#实验一 Git和Markdown基础
班级： 21计科1
学号： B20210302131
姓名： 李佳琪
Github地址：https://github.com/Seven116

##实验目的
1、Git基础，使用Git进行版本控制
2、Markdown基础，使用Markdown进行文档编辑
##实验环境
1、Git
2、VSCode
3、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，课程仓库会默认存放在Windows的用户文件夹下） git clone https://github.com/zhoujing204/python_course.git 如果你在使用git clone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 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：error setting certificate file，请运行下面的命令重新指定git的安全证书：

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git pull和git push命令保持远程仓库和本地仓库的同步。 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。

##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 checkout -b bugFix
git commit
git checkout main
git commit
git merge bugFix
```

###第四题Git Rebase

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

###第五题分离HEAD

```
git checkout C4
```

###第六题相对引用

```
git checkout bugFix
```

###第七题相对引用2

```

```

```
git branch -f main C6
git checkout HEAD^
git checkout -f bugFix HEAD~1
```

###第八题撤销变更

```
git checkout pushed
git revert HEAD
```

##实验考查

请使用自己的语言回答下面的问题，这些问题将在实验检查时用于提问和答辩，并要求进行实际的操作。

- **什么是版本控制？使用Git作为版本控制软件有什么优点？**
  版本控制是一种软件工程技巧，籍以在开发的过程中，确保由不同的人所编辑的同一档案都得到更新。
- **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？（实际操作）**
  如果只是想撤销对某个文件的修改，可以使用命令 git checkout -- 文件路径。这会将文件还原为最近一次提交时的状态。如果想撤销对所有文件的修改，可以使用命令 git reset --hard HEAD。这会将工作区和暂存区的修改全部撤销，回到最近一次提交时的状态。可以使用命令 git log 查看提交历史，并获取要检出的目标提交的哈希值。使用命令 git checkout 提交哈希值 检出以前的提交。
- **Git中的HEAD是什么？如何让HEAD处于detached HEAD状态？（实际操作）**
  HEAD指的就是 .git/HEAD 文件，它存储着当前working directory所处的某次commit。首先，使用 git log 命令查看当前仓库的提交历史，并找到要指向的提交哈希值。然后，使用命令 git checkout commit哈希值 将"HEAD"指向指定的提交。现在，"HEAD"处于分离的HEAD状态，可以进行查看和修改操作，这些更改不会自动添加到任何分支上。如果要保留更改，可以使用 git checkout -b 新分支名 命令创建一个新的分支，将分离的HEAD状态的更改添加到该分支上。
- **什么是分支（Branch）？如何创建分支？如何切换分支？（实际操作）**
  分支（Branch）是Git中用于并行开发和管理代码的重要概念。每个分支都是Git仓库中的一个独立的副本，在分支上可以进行不同的开发工作而不影响其他分支。 创建分支：git branch 新分支名称 切换分支：git checkout 分支名称 创建分支的同时，切换到该分支上：git checkout -b 新分支名称
- **如何合并分支？git merge和git rebase的区别在哪里？（实际操作）**
  在Git中，合并分支可以使用git merge和git rebase两个命令.rebase 和 merge 都是在 Git 中用来合并分支的命令。 merge 会在两个分支上创建一个新的合并节点，将两个分支合并在一起。 这样会导致历史记录中出现大量的合并节点，使得历史记录变得很杂乱。 rebase 则是将一个分支的提交记录“放到”另一个分支的最后面，使得两个分支的历史记录看起来更加连续。
- **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？（实际操作）**
  在文本前加上（#）作为标题分级，Markdown 支持有序列表和无序列表。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.使用方括号（[]）来包裹链接的显示文本，使用圆括号（()）来包裹链接的URL.

  ##实验总结

  总结一下这次实验你学习和使用到的知识，例如：编程工具的使用、数据结构、程序语言的语法、算法、编程技巧、编程思想。

  本次实验主要学习了实验环境的安装配置，完成了Git基础的学习，以及对Markdown基本运用有了一定

了解，学会使用Markdown进行文档编辑，并运用gitup仓库来管理自己的实验报告及代码，大幅提高了学习效率。