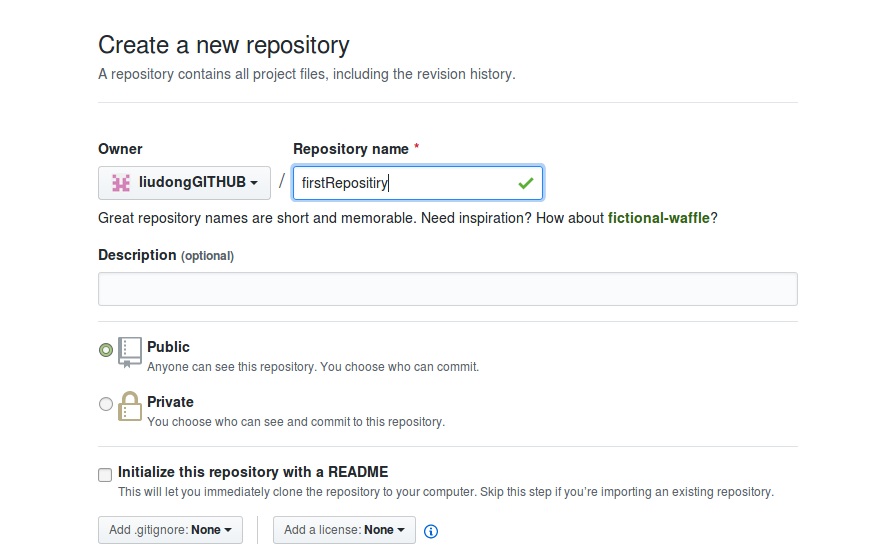
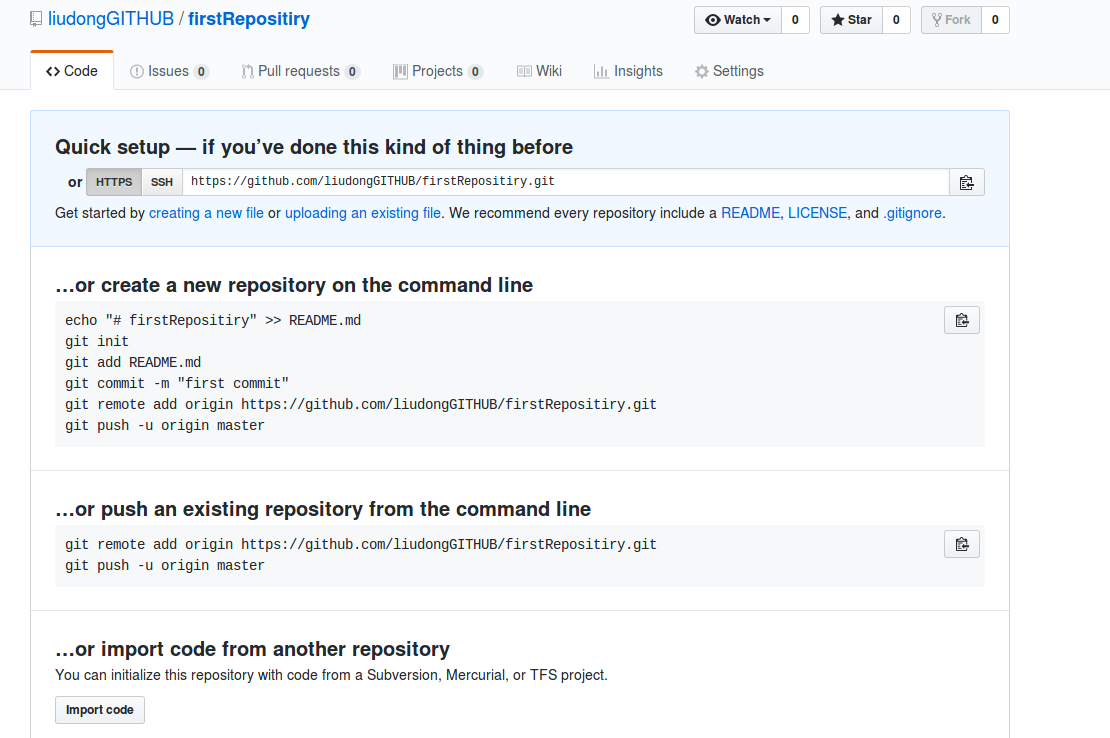
**git基本使用**

1.首先自己申请一个git仓库账号用于练习，现在比较流行的git仓库有github，gitlab，开源中国，CSDN代码托管平台， 建议使用github练习一下

github网址 ： https://github.com/

2.在仓库新建一个仓库（repository）如下图：



如上图就行已经创建好一个仓库：<https://github.com/liudongGITHUB/firstRepositiry.git> 这个是你git仓库的地址

下面是git仓库的一些基本命令，后面会说

echo "# firstRepositiry" >> README.md

git init

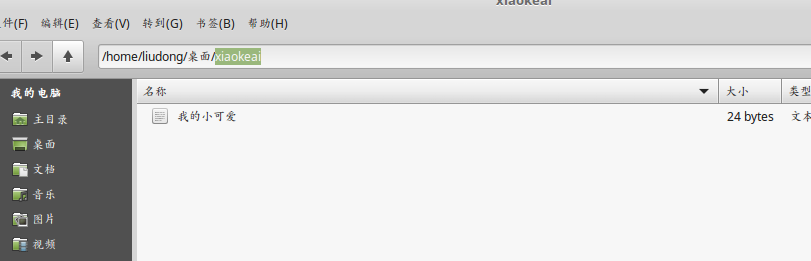
git add README.md

git commit -m "first commit"

git remote add origin https://github.com/liudongGITHUB/firstRepositiry.git

git push -u origin master

3.本地新建一个文件夹作为本地代码的存放地址，然后和网络仓库相关联



xiaokeai 就是新建的文件夹，我在文件夹下面写了一个文件：我的小可爱，待会要把xiaokeai 这个文件夹和网络仓库关联起来

**打开命令窗口-->定位到xiaokeai 文件夹下**

**git init** :初始化本地文件夹xiaokeai 为git版本控制的文件夹，初始化成功xiaokeai 下会生成.git文件夹，里面放的是git相关的文件

**git remote add origin https://github.com/liudongGITHUB/firstRepositiry.git**：将xiaokeai 文件夹和网络仓库绑定，

查看是否绑定成功：**git remote -v**

出现origin https://github.com/liudongGITHUB/firstRepositiry.git (fetch)则表示绑定成功



将本地的文件提交到网络仓库中：

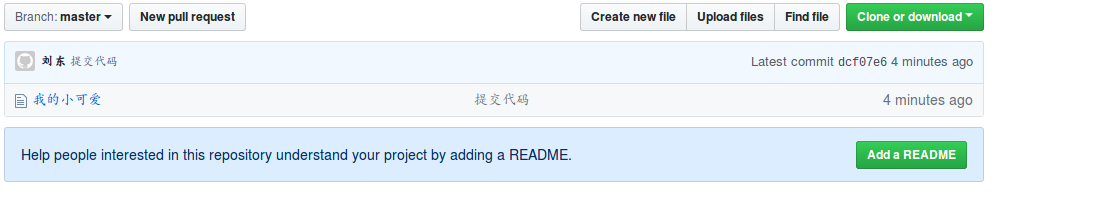
**git add .** 选中xiaokeai文件夹中所有未提交的文件，如果想选中其中一个文件可以这样 git add 我的小可爱

**git commit -m”提交代码”**：将选中的代码提交到本地仓库中，-m后面是提交说明

**git push origin master:master** : 本地在初始化后有一个默认分支master，网络仓库还没有，此命令是将本地仓库中文件同步上传到网络仓库，并且在网络仓库新建一个master分支（有可能提交的时候需要输入github的账号和密码，以后可以设置自动保存不用每次提交都输入）



现在可以看下网络仓库是否已经有了我们上传的文件：



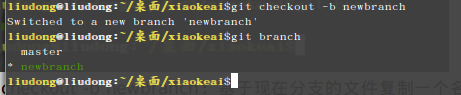
已经存在了，说明本地文件和网络仓库同步成功

1. 现在本地文件已经和网络仓库关联起来了，可以练习基本的git命令

git branch : 查看现在所在的分支

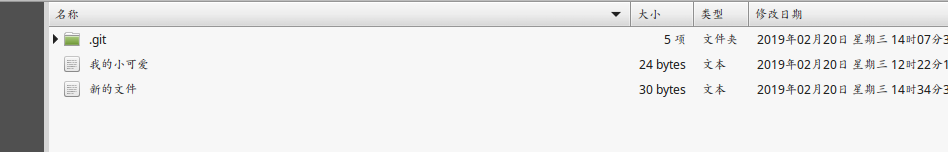


git checkout -b newbranch：基于现在分支的文件复制一个名叫newbranch的分支，并且切换到newbranch这个分支：如下可以看到已经切换到新分支上了

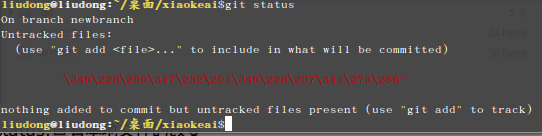


我们在本地新增一个文件并上传到网络仓库：

（新增了一个叫 新的文件）文件内容随便写几个字

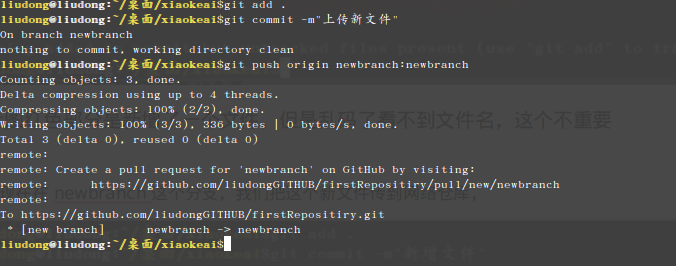


git status:查看本地文件的改变

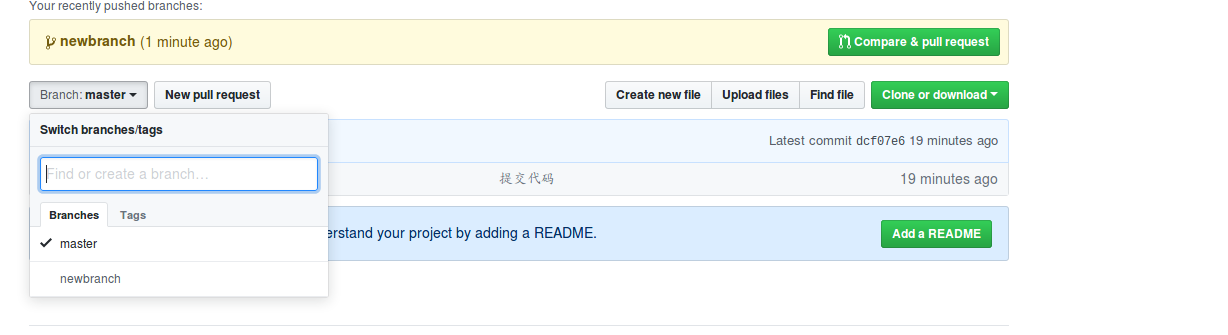


如上图红色部分是新增了一个文件，但是乱码了看不到文件名，这个不重要

我们现在在newbranch这个分支，我们把这个新文件传到网络仓库，



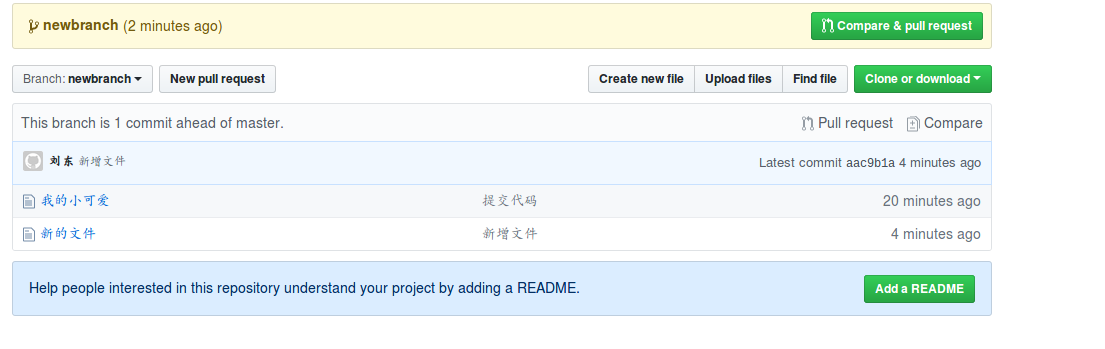
现在网络仓库应该有了新的分支，来看一下



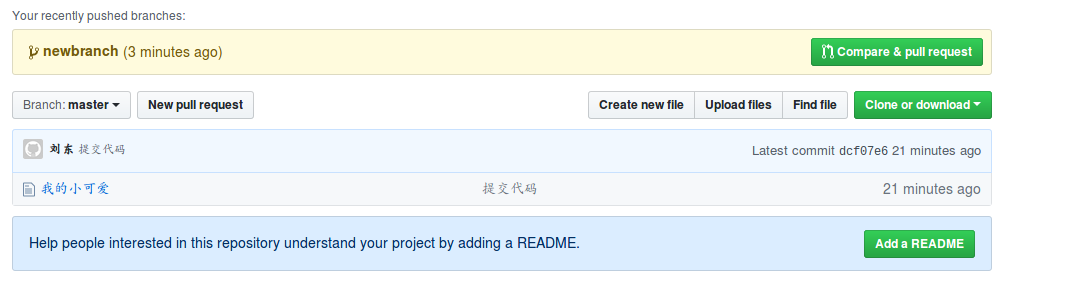
有个叫newbranch的分支，

来看一下两个分支的区别：

Newbranch分支：能看到有两个文件

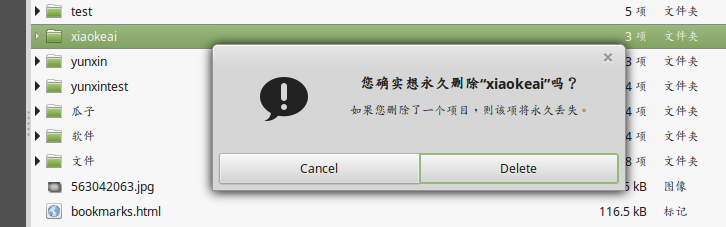


Master：

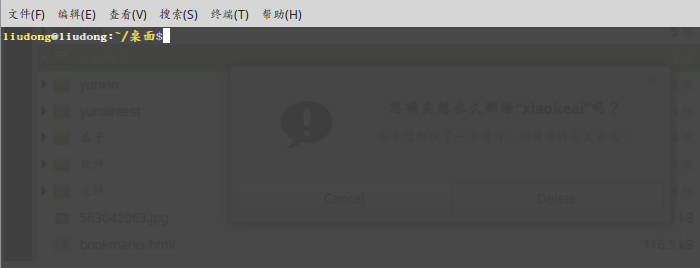


还是之前那一个文件

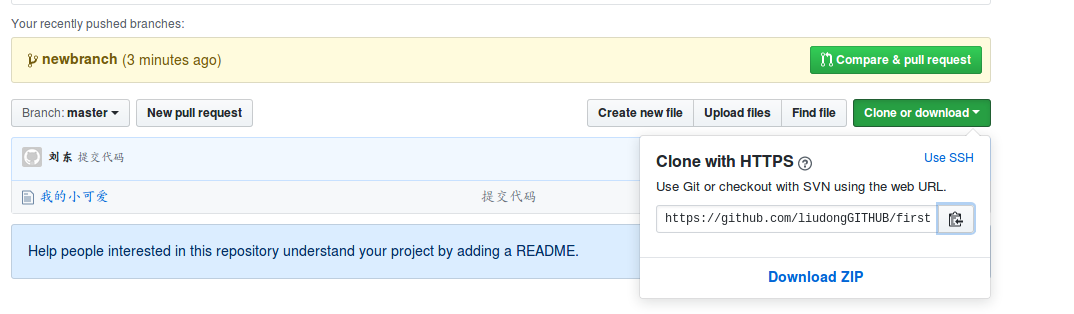
现在把本地的文件仓库都删掉，重新从网络仓库拉下来：



选好要保存文件的位置打开命令窗口（我存在桌面上）



开始拉代码：

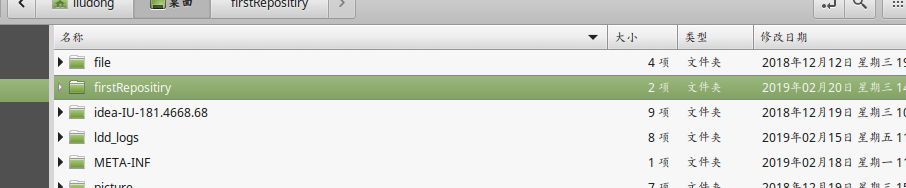


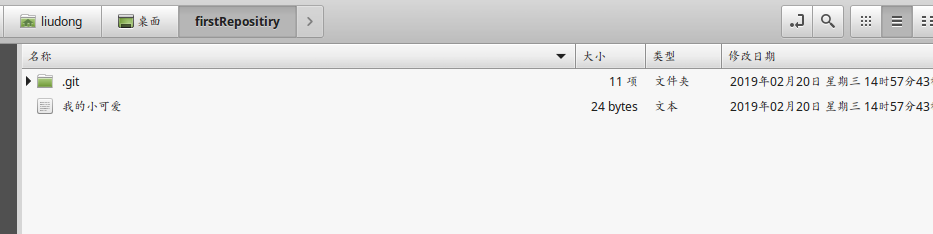
右边这个地址是网络仓库代码地址哦，复制一下

git clone https://github.com/liudongGITHUB/firstRepositiry.git



这就已经从网络仓库下载下来了，去桌面看下有没有

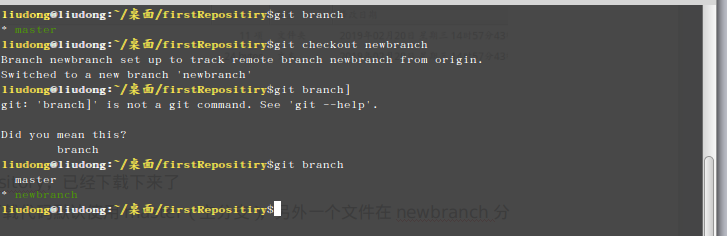




我们网络仓库起的名字是firstRepository，已经下载下来了

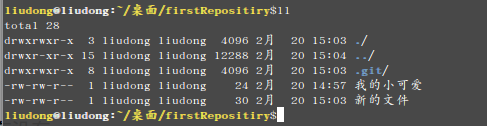
但是里面只有一个文件：这是因为下载代码默认使用master（主分支），另外一个文件在newbranch分支

我们切换到分支newbranch：git checkout newbranch



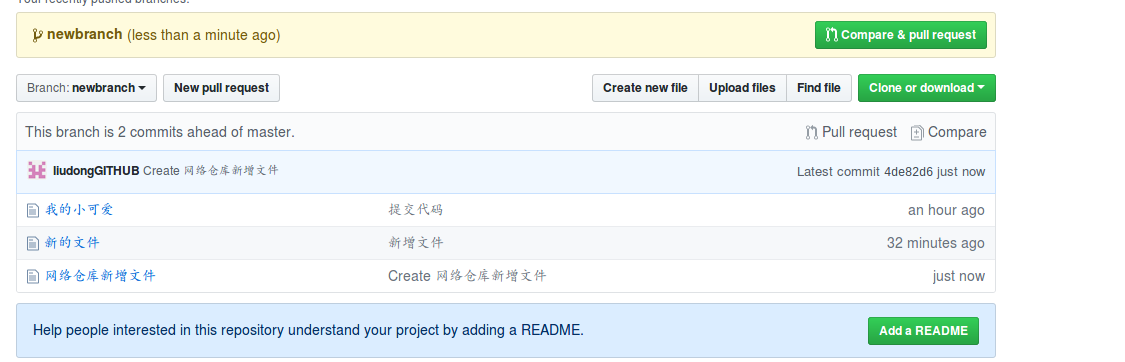
已经在newbranch分支了

现在看看本地文件另一个文件有没有：如下图已经是两个文件了

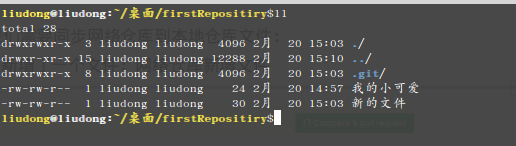


如果别人更新了网络仓库的文件，我们需要同步网络仓库到本地仓库文件：

1. 在网络仓库中新增一个文件：刚刚新增了一个文件：网络仓库新增文件

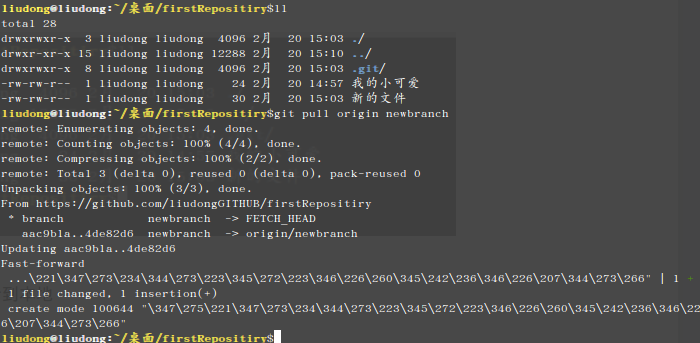


1. 看下本地是否有这个文件

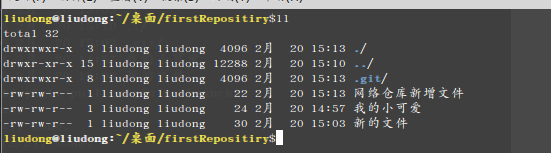


发现本地还是只有两个文件，

3.那么我们把网络仓库的新文件同步到本地

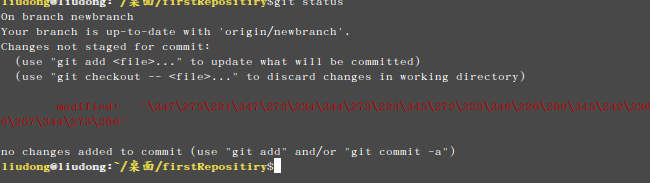


1. 在查看本地文件：已经多了一个文件



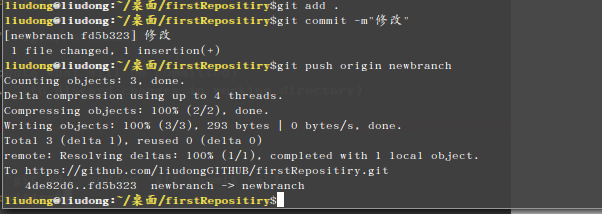
现在把本地newbranch分支的文件修改一下，同步到网络仓库中去

我自己手动改了一个文件，现在看看本地git状态:git status



显示修改了一个文件

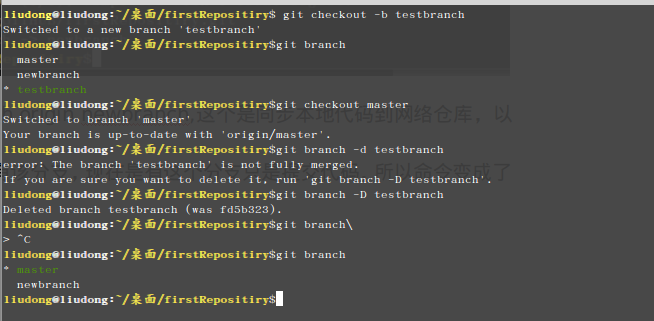
把这个文件提交到网络仓库中去



有一个地方需要注意：git push origin newbranch,这个是同步本地代码到网络仓库，以前同步的时候都是网络仓库没有该分支，现在是有这个分支只是提交代码，所以命令变成了这样

删除本地分支：

首先本地新建一个分支再删掉



解释下上述命令

1. 新建一个叫testbranch的分支 git checkout -b testbranch
2. 查看当前所在分支，发现已经在新建的testbranch分支了 git branch
3. 切换所在分支到master分支 git checkout master
4. 删除testbranch分支 git branch -d testbranch

提示有错：因为testbranch分支的内容没有合并到主分支master分支，所以不让删除testbranch分支

5.强制删除testbranch分支(d 和 D的区别)：git branch -D testbranch

6.查看当前所在分支 （发现已经在master分支并且新建的分支已经删除了）git branch：

常用命令：

**分支的创建与切换**:

**创建本地分支:git branch newBranch**

**切换本地分支:git checkout newBranch**

**创建并切换本地分支:git checkout -b newBranch**

**拉取远程分支并作为要创建的本地分支:git checkout remoteBranch**

**相当于(git checkout -b remoteBranch && git pull origin remoteBranch && git branch --set-upstream-to=origin/remoteBranch)**

**删除本地分支:git branch -d(delete) branchname**

**根据其他远程分支创建新分支**:  **git branch branchName origin/otherBranchName**

**根据其他远程分支创建新分支并且切换到新创建的分支**:  **git branch –b branchName origin/otherBranchName**

**把远程分支newBranch拉到本地当前分支:git pull origin newBranch**

**把远程分支newBranch拉到本地localbranch分支:git pull origin newBranch:localbranch**

**拉取远程分支并创建本地分支:git checkout -b 本地分支名x origin/远程分支名x**

**提交本地分支作为远程分支**:**git push origin test:test**

**关联本地分支和远程分支:git branch --set-upstream devtest origin/devtest**

**git branch --set-upstream-to origin/remoteBranch localBranch**

**git branch --set-upstream-to=origin/remoteBranch**

**重命名分支: git branch –m oldBranchName newBranchName**

**强制重命名分支: git branch –M oldBranchName newBranchName**

**提交本地分支作为master分支: git push origin branchName:master**

**撤销最近一次提交:**  **git reset HEAD^**

**文件回退: git reset --hard 和 git reset --sort区别**

**git reset --soft**：回退到某个版本，只回退了commit的信息，不会恢复到index file 一级。如果还要提交，直接commit即可；

**git reset --hard**：彻底回退到某个版本，本地的源码也会变为上一个版本的内容， 撤销的commit中所包含的更改被冲掉；

**git reset --hard** **版本号**(git log) 回退到某一个版本

**合并分支: 将newBranch 分支合并到当前分支:git merge newBranch**

**查看分支信息:**

**查看本地分支和远程分支的对应关系:git branch -vv**

**可以看见每一个分支的最后一次提交:git branch -v**

**查看所有分支最后一次提交:**  **git branch –v**

**查看项目的远程克隆地址 git remote -v**

**列出本地所有分支,当前分支会被星号标示出:git branch**

**git branch --all**

**将远程分支信息获取到本地**: **git fetch**

**拉取远程仓库分支目录:git pull --all**

**将本地分支目录同步到远程:git push --all**

**没有合并的分支:git branch --no-merged**

**查看哪些分支合并到当前分支来: git branch --merged**

**查看当前分支是从哪个分支切出来的:git log --oneline --graph --decorate**

**删除分支**

**远程删除git服务器上的分支:git push origin --delete branchName**

**删除本地的远程分支：git branch -r -D origin/BranchName (可以使用这种语法，推送 一个空分支到远程分支，其实就相当于删除远程分支**：$ **git push origin :branchName** )

**git添加远程仓库地址:git remote add origin https://gitlab.com/liudongGITHUB/idea.git**

**git取消远程仓库地址关联:git remote remove origin**

**git push -u origin master** 上面命令将本地的master分支推送到origin主机，同时指定origin为默认主机，后面就可以不加任何参数使用git push了。