实验一 Git和Markdown基础

班级： 21计科3

学号：B20210302324

姓名： 唐佳喜

Github地址：https://github.com/Jecxy/Python_Learning

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

1. 安装git，从git官网下载后直接点击可以安装：git官网地址
2. 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
```

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
```

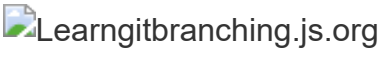
3. 注册Github账号，创建一个新的仓库，用于存放实验报告和实验代码。
4. 安装VScode，下载地址：Visual Studio Code
5. 安装下列VScode插件
    - GitLens
    - Git Graph
    - Git History
    - Markdown All in One
    - Markdown Preview Enhanced
    - Markdown PDF
    - Auto-Open Markdown Preview
    - Paste Image
    - markdownlint

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。



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

第四部分 Markdown基础

查看Markdown cheat-sheet，学习Markdown的基础语法

使用Markdown编辑器（例如VScode）编写本次实验的实验报告，包括实验过程与结果、实验考查和实验总结，并将其导出为 **PDF格式** 来提交。

实验过程与结果

python测试：

```
print("Hello Git World!")
print("Hello everyone.")
print("Oh no, I broke the project!")
```

显示效果如下：

```
Hello Git World!
Hello everyone.
Oh no, I broke the project!
```

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。

1. 什么是版本控制？使用Git作为版本控制软件有什么优点？

   版本控制是对软件开发过程中各种程序代码、配置文件及说明文档等文件变更的管理，是软件配置管理的核心思想之一。Git简洁、高效，分支管理功能强大，能够管理大规模项目，且支持离线开发。

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？（实际操作）

   使用git reset撤销已经add在缓存区的文件。git checkout可以切换分支，也可以切换到某个commit的版本。

3. Git中的HEAD是什么？如何让HEAD处于detached HEAD状态？（实际操作）

   HEAD指向最新提交的版本，当HEAD处于detached HEAD状态时，HEAD指向的提交为最新提交的父提交，此时HEAD指向的提交不是任何分支。创建临时branch temp来保存commit； merge临时branch temp到target branch; 删除临时branch temp。

4. 什么是分支（Branch）？如何创建分支？如何切换分支？（实际操作）

   分支是用来标记特定代码的版本。创建分支的命令是git branch name，切换分支的命令是git checkout name。

5. 如何合并分支？git merge和git rebase的区别在哪里？（实际操作）

   git merge和git rebase都是用来合并分支的，git merge是合并两个分支，git rebase是合并两个分支的commit。git rebase的合并方式是直接在当前分支上修改代码，git merge的合并方式是创建一个新的commit。

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？（实际操作）

标题：使用#号+空格+标题

数字列表：1. + 空格 + 标题

无序列表：*/+/- + 空格 + 标题

多级列表：每深入一级，句首多加个Tab键

链接：使用格式

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。

总结：在本次实验中，我学习了Git的基本使用，包括Git的安装、Git的常用命令、Git的分支管理、Git的冲突解决、Git的合并、Git的撤销、Git的检出等。同时，我也学习了Markdown的基本语法，包括标题、数字列表、无序列表、超链接等。