

实验报告——git 和 Latex

姓名：徐亚齐 学号：23020007136

2024 年 8 月 23 日

目录

1 练习内容	2
1.1 git 实例	2
1.1.1 获取 Git 仓库	2
1.1.2 检查当前文件状态	3
1.1.3 跟踪新文件	3
1.1.4 Git 别名	4
1.1.5 忽略文件	4
1.1.6 查看修改	5
1.1.7 提交更新	6
1.1.8 移除文件	7
1.1.9 文件改名	7
1.1.10 查看提交历史	7
1.2 Latex 文档编辑实例	9
1.2.1 标题的添加	9
1.2.2 标签的创建与引用	10
1.2.3 目录	10
1.2.4 改变字体的颜色	11
1.2.5 段落缩进问题	11
1.2.6 特殊字符的表示	13
1.2.7 在 Latex 文本中输出命令语句	13

1.2.8	画表格	14
1.2.9	插入图片	15
1.2.10	数学公式的表示	15
2	结果	16
3	解题感悟	16

1 练习内容

1.1 git 实例

学习了 git 的基本内容，尝试练习使用 git

1.1.1 获取 Git 仓库

安装配置好 git 后，需要初始化仓库，输入语句 `git init` 后，初始化完成。然后利用 `git add` 来添加追踪文件，得到存在被追踪文件与初始提交的 Git 仓库。也可以通过克隆现有的仓库来得到 Git 仓库。

结果：



```
MINGW64:/D/git
Lenovo@LAPTOP-FGVNVP6K MINGW64 ~ (main)
$ cd /D/git
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git
$ git init
Initialized empty Git repository in D:/git/.git/
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git add track.txt
```

图 1: 实例 1 (1)

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git clone https://github.com/schacon/simplegit-progit
Cloning into 'simplegit-progit'...
remote: Enumerating objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13 (from 1)
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```

图 2: 实例 1 (2)

1.1.2 检查当前文件状态

检查文件状态用 `git status` 命令，它可以将已追踪文件上的更改显示出来。也可以用 `git status -s` 命令，使状态以另一种方式显示出来。

结果：

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   track.txt
```

图 3: 实例 2

1.1.3 跟踪新文件

增加新的追踪文件时，需要先添加新文件，再进行追踪。

可以用 `echo 'New file' > new` 命令来添加新文件。新添加的不会被追踪，需要用 `git add` 来对新文件进行追踪。

结果：

```

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   track.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        new

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git add new
warning: in the working copy of 'new', LF will be replaced by CRLF the next time
        Git touches it

```

图 4: 实例 3

1.1.4 Git 别名

如果想让指令更简洁, 可以通过 `git config` 来起别名, 我们可以通过起别名获得两个等价的命令, 通过这种操作命令也会更加通俗易懂。

结果:

```

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git config --global alias.last 'log -1 HEAD'

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git last
commit 7a332b39dbedbcf0aa38016788d26b03ecc62cba (HEAD -> main)
Author: xyq <3503733900@qq.com>
Date:   Wed Aug 28 21:39:59 2024 +0800

    第二次提交

```

图 5: 实例 4

1.1.5 忽略文件

对于不需要 `git` 管理的文件, 可以进行忽略, 这时需要创建一个 `.gitignore` 文件, 并将需要忽略的文件写在里面。然后再运行 `cat .gitignore` 命令就可以了。

结果:

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ cat .gitignore
# 忽略所有的 .a 文件
*.a

# 但跟踪所有的 lib.a, 即便你在前面忽略了 .a 文件
!lib.a

#忽略所有以~结尾的文件
*~
```

图 6: 实例 5

1.1.6 查看修改

按照资料的指引, 先修改 new 文件后暂存, 再修改 contributing.md 后不暂存, 此时用 `git diff` 命令, 就可以查看未暂存的文件更新的部分。用 `git diff --staged` 命令就可以查看已暂存的文件将要添加到下次提交的内容。

结果:

```

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git diff
diff --git a/contributing.md b/contributing.md
index 987ff27..5fc7880 100644
--- a/contributing.md
+++ b/contributing.md
@@ -1,2 @@
-系统开发工具基础
\ No newline at end of file
+系统开发工具基础
+GIT
\ No newline at end of file

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git diff --staged
diff --git a/contributing.md b/contributing.md
new file mode 100644
index 0000000..987ff27
--- /dev/null
+++ b/contributing.md
@@ -0,0 +1 @@
+系统开发工具基础
\ No newline at end of file
diff --git a/new b/new
new file mode 100644
index 0000000..8f710b9
--- /dev/null
+++ b/new
@@ -0,0 +1,2 @@
+New file
+Hello world
\ No newline at end of file
diff --git a/track.txt b/track.txt
new file mode 100644
index 0000000..e69de29

```

图 7: 实例 6

1.1.7 提交更新

提交前先确认修改的内容都放到了暂存区，然后再用 `git commit` 命令提交，在 `vim` 编辑器中写上提交说明，就可以了。

结果：

```

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git commit
[main (root-commit) 7748e68] 第一次提交
3 files changed, 4 insertions(+)
create mode 100644 contributing.md
create mode 100644 new
create mode 100644 track.txt

```

图 8: 实例 7

1.1.8 移除文件

移除文件用 `git rm` 命令，执行命令后用 `git status` 查看就可以发现已经移除了。如果只想把文件从暂存区删除，就要用 `git rm --cached` 命令。

结果：

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git rm track.txt
rm 'track.txt'

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    track.txt
```

图 9: 实例 8

1.1.9 文件改名

若想将追踪的文件改名，需要用 `git mv file_from file_to` 命令，操作后再次用 `git status`

结果：

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git mv contributing.md contributing

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    renamed:   contributing.md -> contributing
    deleted:   track.txt
```

图 10: 实例 9

1.1.10 查看提交历史

若想查看提交的历史记录，只需要在项目里进行 `git log` 命令即可，它可以按时间先后将提交的基本信息显示出来，若在后面加上 `-p` 则显示每次提交的差异，再加上 `-n` 则显示最近 `n` 次提交的说明。也可以在后面加上 `-stat`，来看提交的简略统计信息。结果：

```

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git log -p -2
commit 7a332b39dbedbcf0aa38016788d26b03ecc62cba (HEAD -> main)
Author: xyq <3503733900@qq.com>
Date:   Wed Aug 28 21:39:59 2024 +0800

    第二次提交

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..4ce4fd4
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,8 @@
+# 忽略所有的 .a 文件
+*.a
+
+# 但跟踪所有的 lib.a, 即便你在前面忽略了 .a 文件
+!lib.a
+
+#忽略所有以~结尾的文件
+*~
\ No newline at end of file
diff --git a/contributing.md b/contributing
similarity index 100%
rename from contributing.md
rename to contributing
diff --git a/track.txt b/track.txt
deleted file mode 100644
index e69de29..0000000

commit 7748e6849ce6fc82c995e2462b80df678ba9f9c7
Author: xyq <3503733900@qq.com>
Date:   Wed Aug 28 21:07:10 2024 +0800

    第一次提交

diff --git a/contributing.md b/contributing.md
new file mode 100644
index 0000000..5fc7880
--- /dev/null
+++ b/contributing.md
@@ -0,0 +1,2 @@
+系统开发工具基础
+...skipping...

```

图 11: 实例 10 (1)


```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git log --stat
commit 7a332b39dbedbcf0aa38016788d26b03ecc62cba (HEAD -> main)
Author: xyq <3503733900@qq.com>
Date:   Wed Aug 28 21:39:59 2024 +0800

    第二次提交

.gitignore      | 8 ++++++
contributing.md => contributing | 0
track.txt       | 0
3 files changed, 8 insertions(+)

commit 7748e6849ce6fc82c995e2462b80df678ba9f9c7
Author: xyq <3503733900@qq.com>
Date:   Wed Aug 28 21:07:10 2024 +0800

    第一次提交

contributing.md | 2 ++
new             | 2 ++
track.txt       | 0
3 files changed, 4 insertions(+)

Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
```

图 12: 实例 10 (2)

也可以在后面加 `-pretty` , 再加上 `format` 可以定制记录显示格式, 且此时的格式不会随 `git` 更新而改变。若要限制提交的时间, 用 `-since` 即可, 可以限制在一天, 也可以是一个时间段。

结果:

```
Lenovo@LAPTOP-FGVNVP6K MINGW64 /D/git (main)
$ git log --pretty="%h - %s" --since="2024-8-27"
7a332b3 - 第二次提交
7748e68 - 第一次提交
```

图 13: 实例 10 (3)

1.2 Latex 文档编辑实例

学习了 Latex 的基础命令, 并根据资料进行了练习

1.2.1 标题的添加

添加标题时, 使用 `\maketitle` 的命令, 将报告的题目用 `\title` 写出来, 姓名等信息用 `\author`, 日期可写可不写, 不表示的话会默认当前的时间。

练习代码与结果：



图 14: 实例 1

1.2.2 标签的创建与引用

使用`\label{labelname}`创建标签,如果想要引用,就用`\ref{labelname}`,这样就可以达到引用的效果。

练习代码与结果：

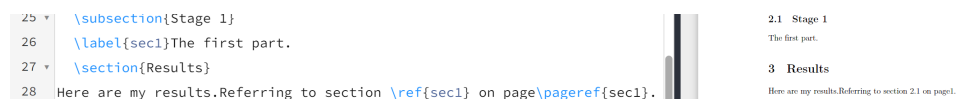


图 15: 实例 2

1.2.3 目录

创建目录需要使用`\tableofcontents`语句,在文档内输入它编译后就可以生成目录,在标注页码的同时还可以选择数字类型,比如阿拉伯数字或罗马数字,用`\pagenumbering{roman}`或者`\pagenumbering{arabic}`就可以达到改变数字类型的目的。

练习代码与结果：



图 16: 实例 3

1.2.4 改变字体的颜色

改变字体的颜色需要使用包,包引用的命令 `\usepackage[options]{color}` 应该放在文档的前导命令处,即`\begin{ararticle}`之前。在文本中使用彩色的字体,命令为`{\color{colorname}text}`,同时还可以尝试将字体的背景颜色也改变,此时的命令为`\colorbox{colorname}{text}`

练习的代码与结果:

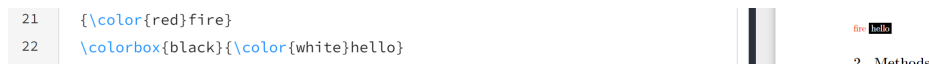


图 17: 实例 4

1.2.5 段落缩进问题

关于段落缩进问题,Latex 有默认你设置,若想段落顶格,需要在顶格的段落前加上`\noindent`命令,还有一个命令`\setlength{\parindent}{0pt}`,使用后可以使它后面的所偶有段落都顶格。

练习的代码与结果:

20	I have been in love with words and stories since I was a child. Reading books opened up a whole new world for me. I was fascinated by the power of language to create vivid images, evoke emotions, and inspire people. I began to write my own stories and poems, pouring out my thoughts and feelings onto the pages.	1 Introduction
22		I have been in love with words and stories since I was a child. Reading books opened up a whole new world for me. I was fascinated by the power of language to create vivid images, evoke emotions, and inspire people. I began to write my own stories and poems, pouring out my thoughts and feelings onto the pages.
23	As I grew older, my passion for writing only intensified. I realized that writing is not just a hobby, but a calling. I want to use my words to make a difference in the world. I want to tell stories that touch people's hearts, raise awareness about important issues, and bring hope and inspiration to those who need it.	As I grew older, my passion for writing only intensified. I realized that writing is not just a hobby, but a calling. I want to use my words to make a difference in the world. I want to tell stories that touch people's hearts, raise awareness about important issues, and bring hope and inspiration to those who need it.
24		To achieve my dream, I know that I need to work hard. I read widely to improve my vocabulary and understanding of different writing styles. I practice writing every day, honing my skills and experimenting with different genres. I also seek feedback from others and learn from my mistakes.
25	To achieve my dream, I know that I need to work hard. I read widely to improve my vocabulary and understanding of different writing styles. I practice writing every day, honing my skills and experimenting with different genres. I also seek feedback from others and learn from my mistakes.	I believe that with determination and perseverance, I can make my dream come true. One day, I hope to see my name on the cover of a best-selling book, knowing that my words have reached countless readers and made a positive impact on their lives.
26		2 Methods
27	I believe that with determination and perseverance, I can make my dream come true. One day, I hope to see my name on the cover of a best-selling book, knowing that my words have reached countless readers and made a positive impact on their lives.	2.1 Stage 1 The first part.
		2.2 stage 2
		3 Results
		Here are my results.Referring to section 2.1 on page1.

图 18: 实例 5

21	I have been in love with words and stories since I was a child. Reading books opened up a whole new world for me. I was fascinated by the power of language to create vivid images, evoke emotions, and inspire people. I began to write my own stories and poems, pouring out my thoughts and feelings onto the pages.	
22		
23	<code>\noindent</code>	
24	As I grew older, my passion for writing only intensified. I realized that writing is not just a hobby, but a calling. I want to use my words to make a difference in the world. I want to tell stories that touch people's hearts, raise awareness about important issues, and bring hope and inspiration to those who need it.	
25		
26	To achieve my dream, I know that I need to work hard. I read widely to improve my vocabulary and understanding of different writing styles. I practice writing every day, honing my skills and experimenting with different genres. I also seek feedback from others and learn from my mistakes.	
27		
28	I believe that with determination and perseverance, I can make my dream come true. One day, I hope to see my name on the cover of a best-selling book, knowing that my words have reached countless readers and made a positive impact on their lives.	

图 19: 实例 5(一段顶格)

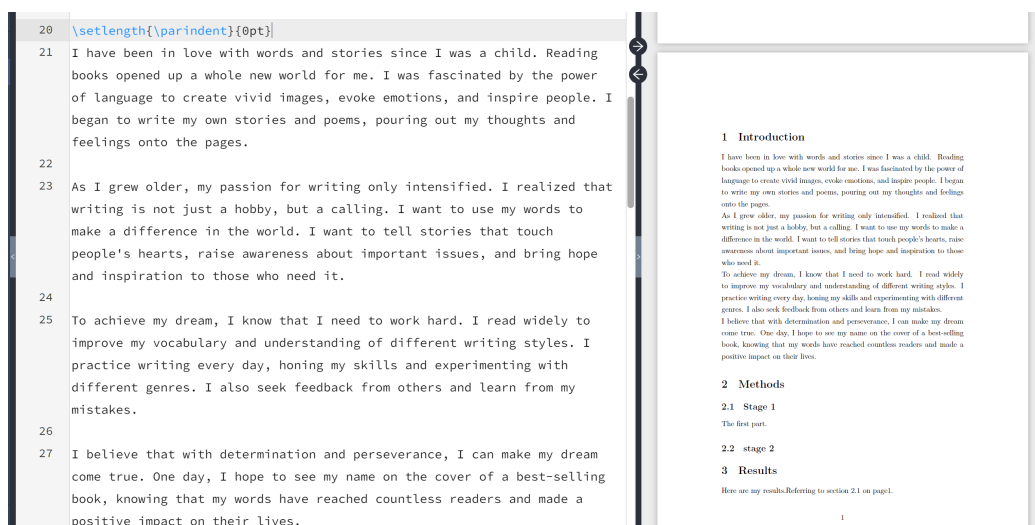


图 20: 实例 5 (全顶格)

1.2.6 特殊字符的表示

在 Latex 中, 想要输出一些有含义的特殊字符不能直接写, 应该在前面加 `\` 进行转义, 例如: `\# \$ \% \& _ \{ \}`。注意三个特殊的符号: `\^{\}` 和 `\~{\}`, 需要加 `{ }`。`\` 需要用命令 `\textbackslash` 来输出, 因为 `\\` 是换行。

练习的代码和结果:

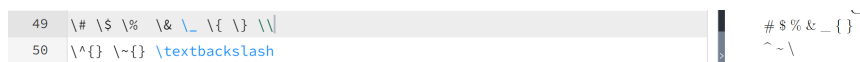


图 21: 实例 6

1.2.7 在 Latex 文本中输出命令语句

在 Latex 中的命令语句如果直接写在内容中, 会被看作一个操作命令, 无法直接在文本中表示出来, 所以需要用 `\verb` 来进行正常的文本输出, 注意使用时要用 `|` 标明开始位置和结束位置。

练习的代码和结果:



图 22: 实例 7

1.2.8 画表格

画表格开始的命令语句为`\begin{tabular}{}`, 结束的语句为`\end{tabular}`, 首先我了解了以下字母的含义:

l 表示一个左对齐的列;

r 表示一个右对齐的列;

c 表示一个向中对齐的列;

| 表示一个列的竖线;

在写表格的内容时, 要掌握: & 用于分割列; \\ 用于换行;

在标注表格的横线时, 需要了解的命令语句有两个:

\hline 表示插入一个贯穿所有列的横着的分割线;

\cline{n-m} 会在第n列和第m列插入一个横着的分割线。

练习的代码和结果:

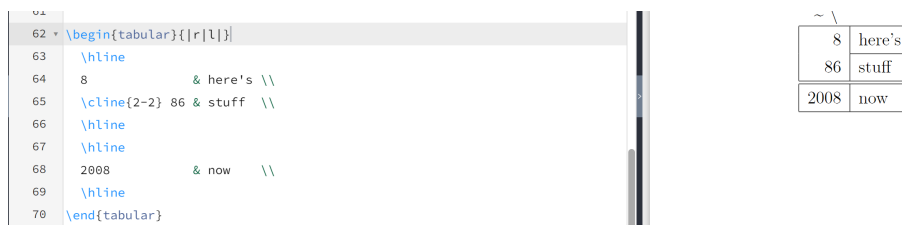


图 23: 实例 8 (1)

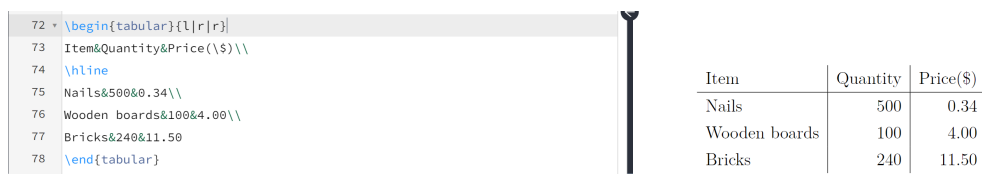


图 24: 实例 8 (2)

1.2.9 插入图片

插入图片时，可以设置图片的位置，默认左对齐。可以用`\centering`将图片放在页面中央

用`\includegraphics{}`将同一目录下的图片放在文档中。

此外还可以用`\caption{}`给图片起个标题

练习的代码与结果：



图 25: 实例 9

1.2.10 数学公式的表示

用 `$` 开始数学模式

学习了分数，上标和下标，根号，求和与积分等数学相关的内容

上标用`^`，下标用`_`，分数用`\frac{numerator}{denominator}`来表示，根号用`\sqrt{}`，求和用`\sum`，积分用`\int`。

练习的代码和结果：



图 26: 实例 10

2 结果

github 链接: <https://github.com/XY568/G-and-L.git>

部分文件上传截图:

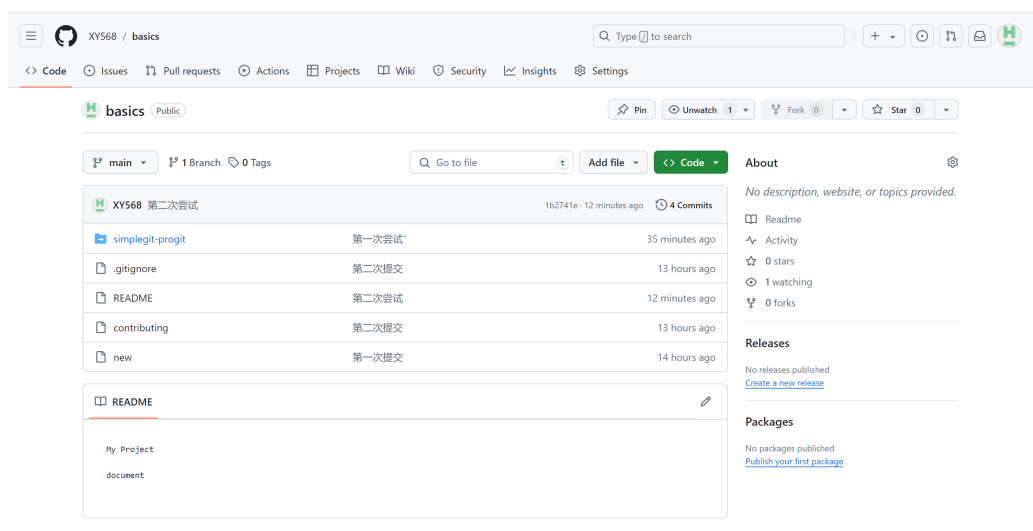


图 27: github 截图

3 解题感悟

在学习 Git 的过程中,我感受到了 Git 的重要意义,通过它我们可以保存文件的历史记录,看到每个人对文件的修改。学习 Git 主要是了解每个命令操作,我在学习的过程中主要是了解基础的命令语句,并通过练习加强记忆,遇到不懂的问题就反复尝试以寻找正确的方法,或在网上搜寻解决思路。Git 较多的命令语句对记忆是个考验,但作为一个版本控制的重要系统,它让我计划在接下来的学习中继续加强练习,我相信,在未来的学习和工作中, Git 一定会成为一个非常有帮助的存在。

学习 LaTeX 让我又掌握了一个新的排版工具,在开始看老师推荐的资料时,我看着许多代码有点无从下手,但在接触过后,我体验到了它功能的强大。尤其是目录的生成和表格的绘制,原本复杂的它们在用 LaTeX 后,仅仅需要简短的代码就可以实现自动生成。随着对 LaTeX 的深入学习,我渐渐对它产生了一些兴趣,即使还有许多的内容需要我继续探索,但在掌

握基础后，我不再感觉它是一个难以驾驭的系统。我也相信，它在我今后的学习中会成为帮助我提升自己的有用工具。

无论是 Git 还是 LaTeX，在最开始接触的时候，总会给我一种难以掌握的感觉，但在一点点的了解与练习后，我发现它们并没有我想象的那么难，所以在学习新的东西时，我们不能还没有了解就想退缩，要努力尝试与实践，再复杂的事物也可以在付出中被掌握，更何况在没有接触的情况下，我们不知道它是不是真的不简单。而且更让人开心的是，在学习 Git 和 LaTeX 后，我获得了成就感，我也会在未来继续两种系统的学习，让它们成为我学习路上的实用工具。