1. 提交代码命令
2. 初始化本地仓库

git init

1. 克隆远程仓库

git clone 远程仓库地址

1. 查看当前分支的状态

git status

1. 查看修改内容

git diff <fileName>

1. 添加修改后的文件到暂存区

添加具体文件：git add <fileName>

添加当前目录所有的文件：git add .

添加任意的的文件：git add \*

1. 提交修改后的文件到版本库

git commit -m “注释”

1. 添加并提交到版本库

git commit -a -m “注释”

1. 本地仓库代码提交到远程仓库

git push origin <branchname>

1. 查看历史记录

git log （参数：--pretty=oneline）

1. 查看历史命令记录

git reflog

1. 还原代码命令

1.修改了工作区或删除了某个文件，想还原时：

git checkout -- <fileName>

2.修改的文件添加加到暂存区时（即执行了add操作），想放弃修改时：

需要两步：第一步：git reset HEAD <fileName>

第二步：git checkout -- <fileName>

3.修改的文件已经提交了修改（即执行了commit操作），想要撤销本次提交，前提是还没有推送到远程库：

git reset --hard HEAD^

或 git reset –-hard HEAD~1

4.还原文件到某个版本:

git reset --hard <版本号> （git log 或 git reflog获取版本号）

1. 解决冲突
2. 情形：
   1. 本地仓库的旧版本和远程仓库新版本都改了同一个文件，在同步远程仓库的文件时会引发冲突
   2. 两个分支都改了某个文件，合并分支时也会引发冲突
3. 解决：
   1. 先观察命令行的提示，比如出现 (master|MERGING)
   2. 定位冲突文件
   3. 修改冲突文件
      1. <<<<<<<HEAD和=========之间是本地仓库的文件的内容，=========和>>>>>>>>52120ddaa555544daa之间是远程仓库的文件的内容
      2. 合并、修改、删除冲突内容
   4. 分别git add 、git commit、git push解决冲突后的文件
4. 分支策略
   1. 永久分支
      1. Master（主分支）
      2. Develop（开发分支）
   2. 临时分支
      1. Fixbug（修复bug分支）
      2. Feature（功能分支）
      3. Release（预发布分支）
5. 远程分支
   1. 远程分支第一次提交需要先执行命令：git push --set-upstream origin develop