

1 理论及命令

1.1 说明 git、gitee、GitHub 三者之间的关系

1.2 写出如下 git 命令

2 操作题

2.1 课上练习题

2.1.1 练习1

2.1.2 练习2

2.1.3 练习3

2.1.4 练习

2.1.5 练习5

1 理论及命令

1.1 说明 git、gitee、GitHub 三者之间的关系

1.2 写出如下 git 命令

- 初始化本地仓库: `git init`
- 查看状态: `git status`
- 添加到暂存区: `git add 文件名`
- 提交到本地仓库: `git commit -m "提示消息" 文件名` **生成历史版本**
- 查看历史版本: `git reflog`
- 版本穿梭: `git reset --hard 版本号`
- 创建分支: `git branch 分支名`
- 查看分支: `git branch -v`
- 切换分支: `git checkout 分支名`
- 合并分支: `git merge 分支名`
- 本地仓库推送到远程仓库: `git push 别名|远程仓库地址 分支名`
- 克隆远程仓库到本地: `git clone 别名|远程仓库地址 分支名`
- 拉取远程仓库代码到本地: `git pull 别名|远程仓库地址 分支名`

2 操作题

2.1 课上练习题

2.1.1 练习1

1. D盘下创建目录 egg1;
2. 在该目录下打开 `Git Bash` 并初始化本地仓库;
3. 创建文件 `eg1.txt`, 并任意输入内容;
4. 查看状态;
5. 将文件添加到暂存区并查看状态;
6. 提交到本地仓库;
7. 修改 `eg1.txt` 文件内容并查看状态;
8. 添加到暂存区并查看状态;
9. 提交到本地仓库。

2.1.2 练习2

1. D盘下创建目录: `eggit2` , 并初始化仓库;
2. 创建文件 `eg2.txt` , 并查看状态;
3. 提交到暂存区并查看状态;
4. 提交到本地仓库并查看状态;
5. 修改文件内容并再次提交到暂存区和本地仓库;
6. 查看提交版本信息;
7. 穿梭到第一次提交的版本并查看文件内容确认;
8. 穿梭到第二次提交的版本并查看文件内容确认。

2.1.3 练习3

1. D盘下创建目录: `eggit3` 并初始化仓库;
2. 创建文件 `eg3.txt` 并存入任意内容;
3. 添加到暂存区, 并提交到本地仓库;
4. 创建新的分支 `feature-game` , 并查看分支;
5. 切换到分支 `feature-game` , 并修改文件内容【只允许在下面新添加内容, 不允许修改原内容】;
6. 添加到暂存区, 并提交到本地仓库, 查看内容;
7. 切换到分支 `master` , 并查看文件内容确认。
8. 将 `feature-game` 分支合并到 `master` 分支

2.1.4 练习

1. D盘目录下创建目录: `eggit4` 并初始化本地仓库;
2. 创建文件 `eg4.txt` , 输入内容: `abcdefg` , 并添加到暂存区提交到本地仓库;
3. 创建分支 `hot-fix` ;
4. 修改 `master` 分支文件内容, 在第二行添加内容 `master-666` ;
5. 添加到暂存区并提交到本地仓库;
6. 切换到 `hot-fix` 分支, 修改文件内容, 在第二行添加 `hot-fix-888` ;
7. 添加到暂存区并提交到本地仓库;
8. 切换到 `master` 分支, 将 `hot-fix` 分支合并到 `master` 分支中;
9. 解决合并冲突。

2.1.5 练习5

- 第1步: 创建一个自己的远程仓库并设置开源, 名字自定义;
- 第2步: 将自己电脑中的工程 `jsd2311-ssm` 推送到自己的远程仓库中;
- 第3步【选做】: 回到家后, 使用另一台电脑将工程克隆到本地电脑。