基本命令：

1.检出到新分支，新分支将会保留当前分支的推送历史  
git checkout -b new\_branch\_name

2.开发完毕后推送到远程仓库  
1）git add your\_file\_modified 或者 git add .

2）git commit -m 'your\_commit\_message'

3）git push origin origin\_branch\_name

3.拉取远程仓库的分支到本地

git checkout -b origin\_branch\_name origin/ origin\_branch\_name

4.更新远程仓库的提交和分支等信息

git remote update

5.删除远程仓库的指定分支

git push origin -d del\_branch\_name

实际应用：

1.做hotfix步骤，（开发新功能同样）

1）检出到需要做hotfix的分支 如：git checkout master

2）新建hotfix分支，并检出到这个新分支

git checkout -b hotfix\_branch

3）进行hotfix开发工作

4）提交hotfix开发工作

git add .

git commit -m '这次hotfix所做的工作'

(3和4可能会反复进行，产生多次提交记录)

5）记录下所提交hotfix的commitid，可通过git log查看，如果有多个，顺序按照提交顺序排列

6）切换回主分支（需要合并hotfix的分支）

git checkout master

7）进行cherry-pick

git cherry-pick commitidA commitidB commitC commitD ……

可能的情况如下：

a）一切正常无冲突，直接提交到远程版本库的主分支

git push origin master

b）有冲突，先用git status查看冲突的文件，解决冲突后

增加更改：git add .

继续cherry-pick：cherry-pick --continue

重复b步骤，直到完全没有问题为止，最后执行a操作

2.把多个提交合并成一个

假设有如下提交历史（commitid）

commitA

commitB

commitC

commitD

需要把ABC合并成一个提交

1）git rebase -i commitD

在弹出的vim界面上，输入:2,$s/pick/s/g (注意：包含开头的冒号)并保存

在弹出的编辑提交信息vim界面上，编辑此次合并后的commit message（可选）并保存

此时通过git log即可看到前三个提交已经合并成一个（此时可利用这个commitid进行cherry-pick等操作，也可以把这个分支复制成一个新分支并推送到远程仓库，这样其他成员就可以直接使用这个commitid进行其他操作，1:复制到一个新分支 git checkout -b new\_branch\_name，2:推送到远程仓库git push origin new\_branch\_name，3：其他成员通过git remote update之后即可使用这个commitid，也可以拉取这个分支到本地进行其他操作git checkout -b origin\_branch\_name origin/origin\_branch\_name）