# Git

## 1.本地操作

1. git init 创建git仓库
2. git add . 添加文件，可以只添加一个文件，添加到暂存区
3. git commit –m “xx” 向git仓库提交
4. git reset –hard “commit\_id” 回退/进版本 HEAD 指向当前版本 HEAD^指向上一个版本
5. git log –pretty=oneline 查看提交历史
6. git reflog 查看历史命令 可以查看以后的版本号
7. git status 查看当前状态
8. git diff HEAD --xx.js 查看与版本库中最新的xx.js的区别
9. git checkout -- xx.js 丢弃工作区的修改/恢复到HEAD

## 2.创建远程

1. 在github创建仓库，new respo ，取名字xxx
2. git remote add origin [git@github.com:13164286167/xxx.git](mailto:git@github.com:13164286167/xxx.git) 关联本地与远程仓库
3. ssh-keygen -t rsa -C "13164286167@163.com" 创建秘钥 在c:\User\Admain\*\.ssh中会有.Pub生成 在github中创建秘钥即可关联
4. git push –u origin master （第一次push的时候 加上-u参数）
5. git clone [git@github.com:13164286167/xxx.git](mailto:git@github.com:13164286167/xxx.git) 即可克隆代码

## 3.创建分支

1. 查看分支：git branch
2. 创建分支：git branch <name>
3. 切换分支：git checkout <name>
4. 创建+切换分支：git checkout -b <name>
5. 合并某分支到当前分支：git merge <name>
6. 删除分支：git branch -d <name>
7. 合并分支禁用快速合并 git merge - -no-ff –m ”message” dev
8. 暂存工作现场： git stash 恢复：git stash pop 查看：git stash list

## 4. 多人协作

1. 查看远程库信息，使用git remote -v；
2. 本地新建的分支如果不推送到远程，对其他人就是不可见的；
3. 从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；
4. 在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；
5. 建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；
6. 从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

1.本地添加新的仓库  
可以在vscode左下角添加或者git checkout –b “名称”

2.往远程仓库添加新的分支

git push --set-upstream origin tangbooBranch

3.有自己的分支需要提交代码时

git add .

git commit –m “xxx”

git checkout master

git pull

git merge tangboBranch 有冲突就改，改完再往仓库提交

最后push

## 5.创建ssh

1.查看是否已经有ssh key

1)cd ~/ssh 进入ssh

2)ls 查看是否有id\_rsa.pub 或 id\_dsa.pub

2.创建ssh key

ssh-keygen -t rsa -C "your\_email@example.com"

3.粘贴ssh key

clip < ~/.ssh/id\_rsa.pub

4.进入github中setting ssh添加即可

## 6.关于github

当提交远程github没有记录点的时候，是因为当前邮箱与远程邮箱没有对应，修改为相同邮箱就有记录了。

## 7.功能点

### 7.1本地git迁移

1. 远程仓库创建
2. 打开本地git bush
3. git remote rename origin old-origin
4. git remote add origin xxx.git
5. git push -u origin --all
6. git push -u origin --tags

7.2取消git add . git commit –m “xxx”

**git reset --head HEAD~1**

## --mixed

意思是：不删除工作空间改动代码，撤销commit，并且撤销git add . 操作

这个为默认参数,git reset --mixed HEAD^ 和 git reset HEAD^ 效果是一样的。

## --soft

不删除工作空间改动代码，撤销commit，不撤销git add .

## --hard

删除工作空间改动代码，撤销commit，撤销git add .

注意完成这个操作后，就恢复到了上一次的commit状态。

### 顺便说一下，如果commit注释写错了，只是想改一下注释，只需要：

git commit --amend