

# 课程报告一：Git 与 Latex

杜培绪

2024 年 9 月 10 日

## 题目 1.

克隆 github 上的远程仓库，将版本历史可视化并进行探索，检查是谁最后修改了 README.md 文件，以及最后一次修改时的提交信息是什么

### 解决方法：

首先需要选择要复制到的文件夹，打开 git，输入 `git clone` 远程仓库网址，克隆完毕后即可在当前文件夹下看到克隆下来的远程仓库了。

使用 `git log -n` 文件名命令，即可查看指定文件前 `n` 次的修改记录。输入 `git log -1 README.md` 命令，就能查看最后修改 README.md 文件的人。

为了查看最后一次修改的提交信息，需要输入 `git blame __config.yml | grep collections` 命令，获取 config.yml 文件中 collections 行的修改情况，并使用 `git show -pretty=format:"%s" a88b4eac | head -1` 进行输出。

## 题目 2.

使用 Git 时的一个常见错误是提交本不应该由 Git 管理的大文件，或是将含有敏感信息的文件提交给 Git。尝试向仓库中添加一个文件并添加提交信息，然后将其从历史中删除

### 解决方法：

首先，向仓库中添加一个文件。输入 `echo "password123">my_password` 即可创建一个名为 my\_password 的文件，并在里面写入 password123。再使用 `git add .` 命令跟踪所有文件，这样这个文件就进入了暂存区。再使用 `git commit` 命令将其提交到仓库中。之后使用 `git filter-branch` 命令即可将

提交记录删除。

### 题目 3.

与其他的命令行工具一样，Git 也提供了一个名为 `/.gitconfig` 配置文件 (或 dotfile)。请在 `/.gitconfig` 中创建一个别名，使您在运行 `git graph` 时，您可以得到 `git log -all -graph -decorate -oneline` 的输出结果

#### 解决方法：

使用 `cd .gitconfig` 命令进入 `/.gitconfig`，在其中创建一个别名。打开 vim 编辑窗口，输入代码

```
[alias]graph = log --all --graph --decorate --oneline
```

即可完成别名的创建，此时可以用 `graph` 代替 `log -all -graph -decorate -oneline`。

### 题目 4.

您可以通过执行 `git config --global core.excludesfile /.gitignore_global` 在 `/.gitignore_global` 中创建全局忽略规则。配置您的全局 `gitignore` 文件来自动忽略系统或编辑器的临时文件，例如 `.DS_Store`

#### 解决方法：

需要先创建一个 `.gitignore` 文件，用 vim 打开编辑窗口，在其中加入后缀 `.DS_Store`，即可自动忽略这一类文件。

接下来尝试用 `echo` 创建一个后缀为 `.DS_Store` 的文件，用 `git status` 查看，会发现这个文件被自动忽略，无法查看。

### 题目 5.

如果提交了错误的信息，该如何撤销提交到仓库里的信息？

#### 解决方法：

想要撤销提交的信息，需要将仓库返回到提交前的状态。可以使用命令行

```
git reset --soft HEAD^n
```

将仓库重置为上一次提交前的状态。

## 题目 6.

如何修改修改提交的历史信息？

### 解决方法：

如果只是修改前一次的提交信息，则只需要使用命令行

```
git commit -amend
```

再用 vim 命令打开文件进行编辑即可。

修改多次之前的提交信息较为复杂，首先我们需要用命令行

```
git log -n
```

查看前 n 次的提交信息，找到想要更改的内容。假设需要修改第 5 次的提交信息，就需要用

```
git rebase -i HEAD 5
```

将前 i 项提交信息倒序排列出来。之后将第一行的 pick 改为 edit，输入：wq 退出编辑。之后执行命令

```
git commit -amend
```

进入 vim 编辑器修改并提交修改信息。最后使用命令

```
git rebase -continue
```

即可完成更改。

## 题目 7.

如何删除文件？

### 解决方法：

例如我们需要删除库中的 my.txt 文件，需要先输入命令

```
rm my.txt
```

此时我们使用 `git status` 和 `git list` 查看，会发现工作区多了一个删除命令，且库中的 `my.txt` 并没有被删除。接下来需要输入

```
git add .
```

命令行将删除命令保存到暂存区。再次查看，会发现该文件已被删除。

### 题目 8.

如果不小心对暂存区的文件进行了修改，但没有将其提交到暂存区，如何确定修改内容？

#### 解决方法：

可以输入命令行

```
git diff
```

这个命令会输出工作区和暂存区的差异，还会列出删除了什么内容、增加了什么内容。

这个函数还有其他功能，如果在后面加上一个参数，变为

```
git diff HEAD
```

就可以比较与版本库中内容的差异。

### 题目 9.

如何把本地仓库的内容推向一个空的远程仓库？

#### 解决方法：

要将本地仓库内容推向远程仓库，首先要将二者连接起来。使用命令行

```
git remote add origin ...
```

其中的... 为远程仓库的地址。如果是首次推送，需要使用命令

```
git push -u origin master
```

之后的推送就可以去掉-u。

**题目 10.**

回退命令 `git reset` 可以加上哪些参数，分别有什么效果？

**解决方法：**

*`git reset --soft`*

这个命令会将仓库回退到某一个版本，并保留工作区和暂存区的内容。

*`git reset --hard`*

这个命令会在回退的同时删除工作区和暂存区的内容。

*`git reset --mixed`*

这一命令介于上面二者之间，保留工作区内容，但是会删除暂存区内容。

**题目 11.**

如何查看分支上提交的历史记录？

**解决方法：**

查看分支提交的历史记录，需要使用

*`git reflog -n`*

其中 `n` 表示查看前 `n` 条历史记录，如果没有这个参数，则默认查看所有记录。还可以使用

*`git reflog -n --pretty=oneline`*

查看简化的历史记录。

**题目 12.**

在工作中需要了解提交记录的提交者，如何设置自己的用户名并标明邮箱？

**解决方法：**

首先创建用户名，使用命令行

```
git config --global user.name "Name"
```

修改引号中的 Name 即可修改自己的用户名。修改邮箱同理，使用命令行

```
git config --global user.email "Email"
```

修改 Email 内容，将其作为自己的邮箱。

**题目 13.**

“git pull” 和 “git fetch” 之间有什么区别？

**解决方法：**

这两条命令都可以从远程仓库更新本地仓库，但也存在区别。

```
git pull
```

能够从远程仓库下载到本地仓库的最新内容，但不会自动合并到当前的工作分支中。这个命令会获取远程仓库所有分支的最新提交。

```
git fetch
```

可以查看远程仓库中的所有更新，但不会影响本地的工作状态。

**题目 14.**

创建仓库有哪些方法？

**题目 15.**

可以先选择或创建一个文件夹，用 git 打开后输入命令

```
git init
```

就可以把当前文件夹作为仓库进行使用。

另一种方法是克隆一个远程仓库，使用克隆命令

```
git clone
```

即可完成。

### 题目 16.

Latex 中不识别空格，怎样添加空格？

**解决方法：**

$$\backslash enspace$$

这个代码可以输出小空格。

$$\backslash quad$$

这个命令能够输出大空格。

### 题目 17.

如何在 Latex 中设计页面参数？

**解决方法：**

页面参数包含纸张宽度、上边缘高度、边注与版心、页眉高、页脚基线与正文最后一行基线的间距等数据，这些数据都可以用 `\setlength` 来完成。例如

$$\backslash setlength\textit{textheight}\{7in\}$$

这个代码可以设定版心的宽度。

还可以使用 `geometry` 宏包，它为我们提供了一个相对简单的接口。例如

$$\backslash usepackage[top = 30mm, bottom = 30mm, inner = 20mm, outer = 45mm]\{geometry\}$$

这个宏包让我们可以一次性设置多个数据。

### 题目 18.

如何在 Latex 中输入一些特殊符号？

**解决方法：**

Latex 中很多符号有特殊含义，不能直接输入，但可以通过其他方式输入。例如 {、}、%、#、\_。这些符号的输入有着相同的规则，即在前面加上一个 “\”。但反斜杠的输入比较特殊，需要使用

$$\backslashtextbackslash$$

进行输入。

### 题目 19.

标题等首页内容的设计方法。

**解决方法：**

$$\title{}$$

用以设置标题。

$$\author{}$$

用以设置作者，括号中用 \and 分割多个作者。

$$\date{}$$

用以设置日期。

### 题目 20.

如何在 Latex 中控制格式？

**解决方法：**

在导言部分用 \newenvironment 语句设置正文模块，之后就可以在正文中用 \begin 和 \end 语句使用改模块。例如本次作业使用的 solution 模块，

$$\newenvironment{solution}{\par\noindent\textbf{}}{\par}$$

其次是控制首行缩进。首先引用宏包

$$\usepackage[indentfirst]$$



