

实验一 Git和Markdown基础

班级：21计科1

学号：B20210302123

姓名：何香仪

Github地址：<https://github.com/deliciousbeef>

实验目的

Git基础，使用Git进行版本控制 Markdown基础，使用Markdown进行文档编辑 实验环境 git VSCode VSCode插件

实验内容和步骤

第一部分 实验环境的安装 安装git，从git官网下载后直接点击可以安装：git官网地址 从Github克隆课程的仓库：课程的仓库地址，运行git bash应用（该应用包含在git安装包内），在命令行输入下面的命令（命令运行成功后，课程仓库会存放默认在Windows的用户文件夹下）git clone

https://github.com/zhoujing204/python_course.git 如果你在使用命令时遇到SSL错误，请运行下面的git命令（这里假设你的Git使用了默认安装目录）：git clone

git config --global http.sslCAInfo "C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt" 或者运行下面的命令：

git config --global http.sslVerify false 如果遇到错误：，请运行下面的命令重新指定git的安全证书：error setting certificate file

git config --global --unset http.sslCAInfo git config --global http.sslCAInfo "C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt" 该仓库的课程材料后续会有更新，如果需要更新课程材料，可以在本地课程仓库的目录下运行下面的命令：

git pull 在本地的仓库内容有更新后，可以运行下面的命令，将本地仓库的内容和远程仓库的内容同步：

git push origin main 注册Github账号或者Gitee帐号，创建一个新的仓库，使用上面同样的方法将该仓库clone到本地，用于存放实验报告和实验代码，使用和命令保持远程仓库和本地仓库的同步。git pullgit push 安装VScode，下载地址：Visual Studio Code 安装下列VScode插件 GitLens Git Graph Git History Markdown All in One Markdown Preview Enhanced Markdown PDF Auto-Open Markdown Preview Paste Image markdownlint 第二部分 Git基础 教材《Python编程从入门到实践》P440附录D：使用Git进行版本控制，按照教材的步骤，完成Git基础的学习。

第三部分 learngitbranching.js.org 访问 learngitbranching.js.org，如下图所示完成Main部分的Introduction Sequence和Ramping Up两个小节的学习。

[Learngitbranching.js.org](https://learngitbranching.js.org)

上面你学习到的git命令基本上可以应付百分之九十以上的日常使用，如果你想继续深入学习git，可以：

继续学习 learngitbranching.js.org 后面的几个小节（包括Main和Remote） 在日常的开发中使用git来管理你的代码和文档，用得越多，记得越牢 在git使用过程中，如果遇到任何问题，例如：错误删除了某个分支、从错误

的分支拉取了内容等等，请查询git-flight-rules

实验过程与结果

第一题Git commit

```
git commit  
git commit
```

第二题Git branch

```
git branch bugFix  
git checkout bugFix
```

第三题Git merge

```
git checjout - b bugFix  
git commit  
git checkout master  
git commit  
git merge bugFix
```

第四题Git Rebase

```
git checkout -b bugFix  
git commit  
git checkout master  
git commit  
git checkout bugFix  
git rebase master
```

第五题 分离HEAD

```
git checkout C4
```

第六题 相对引用

```
git checkout bugFix  
git checkout HEAD^
```

第七题 相对引用2

```
git branch -f master C6
git branch -f bugFix C0
git checkout C1
```

第八题 撤销变更

```
git reset HEAD^
git checkout pushed
git revert HAED
```

实验考查

- 1.什么是版本控制？使用Git作为版本控制软件有什么优点？ a:Git有能力高效管理类似Linux内核一样的超大规模项目； b:Git实现了离线开发、代码审核特性，解决了跨地域协同开发中代码质量和编码协同的问题； c:分支管理功能强大，便于查询和追溯分支间的提交历史；
- 2.如何使用Git撤销还没有Commit的修改？如何使用Git检出（Checkout）已经以前的Commit？我们可用：git checkout. 就可以清除所有变更内容；
- 3.Git中的HEAD是什么？如何让HEAD处于detached HEAD状态？ Git 中的 HEAD 可以理解为指针，指向当前仓库所处的分支。一般在有 Git 管理的目录下打开 Git 终端都能在当前路径的尾巴上，看到所处的分支名。
- 4.什么是分支（Branch）？如何创建分支？如何切换分支？ a.创建新分支 git branch 新分支名称； b.切换分支 git checkout 分支名称； c.创建分支的同时，切换到该分支上 git checkout -b 新分支名称；
- 5.如何合并分支？git merge和git rebase的区别在哪里？ git merge和git rebase都是用来合并分支的命令，但是两者合并的方式不同。 git merge是将两个分支的修改合并到一个新的提交中，而git rebase则是将当前分支的修改放在目标分支的最新提交后面，使得提交历史更加线性、清晰。使用merge会保留原有分支的提交历史，使用rebase则会改变原有分支的提交历史。
- 6.如何在Markdown格式的文本中使用标题、数字列表、无序列表和超链接？ a.无序列表使用星号(*)、加号(+)或是减号(-)作为列表标记，这些标记后面要添加一个空格，然后再填写内容; b.有序列表使用数字并加上. 号来表示

实验总结

通过本次实验，我学会了如何配置python和git环境、如何使用git命令，并且学会了如何在VScode中使用Markdown插件，在github中学会创建自己的仓库。