从本地创建仓库：

git init ->ls –ah 可查看到.git文件

git add [file] ->git add readme.txt

git commit –m [message] ->git commit –m “first commit”

git log (--pretty=oneline) 查看log

git reset --hard commit\_id -> git reset --hard HEAD^ {HEAD^ 特指最后版本的上一个版本} 删除git commit/add 以及本地添加的代码

git reset –-soft HEAD^ 撤销commit 不撤销add 本地

git reset -–mixed HEAD^ 撤销commit add 不撤销本地

git commit -–amend 修改注释

git reflog 查看命令历史

git diff [commit\_id] –- (file) ->查看与commit\_id的区别/git diff 查看当前所有文件和head的区别

git status 查看当前所有文件的状态，add还是commit

git checkout –- [file] 将工作区中的修改作废，滚退到暂存区或者最新版本（取决于有没有git add）

git reset HEAD -- [file] 取消git add [file]的作用

git remote add [git name] [git address] ->[git name]成了[git address]的别名

git push (-u) [git name] [branch] –>推送到github branch上，加上-u表示第一次推送

git clone [git address] 克隆

git branch [branch name] 创建本地分支

git checkout [branch name] 切换本地分支

git branch –d(-D) [branch name] 删除本地分支

git checkout –b [branch name] 创建并切换本地分支

git merge [branch name] 合并分支

git branch (-a） 查看分支本地（本地和远端）

git stash 将当前分支放到stash区

git stash list 查看当前stash区有哪些分支

git stash pop 取出stash分支并在stash区删除

git cherry-pick <commit> 当前分支合并某个commit

git checkout -b branch-name origin/branch-name 从远端拉取分支

git branch --set-upstream branch-name origin/branch-name 将本地分支和远端分支关联

git show commit\_id 显示log详细内容

git checkout commit\_id [file] 找回被删除的文件

git push origin --delete [BranchName] 删除远端分支

git log –all –oneline 查看所有分支及已经提交过的log

git 设置SSH密钥

打开git 命令行

cd ~/.ssh

ssh-keygen -t rsa -C "your email"

打开复制id\_rsa.pub里的内容

复制到github ssh中