
0.1.4	正文	1
0.1.5	题目作者与日期	1
0.1.6	章节和子章节	1
0.1.7	文章目录	1
0.1.8	字体颜色和大小	2
0.1.9	有序/无序列表	2
0.1.10	转义字符	2
0.1.11	表格	3
0.1.12	公式	3
0.1.13	图片	4
0.2	Git 初学内容及简单演示	4
0.2.1	绑定 Github 账号	4
0.2.2	与 github 连接	5
0.2.3	创建新项目	5
0.2.4	从本地到 github	5
0.2.5	从 github 到本地	5
0.2.6	使用 vscode	5
0.2.7	搜索	5
0.3	简短感悟	6
0.4	commit 记录	6

0.1 LaTeX 初学内容及简单演示

0.1.1 发出指令

所有的命令都要以\开头

0.1.2 文章类型与格式

使用“`\documentclass[a4paper, 12pt]{report}`”来指定文档类型,其中 a4paper 是页面大小, 12pt 是默认字体大小, report 是文章类型

0.1.3 注释与换行

使用%来表示从% 之后该行所有内容视为注释, 相当于//(同 C++ 作比, 下同)

使用\\来表示换行, 相当于 endl

0.1.4 正文

正文所有部分都要在“`\begin{document}`”和“`\end{document}`”之间, begin 之前的视为文章配置 (前导命令), 相当于 #include, end 之后的会被忽视

0.1.5 题目作者与日期

在`\title{此处为文章题目}`中填写文章题目

在`\author{此处为作者}`中填写作者信息

在`\date{此处为文章日期}`中填写文章完成日期, 也可以从中填写`\today`自动定位当日日期

0.1.6 章节和子章节

使用`\section{此处为章节名字}`为你创建一个新的章节, 章节自动排序

使用`\subsection{此处为子章节名字}`为你的章节创建一个子章节, 子章节也自动排序

0.1.7 文章目录

1 使用`\pagenumbering{roman}`将页码改为罗马数字, 目录页单独一种编排方式, 便于正文页码编排

2 使用`\tableofcontents`为你的所有章节创建目录

- 3 使用`\newpage`为你新开一页，使目录单独成页
- 4 使用`\pagenumbering{arabic}`将页码改回阿拉伯数字以编排正文页码

0.1.8 字体颜色和大小

- 1 用 `{ }` 来限定特殊颜色/字体的范围
- 2 如果想要改变字体颜色, 首先要在前导部分(begin 之前)输入`\usepackage{color}`导入颜色包, 下面举例使用方法: `{\color{green}I am green}`—>`I am green`
- 3 如果想要改变字体大小, 只需在原文本前加上相关大小指令如`\huge`, 如`{\huge I am huge}`—>`I am huge`

0.1.9 有序/无序列表

列表形式如下:

其中 `enumerate` 表示有序列表, `itemize` 表示无序列表

```
\begin{enumerate}
  \item[++] first
  \item[--] second
  \begin{itemize}
    \item one
    \item two
    \item three
  \end{itemize}
  \item [...]third
\end{enumerate}
```

Figure 1: 列表代码展示

0.1.10 转义字符

一些字符比如 `# $ % ^ & _ { } ~ â ã` 不能直接输出, 需要用`\`来转义以达到输出的目的

0.1.11 表格

表格形式如下：

FIRST	F	1
SECOND	S	2
THIRD	T	3

```
\begin{tabular}{|l|l|l|}  
\hline  
FIRST & F & 1\\  
\hline  
\hline  
SECOND & S & 2\\  
\cline{2-3}  
THIRD & T & 3 \\  
\cline{1-1}  
\end{tabular}
```

Figure 2: 表格代码展示

0.1.12 公式

LaTeX 中提供了很多数学/其他科目公式，例如上标下标，求和积分，乘方开根等，下面是一些样例

$$666 = 666 \quad (1)$$

$$a = b + c^2 \quad (2)$$

$$b = m + \sqrt{n} \quad (3)$$

$$c = \sum_{i=1}^n n + \int_0^c ab + \sum_{i=0}^m \int_0^n \frac{a}{\frac{a}{b} + 2} \quad (4)$$

0.1.13 图片

图片格式如下：

```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.5\linewidth]{paper.jpg}
    \caption{MywallPaper}
    \label{wallpaper}
\end{figure}
```

Figure 3: 图片代码展示



Figure 4: MyWallPaper

0.2 Git 初学内容及简单演示

结合 VScode 使用更佳

0.2.1 绑定 Github 账号

`git config --global user.name " 这里是你的账号名"` (注意有引号)
`git config --global user.email 这里是你账号的邮箱地址`

0.2.2 与 github 连接

从 git bash 生成自己的 ssh 密钥并添加到 github 中 (ssh-keygen -t rsa), 使用 ssh -T git@github.com 检测是否连接成功

0.2.3 创建新项目

`git init`: 创建一个新的 git 仓库, 其数据会存放在一个名为 .git 的目录下

0.2.4 从本地到 github

`git add <filename>`: 增添新的文件

`git add .`: 本地有变动文件全部增添到 github

`git commit -m "X"`: 创建一个新的提交, 并标注为 X

`git push`: 把提交同步到 github

`git status`: 显示当前的仓库状态

`git log`: 显示历史日志

0.2.5 从 github 到本地

`git clone <url>`: 把对应 url 的仓库 copy 到当前文件

0.2.6 使用 vscode

如果使用 vscode, 创建文件夹之后关联到 git 即可

使用 vscode, 在源代码管理一栏可以给当前版本命名并提交更改, 这样便可以保存下来当前版本的文件

vscode 支持直接查看, 也在源代码管理一列, 红色表示更改内容

vscode 支持直接上传到 github, 位于源代码一列, 并且从 github 上也可直接查看 (位于个人主页位置), 需要绑定 github 账号

0.2.7 搜索

1. `in: name,description,readme` spring 为查找名称, 描述或 README 文件中包含 spring 的文件按名称, 描述或 README 搜索
2. `stars:>1000` 为查找星标超过 1000 的仓库按星标数量搜索
3. `forks:>500` 为查找复刻超过 500 次的仓库按复刻数量搜索
4. `language:python` 为查找主要使用 python 语言的仓库按编程语言搜索
5. `pushed:>2025-01-01` 为查找在 2025 年 1 月 1 日之后有更新的仓库按更新时间搜索

0.3 简短感悟

关于 LaTeX，这是一种很体系的语言，很精准的实现了 PDF 的制作，但是感觉有点繁琐，总体上简单易上手。

关于 git，之前有想过会有类似软件来实现 < 回溯 > 功能，现在正式接触了 git (包括使用 vscode 和命令行 bash 两种方式)，不过没有触及底层逻辑

0.4 commit 记录

下面是 commit 截图和 github 链接:

```
10376@ MINGW64 ~/Desktop/git test/git-test (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   firstWeekReportPDF.pdf

10376@ MINGW64 ~/Desktop/git test/git-test (main)
$ git commit -m "firstWeekReportPDF"
[main a7c2af8] firstWeekReportPDF
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 firstWeekReportPDF.pdf

10376@ MINGW64 ~/Desktop/git test/git-test (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 24 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 247.04 KiB | 24.70 MiB/s, done.
```

Figure 5: commit 记录

<https://github.com/C-learning-beginner/git-test.git>