基本使用方式

1、$ git config --global user.name tanger // 设置名字

2、$ git config --global user.email [tangerluo@qq.com](mailto:tangerluo@qq.com) // 设置邮箱

3、$ git clone 网址 // 获取开源项目

4、$ git init // 在该目录下创建.git代码仓库文件夹，初始化

5、$ git add . // 将本文件夹（在哪里打开的Bash就是哪一级文件夹）下的所有文件及文件夹设置为待提交状态

6、$ git commit -m”备注” // 提交，备注一定要写

7、$ git log // 看日志，commit后面的字符串为每一次提交的标识号

8、$ git checkout HEAD 文件名.后缀 // 从最后一次提交中恢复某文件

9、 $ git add 文件名.后缀 // git add相当于已经提交到缓冲区了，再对原文件修改不影响，除非再add一次

$ git commit -m”备注” // 仅提交目标文件并进行备注

注：每次提交代码应是清晰的，即每完成一个功能就提交，而不是完成了一个后又完成了另一个功能的一半进行提交



重置功能

git revert命令会恢复到指定的提交，但会保留代码库中其他所有提交的历史，并为恢复的改动创建一个新的提交。

另外，git reset命令会恢复到指定的提交，然后删除指定提交之后的所有提交。

$ git revert [commit ID]

$ git reset --hard [commit ID]

忽略文件：



分支



$ git branch 分支名 // 创建分支

$ git checkout 分支名 // 切换分支

$ git merge 分支名 // 将该分支与目标分支合并。若目标分支与该分支下有文件冲突，需要使用git diff查找冲突，并将该分支下的对应文件回退或更新，重新同步

$ git rebase 分支名 // 将该分支变基，起点为目标分支最后一次的同步点

$ git cherry-pick <commit id> // 单独合并其他分支的一个提交

$ git remote add origin 远程仓库地址

$ git push -u origin 分支名

