1. 去git add <name> 添加文件到暂存区
2. git commit -m “” 添加文件到版本库
3. git init 初始化
4. cd:改变目录
5. pwd 显示所在目录路径
6. clear 清屏
7. Is 列出当前目录所有文件
8. rm 删除一个文件
9. touch 新建一个文件夹
10. mv 移动文件
11. # 表示注释
12. git status 查看当前状态
13. git diff 比较前后版本不同之处
14. git log 版本库状态（查看提交历史）
15. git reset --hard HEAD^ 版本改变
16. git reset --hard 1094a(commit id) 版本改变
17. git reflog 用来找commit id
18. cat <name> 查看文件内容
19. git checkout --readme.txt 丢弃工作区的修改
20. git reset HEAD readme.txt 把暂存区的修改回退到工作区
21. 删除文件 1.手动删除 2.git rm <name> 3.git remove -m “ “
22. git remote -v查看远程信息
23. git remove rm <name> 删除
24. git checkout -b dev 创建dev分支并切换到dev分枝
25. git branch 命令会列出所有分支，当前分支加\*号
26. git checkout master 切换到master分支
27. git merge dev 用于合并指定分支到当前分支
28. git branch -d dev 删除dev 分支
29. git switch -c dev 创建并切换到dev分支
30. git switch master 切换到已有的master分支
31. git log -graph 可以看到分支合并图
32. git merge --no-ff -m ” “ dev 与git merge 不同的是此方法可以看出分支信息
33. git stash 可以把当前工作现场储藏起来
34. git stash pop 恢复工作现场的同时把stash内容删了修bug用的
35. 在master分支修复bug，想要合并到当前dev分支，可以用git cherry - pick <commit>命令，把bug提交的修改复制到当前分支，避免重复劳动。
36. 如果要丢弃一个没有合并过的分支，可以通过git branch -D<name>强行删除
37. git push origin <name> 推送分支
38. rebase 操作可以将本地未push的分叉提交历史整理成直线
39. git tag <tagname> 用于新建一个标签，默认为HEAD 也可以指一个commit id
40. git tag 查看所有标签
41. git clone 克隆远程仓库
42. git pull 合并分支.
43. git remote add origin 传送至远端仓库
44. git push -u origin master
45. 删除远程仓库的文件夹



46 cd 进入目录

47 cd.. 返回上级目录