（1）Gitrebase后可能会出现冲突，出现冲突以后，git会停止rebase，待解决冲突以后使用gitadd，这时就不用在gitcommit，需要执行git rebase –continue

这样git会继续应用（apply）余下的补丁

（2）也可以使用gitrebase –abort终止当前的合并

（3）gitapply和git pop的区别，gitapply只是把栈里面记录的内容应用到当前的版本，而栈里面的内容不会被删除，而gitpop在把栈里面的内容应用到当前版本的同时会把栈里面的内用删除。

（4）gitstashlist显示当前栈里面的版本号。Gitstash apply stash @{1}应用第几个版本。Gitstash clear 清楚栈里面的内容，gitstash备份当前的工作区的内容，从最近的提交中读取相关的内容，同时将工作区的内容保存在栈中，正确的做法就是在gitstash之后再执行gitstashapply之后当前的工作区就会恢复过来了。

（5）删除本地的分支，gitbranch –D {branch Name} 可删除本地分支banch name

（6）切换分支，git checkout – b {本地branch Name} {remote Name}

查看本地分支git branch 切换本地分支还可以gitcheck {branch Name}

（7）git撤销commit ：先通过使用gitlog查看commit日志，找到需要回退的那次commit 的哈希值commit\_ID,然后使用gitreset –hard commit\_ID即可进行回退。

1. ls–ah 可以查看默认隐藏的目录和文件
2. 把自己的本地库和github上的库相关联使用以下命令：

git remote add origin [git@github.com:[gitnamegit](mailto:git@github.com:[gitnamegit)的用户名]/[xxx.git]

第一次提交的时候使用命令:git push –u origin master

1. git 添加用户名和密码：

git config --global user.email 邮箱

git config –global user.name 用户名

上面是配置全局的用户名和密码,即电脑中所有git项目的用户名和密码，如果配置该git 项目的用户名和密码把global去掉。

（4）git reset –mix id 的方法是将HEAD变了（也就是说提交记录变了），但是文件并没有改变，而git reset –soft id 实际上是git reset –mixed 后又做了一次git add 操作而git reset –hard id是将git HEAD变了同时文件也发生了改变，因此可以利用git reset –-soft id 来就将id 以后的提交合并成一个提交。