1. git init ：初始化git；
2. git remote add origin ‘git@......’：创建远程仓库
3. git pull origin master(分支名):拉去分支
4. git push origin master （分支名） 推送分支
5. git add Readme.txt 添加文件 –f 强制添加文件
6. git commit Readme.txt（文件名） –m “提交信息” ：提交文件；然后再推送。因为push和pull相似；pull再commit （个人习惯）。
7. git status 查看状态；文件的修改或者新增；删除等。
8. git diff Readme.txt(文件名)；查看具体的不同(文件名为空查看所有的变动的)
9. git rm test2.txt (文件名):删除仓库的文件；再commit；push；（命令git rm用于删除一个文件。如果一个文件已经被提交到版本库，那么你永远不用担心误删，但是要小心，你只能恢复文件到最新版本，你会丢失最近一次提交后你修改的内容。）
10. git log --pretty=oneline 参数 查看提交历史 查看版本号， 版本id等。
11. git reset –-hard HEAD^返回最近上一个版本；如果你又想返回最新的版本；可以reset –hard bb980f4488f6c1ff6f6764a2b5b77c70bad971f2 最新的版本号
12. git reflog用来记录你的每一次命令：可以查询到你返回的版本id
13. git checkout - -file 撤销修改，- -很重要，没有—是切换分支命令;还可以还原删除
14. git reset HEAD <file> 返回最新的版本： 可以暂存区修改撤销掉 unstage ，重新放回工作区
15. git checkout –b master（分支名） 表示创建并切换；== git branch dev ;git checkout dev;git branch –D 表示强行清除
16. git branch 查看当前分支
17. git merge dev（分支名） 合并指定分支到当前分支；
18. git branch –d dev （分支名） 删除指定的分支
19. git log – -graph 看到分支合并图 git log --graph --pretty=oneline --abbrev-commit
20. git merge dev –no-ff –m “”：普通模式合并 合并后有合并历史。
21. git stash 暂存起来
22. git stash list 查看暂存记录集合
23. git stash pop 回复并删除缓冲；== git stash apply;git stash dorp
24. git remote –v 查看远程仓库的信息
25. git checkout -b dev origin/dev：切换远程分支
26. git branch --set-upstream-to=origin/dev dev：制定远程分支和本地分支的链接地址
27. git rebase <branch> 把分叉的提交历史“整理”成一条直线，看上去更直观
28. git tag 查看标签，git tag <name> 打一个新标签；git tag id给某个id打标签；git tag –a v.01 –m 1095ad ;-a表示指定标签名，-m提交信息
29. git show <name> 查看某个标签详情
30. git tag –d v.01删除标签；
31. git push origin <tagname>将标签推送到远程仓库
32. git push origin :refs/tags/v0.9 这是删除后要推送到远程的操作
33. git config –global color.ui true :GIT显示颜色
34. git check-ignore -v App.class 检查忽略文件的规则
35. git rm -r --cached . 添加忽略列表不起作用
36. git add .需要实现35，36。37三个命令
37. git commit -m 'update .gitignore'
38. git config --global alias.st status 配置命令别名 --global参数是全局参数