mkdir 创建新文件夹

cd 进入路径

pwd 打印当前工作路径

git init 初始化仓库

ls -a 显示所有文件

git add readme.txt 添加到暂存区

git commit -m “comment” 提交到本地仓库

git diff 显示不同

git status 显示状态

git log/ git log --pretty=oneline 显示commit操作记录

git reset --hrad HEAD^/ git reset --hard HEAD~100 回退版本

git reset --hard ***0e62(id)*** 前进版本

git reflog 显示head的变化

git reset HEAD readme.txt 撤销add

git checkout -- readme.txt 撤销本地（从仓库拿一遍覆盖）

git remote add origin [git@github.com:Ian-Salvatore/learngit.git](mailto:git@github.com:Ian-Salvatore/learngit.git) (本地连接空远程仓库)

git push -u origin master (第一次推往空仓库)

git push origin master (普通推送操作)

git clone [git@github.com:Ian-Salvatore/gitskills.git](mailto:git@github.com:Ian-Salvatore/gitskills.git) (从远程仓库直接克隆)

git checkout -b dev (创建并切换到dev)

git branch (显示所有分支)

git checkout master (切换到分支master)

git merge dev (在master分支中合并dev/ 合并某分支到当前分支)

git branch -d dev (删除分支dev)

git branch -D dev (强制删除分支dev)

合并冲突，冲突版本只在本地。手工解决后，还要add再commit才算完成

git log --graph --pretty=oneline --abbrev-commit

git merge --no-ff -m “comment” dev (带分支记录的合并)**推荐**

git stash (先进后出原理)

git stash list

git stash apply/ git stash apply stash@{1}

git stash drop/ git stash drop stash@{3}

git stash pop

git checkout -b dev origin/dev (从远程仓库拿分支)

git branch --set-upstream-to origin/dev (与远程分支建立连接)

先git pull拉取最新，然后处理冲突，最后合并，上传

git tag v1.0 git tag -d v1.0

git tag v1.0 ***0e62(id)***  git push origin v1.0/--tags

git tag -a v1.0 -m “version 1.0 released” ***0e62(id)***  git push origin :refs/tags/v1.0

git tag

git show v1.0