1.创建版本库

在项目文件夹内，执行: git init

2.提交文件

新建文件后，通过git status 进行查看文件状态

将文件添加到暂存区 git add 文件名

提交文件到本地库 git commit

编写注释 ，完成提交

或者也可以git commit –m “注释内容”, 直接带注释提交

3.查看文件提交记录

执行 git log 文件名 进行查看历史记录

git log --pretty=oneline 文件名 简易信息查看

4.回退历史

git reset --hard HEAD^ 回退到上一次提交

git reset --hard HEAD~n 回退n次操作

5.版本穿越

进行查看历史记录的版本号，执行 git reflog 文件名

执行 git reset --hard 版本号

6.还原文件

git checkout -- 文件名

7.撤销某个文件暂存区的提交

git reset 文件名

8.去掉某个文件的版本追踪

git rm -- cached 文件名

9.比较文件

git diff 文件名

git diff HEAD 文件名

创建分支

git branch <分支名>

git branch –v 查看分支

切换分支

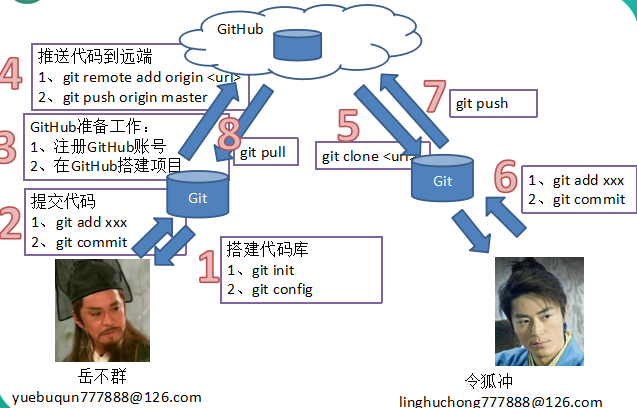
git checkout <分支名>

一步完成： git checkout –b <分支名>

合并分支

先切换到主干 git checkout master

git merge <分支名>



增加远程地址

git remote add <远端代号> <远端地址> 。

<远端代号> 是指远程链接的代号，一般直接用origin作代号，也可以自定义。

<远端地址> 默认远程链接的url

例： git remote add origin https://github.com/user111/Helloworld.git

推送到远程库

git push <远端代号> <本地分支名称>。

<远端代号> 是指远程链接的代号。

<分支名称> 是指要提交的分支名字，比如master。

例： git push origin master

从GitHub上克隆一个项目

git clone <远端地址> <新项目目录名>。

<远端地址> 是指远程链接的地址。

<项目目录名> 是指为克隆的项目在本地新建的目录名称，可以不填，默认是GitHub的项目名。

命令执行完后，会自动为这个远端地址建一个名为origin的代号。

例 git clone https://github.com/user111/Helloworld.git hello\_world

从GitHub更新项目

git pull <远端代号> <远端分支名>。

<远端代号> 是指远程链接的代号。

<远端分支名>是指远端的分支名称，如master。

例 git pull origin master