## Git命令行操作

### Git提交代码操作

1. ssh-keygen -t rsa -C “注册邮箱”：生成ssh密钥，到生成的文件里粘贴
2. 创建一个远程仓库Create a new repository
3. git init ：初始化本地仓库
4. git status ：查看工作树

// committed(已提交)、modified(已修改)、staged(已暂存)

1. git add . ：将目录下的所有文件上传

// 可以将“.”换成具体的文件名 或者“-A”

1. git commit -m “注释” ：提交已暂存的文件到本地仓库
2. git remote add origin [https://github.com/Silence/WorkSpace.git](https://github.com/SilenceLoveLt/WorkSpace.git)

：关联远程仓库，https的地址就是远程仓库地址

1. git remote rm origin ：git删除远程地址
2. git pull origin master ：拉取远程仓库的远程分支，并合并到当前分支。

//此处如果报错：git pull origin master --allow-unrelated-histories

1. git push -u origin master(默认的本地分支名) ：上传代码到远程仓库

### Git解决冲突操作

1. git fetch ：下载所有远程分支的最新代码
2. git rebase origin/master ：以origin/master分支为基线，合入master分支的修改到origin/maste
3. 手动解决代码冲突：<<<<<<< HEAD冲突发生前，没有进行合并时，当前位置的代码 ======= 合并进来的冲突代码 >>>>>>>
4. git add -A ：冲突解决完成之后，提交修改
5. git rebase --continue ：处理完冲突继续合并，合并的过程中，还有可能产生冲突，解决方法同上。
6. git rebase --abort ：取消合并
7. git push origin master ：冲突解决完之后，推送到远端服务器

### Git创建分支合并分支操作

1. git branch ：查看当前所有分支列表
2. git branch -a ：查看远程版本库分支信息
3. \* master ：代表现在检出的那个分支，这就意味着如果这时提交，master分支将会随着新的工作向前移动
4. git branch --marged ：查看哪些分支已经合并到当前分支
5. git branch --no-marged ：查看所有包含未合并工作分支
6. git checkout -b newBranch ：新建分支newBranch并进入分支

// 相当于 git branch newBranch ,git checkout newBranch

1. vim index.html ：创建一个index.html文件
2. git add index.html ：添加到 newBranch 的暂存区
3. git commit -m “add a new index.htm”：提交到本地分支仓库
4. git merge newBranch ：合并newBranch分支到主分支上master
5. git log --graph --pretty=oneline --abbrev-commit ：查看分支合并情况
6. git branch -d newBranch ：删除已经合并之后的newBranch 分支

// 如果该分支包含了未合并的的工作则删除时会报错

总结：所有的平行分支操作创建、合并都是发生于本地的Git版本库，没有与服务器发生交互。平行分支新添加的文件与主分支合并之后，同时会把文件添加到主分支里。

1. 进入合并注释命令操作：
2. i进入insert操作
3. 添加注释信息
4. 按键盘Esc保存当前操作
5. 输入“:wq”回车键退出

### Git撤销push代码操作

——文件还未提交到暂存区，只是在工作目录中修改

1. git checkout -- 文件 ：撤销该文件的工作区的修改状态
2. git checkout -- . ：撤销工作区中所有文件的修改

——文件add到暂存区，还没有提交到本地仓库

1. git reset head 文件 ：撤销该文件从暂存区回滚到修改状态
2. git reset head . ：撤销暂存区的所有内容

——文件提交到本地仓库，还没有push到远程仓库

1. git log ：查看提交信息，以便获取需要回退至的版本号
2. git reset --soft <回退到的commit版本号> ：重置到指定的版本提交，撤销push

// git reset --soft ：不会改变stage(暂存区),仅仅将commit回退到指定的提交

// git reset --mixed ：不会改变工作区，但是会用指定的commit覆盖stage区，之前所有暂存的内容都变成了暂存的状态

// git reset --hard ：使用指定的commit的内容覆盖stage区和工作区

// git reset --hard HEAD^：撤销所有本地到上一次修改,回退到上一个版本

1. git push origin master --force ：强制提交当前版本号
2. git reflog ：查看当前项目中commits（版本号）
3. git checkout <commits\_id> <file\_path> ：检出指定文件到本地仓库中（远程仓库里仍保持上一次提交操作）

### git版本回退

1. git revert c1 ：撤销某一个提交如c1,(可以有多个提交)
2. git push ：然后再执行推送到远程仓库，起到回退的作用
3. git log ：查看提交历史记录
4. git reset --hard head^ ：回退到上一个版本 head^^回退到上上的版本
5. git reset --hard 指定未来的版本号 ：回退到未来的某个版本
6. git reflog ：查看当前项目中所有命令历史记录的commits（版本号）

### 命令行常用操作

1. makdir demo ：创建demo文件夹
2. echo “文本内容”>> text.txt ：给text文本输入内容
3. cat text.txt ：查看text文本内容