1. git创建一个版本库，pwd显示所在位置

$ mkdir learngit

$ cd learngit

$ pwd

/Users/michael/learngit

$ git init

#变成git可管理的库

1. 添加文件

git add <file> #记得文件要放在库里面

1. 提交文件

git commit -m <message> #一次可以提交多个

1. 版本回顾

查看你的修改： Git log #记得每次修改完要add commit才会有记录！

想看起来短一点：git log --pretty=oneline

退回前一个版本：git reset --hard HEAD^ #文件会回到前一次储存的样子。

HEAD:当前版本，HEAD^:上个版本，HEAD^^:上上个版本

看一下TXT现在的样子： cat readme.txt

如果你回到过去之后还想回到未来，窗口未关闭可以去前面找版号用前四五位就行

git reset --hard 1094a

查看你的每一次命令：git reflog

1. 工作区和暂存区

工作区：电脑里能看到的目录：例 learngit文件夹

版本库：Git的版本库里存了很多东西，其中最重要的就是称为stage（或者叫index）的暂存区，还有Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。

（工作区add到暂存区，然后从暂存区commit到分支master）

查看目前文件的状态： git status

可以查看他们的添加和上传的情况

1. 管理修改

如果这么改：第一次修改 -> git add -> 第二次修改 -> git commit

那么第二次修改不会被添加！

git diff HEAD -- readme.txt

用于查看工作区和版本库里最新版本的区别

1. 撤销修改

git checkout -- file

丢弃工作区的修改

命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

1.readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

2.readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

总之，就是让这个文件回到最近一次git commit或git add时的状态。

注：一定要记得有--，不然就会变成切换分支的命令！

1. 删除文件

删除文件：$rm test.txt

当工作区和版本库不一致，git status就会记录到你删了文件。

然后，

如果你真的要删除：$git rm test.txt

$git commit –m “remove tset.txt”

如果你是误删的想要恢复：$ git checkout -- test.txt

1. 添加远程仓库

先在github上面建立库，然后根据提示用

$ git remote add origin git@github.com:\*\*\*\*\*/learngit.git

这样就可以建立一个叫做origin的远程库了

然后可以把本地库的内容推送push到远程

$ git push -u origin master

实际上是把当前分支推送到远程，-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

之后，只要本地做了提交，用

$ git push origin master

就可以把master的最新修改推送到github

如果要删除远程库

先用 $ git remote –v 查看一下当前远程库

然后根据名字删除 $git remote rm origin

1. 克隆仓库

我们也可以先建立远程仓库，然后克隆一个本地仓库

$ git clone git@github.com:\*\*\*\*\*/gitskills.git

然后就可以看到本地有库了

$ cd gitskills

$ ls #可以列出文件

注：$ cd ..\ 回到上级目录

1. 创建和合并分支

主分支master，HEAD指向当前分支