# 创建版本库

## Git init 初始化仓库

通过git init命令把这个目录变成Git可以管理的仓库：

## Git Commit –m “…”

用命令git commit告诉Git，把文件提交到仓库

## Git status

git status命令可以让我们时刻掌握仓库当前的状态，上面的命令告诉我们，readme.txt被修改过了，但还没有准备提交的修改。

## Git diff

查看提交文件前的却别，以diff格式显示

### git diff HEAD – 文件名

可以查看工作区和版本库里面最新版本的区别

## Git log

显示当前仓库中的提交记录

### 1git log –pretty=oneline格式化为一行显示

## Git reset –hard HEAD

用来回退版本，HEAD 表示当前版本，回头上一个版本用HEAD^ ,若返回前N个版本，用HEAD~N

## Git reflog

git reflog用来记录你的每一次命令

## git checkout – 文件名

注意：文件名前必须加空格

撤销在工作区的修改**或删除**

## git push -u origin master

第一次推送master分支

## Git checkout –b 分支名

表示创建并切换分支，相当于：

**git branch** 分支名

**git checkout** 分支名

## git merge 分支名

合并指定分支到当前分支

## Git branch –d 分支名

删除指定分支

## Git merge –no-off –m “描述” 分支名

表示禁用fast-forward,git会在merge时创建一个新的commit，使得从分支历史上可以看出分支的信息

## Git stash

保留当前工作现场，如出现bug，需返回master分支修改后提交，此时需保存当前的工作现场（修改等）

## Git stash apply stash名和git stash drop stash名

Git stash apply恢复到保留的工作现场

Git stash drop stash 删除保留的工作现场

## Git stash pop

恢复的同时删除工作内容

Git stash list

查看已保存的工作内容

## Git brach –D 分支名

强行删除一个开发完成但还没有合并的分支

## Git brach –set-upstream 本地分支名 origin/远程分支名

建立本地分支与远程分支的链接，以便推送