**一、查看远程分支**

使用如下git命令查看所有远程分支：

git branch -r

查看远程和本地所有分支：   
git branch -a

查看本地分支：   
git branch   
在输出结果中，前面带\* 的是当前分支, 是本地HEAD所指向的分支。

***二、拉取远程库上创建的分支并创建本地分支***

**方法一**

使用如下命令：

git fetch

git checkout -b 本地分支名x origin/远程分支名x

使用该方式会在本地新建分支x，并自动切换到该本地分支x。

采用此种方法建立的本地分支会和远程分支建立映射关系。

P.s 如果出现Cannot update paths and switch to branch 'xxxxxxxx' at the same time

这是因为当我们在远端建立分支以后。在本地库上还没有origin上还没有fetch到分支的信息，所以直接checkout时，本地库并不知道远端库有新建的这个分支。

**方式二**

使用如下命令：

git fetch origin 远程分支名x: 本地分支名x

使用该方式会在本地新建分支x，但是不会自动切换到该本地分支x，需要手动checkout。

采用此种方法建立的本地分支不会和远程分支建立映射关系。

***二、创建本地分支进而创建远程代码库分支***

在本地库首先:

git checkout -b <新分支名>

git push origin <新分支名> 这将会在远程库上创建该分支

***三、在远程库上删除分支，再删除本地分支***

我们首先在远程库上删除了分支，然后在本地的上删除分支。

git branch -D <分知名> # 删除分支

git remote show origin # 显示所有远端分支情况，进而清除stale状态的关联分支

git remote prune origin # 删除清除stale状态的关联分支

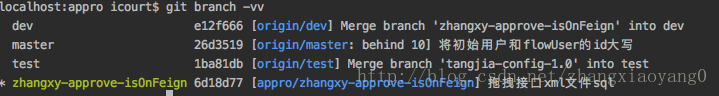
***四、在本地删除分支，再同步到远端删除该分支***

git push origin --delete <BranchName> 删除远端分支以及本地关联分支, 但是本地该分支还存在

git branch delete -D <BranchName> 删除本地该分支

**本地分支和远程分支建立映射关系的作用**

**建立本地分支与远程分支的映射关系（或者为跟踪关系track）。**

这样使用git pull或者git push时就不必每次都要指定从远程的哪个分支拉取合并和推送到远程的哪个分支了。   
git branch -vv   
输出：   


上面的本地分支和远程分支都有映射关系，如果没有，就需要手动建立，建立当前所在本份之与远程分支的联系：   
git branch -u origin/分支名，   
或者   
git branch --set-upstream-to origin/分支名   
origin 为git地址的标志，可以建立当前所在分支与远程分支的映射关系。

**撤销本地分支与远程分支的映射关系**

git branch --unset-upstream   
之后可以再次用git branch -vv 查看本地分支和远程分支映射关系

**P.s: 本地分支可以与远程不同名的分支建立映射关系** 