Git命令：

git log:输出三次版本信息

git log –pretty=oneline:简要输出三次版本信息

git log --graph：看分支合并图

git reset --hard HEAD^：回退到上次版本

注：HEAD指针指向当前版本

git reset --hard commit号：回退/前进到某个版本

git reflog：查看git的每一次命令（关机后再开机也可以记录）

git diff HEAD --空格文件名：查看工作区和暂存区中这个文件的区别

git checkout --空格文件名：让这个文件回到最近一次git commit或git add时的状态，即**丢弃工作区的修改（包括之前提交后，工作区删除了文件，现在想恢复）**

git reset HEAD 文件名：**丢弃暂存区的修改，即使上一次add无效**

**注：**先用git reset HEAD 文件名，再用git checkout --空格文件名 可以使整个工作区和暂存区都恢复干净，即未做过任何改动。

删除：

工作区已经删除了一个文件后，而这个文件之前有提交过，现在如果确定不想要，就用git rm 文件名 ，然后再提交（如果只是add过，则用git reset HEAD 文件名）

远程库：

要关联一个远程库，使用命令

git remote add origin [git@server-name:path/repo-name.git](mailto:git@server-name:path/repo-name.git)

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改。

git push：

git push <远程主机名> <本地分支名>:<远程分支名>

git push origin master

将本地的master分支推送到origin主机的master分支。如果master不存在，则会被新建。

1.如果当前分支与远程分支之间存在追踪关系，则本地分支和远程分支都可以省略，即

git push origin（将当前分支推送到origin主机的对应分支）

1.1如果当前分支只有一个追踪分支，那么主机名都可以省略，即

git push（不带任何参数的git push，默认只推送当前分支）

2.不管是否存在对应的远程分支，将本地的所有分支都推送到远程主机，即

git push --all origin

2.1如果远程主机的版本比本地版本更新，推送时Git会报错，要求先在本地做git pull合并差异，然后再推送到远程主机。这时，如果你一定要推送，可以使用

git push --force origin

注：适合在本地新建分支，然后推送到远程，使远程具有该分支的情况

分支：

创建：git branch 分支名

切换：git checkout 分支名

创建+切换：git checkout -b 分支名

合并某分支到当前分支：git merge 分支名

删除分支：git branch -d 分支名

强制删除：-D

git merge --no-ff -m "merge with no-ff" 分支名：此次禁用Fast forward模式，合并后可以看到合并之前的历史分支。

当手头工作（现在这个分支1）没有完成时，先把工作现场git stash一下，然后去修复bug（一般是切换到另一个分支2上，然后新建一个分支3，在这个分支3里进行bug修复），修复后（应该合并到原先分支2），再切换到分支1，然后git stash pop回到工作现场（可以用git stash list查看有无其他的现场工作保存）。

多人合作模式：

1. 首先，可以试图用git push origin 本地分支名 推送自己的修改；
2. 如果推送失败，则因为远程分支比你的本地更新，需要先用git pull试图合并；
3. 如果合并有冲突，则解决冲突，并在本地提交；如果没有冲突（则会pull之后自动提交）；
4. 再用git push origin 本地分支名 推送就能成功！

注：如果git pull提示“no tracking information”，则说明本地分支和远程分支的链接关系没有创建，用命令git branch --set-upstream 本地分支名 origin/远程分支名

Vim命令：

G：跳转到最后一行

o(小)：在当前行下插入一行

w（小)：保存

q（小）：退出

dd（小）：删除当前行