git config --global user.name 'susu'

git config --global user.email 'lk19931990@163.com'

cd d:/mywork

git init

cd.>susu.txt

git status

git add susu.txt

git commit -m 'add susu.txt'

cat susu.txt

--查看文件的不同之处

git diff susu.txt

git log --pretty=oneline

git reset --hard HEAD^

git reflog

--恢复到6120

git reset --hard 6120

--删除文件

git rm ha.txt

git reset HEAD ha.txt

git checkout -- ha.txt

--生成本机的明文与加密密钥-C大写

ssh-keygen -t rsa -C "lk19931990@163.com"

--本地仓库远程github上建立的远程仓库关联

git remote add origin [git@github.com:FallInLoveOfOne/myhome.git](mailto:git@github.com:FallInLoveOfOne/myhome.git)

--如果上面步骤提示fatal: remote origin already exists.

$ git remote rm origin

--第一次-u将本地仓库同步到远程仓库，origin代表远程，master代表本地仓库的主分支

git push -u origin master

--文件先commit到本地然后push到远程仓库

git push origin master

--查看当前分支

git branch

--创建dev分支且切换到dev分支

git checkout -b dev

--切换到master分支

git checkout master

--删除分支dev2

git branch -d dev2

--如果每个分支都对同一个文件有修改且commit,在当前分支下执行合并，会显示冲突，手动修改冲突文件后再commit到本分支下

git merge dev2

--查看分支合并图

git log --graph

--push本地的dev分支到远程的dev

git push origin dev

--多人协作时，使用git clone同步到本地的默认是远程的master分支，例如经常需要使用dev分支进行开发，可以使用如下命令获取远程的dev分支

git checkout -b dev origin/dev

--如果存在同一个文件已经有人修改push过，则本人在push的时候会提示无法push，解决办法是先pull到本地，解决冲突之后再push

git branch --set-upstream-to=origin/dev dev

git pull

git push origin dev

--查看远程信息库

git remote -v

--删除远程git仓库关联，再可关联码云

git remote rm origin