创建版本库

1.创建版本库的命令

$ mkdir new

$ cd new

$ pwd

/User/name/new

Pwd 命令是用于显示当前目录。在本电脑上，这个仓库的位置是/User/name/new

注意：如果是widow为了避免遇到问题，请确保目录名（包括父目录）不包含中文

2.通过git init 命令把这个目录变成Git可以管理的仓库：

$ git init

Initialized empty Git repository in /User/name/new/.git/

把文件添加到版本库

一定要放到new目录下，因为这是一个Git仓库，放到其他地方Git再厉害也找不到

1. 添加文件命令：

$ git add readme.txt

1. 提交文件到仓库命令;

$ git commit –m “wrote a readme file”(提交文件的描述)

1. 提交多个文件

$ git add file1.txt

$ git add file2.txt file3.txt

$ git commit –m “add 3 files”

Git的时光机

1.版本仓库的状态查看

$ git status

2.查看具体修改了什么内容

$ git diff

3.重新提交修改过的文件

$ git add readme.txt

版本的回退

1. 版本仓内容修改历史记录，查看

$ git log

Git log 命令显示从最近到最远的提交日志；如果嫌输出信息太多，可以试着加上 –pretty=oneline 参数

（git 的版本号是一串大串字符数据）

1. 回退上一个版本

:在Git中，用HEAD表示当前版本，也是最新版本如362888…..799979，上一个版本就是HEAD^,上上一个版本就是HEAD^^,当然往上100个版本写100个^比较容易数不过来所以写成HEAD~100.

$ git reset –hard HEAD^

1. 已经回退了上一个版本，然后再回退到未来原来版本

只要上面的命令窗口还没有被关掉，可以顺着往上找到最之前的版本号如：362888…;于是就可以指定回到未来的某个版本：

$ git reset –hard 362888

1. 要重返未来，就是在第二天后悔了，要退回原来的版本

$ git reflog

用来查看确定要回到未来的哪个版本。

工作区和暂存区

管理修改

1. 如是操作不会提交，第二次修改提交则不会查到

第一次修改----.git add -------第二次修改------git commit

这样子操作会在用 git status 查看的时候发现第二次提交没有提交上去的。

原因：Git 的管理的修改，当用git add 后，在工作区的第一次修改被放入暂存区，准备提交，但是，在工作区的第二次修改并没有放入暂存区，so git commit 只负责提交暂存区的数据，也就是第一次修改的数据，第二次修改的不会被提交

用 git diff HEAD – readme.txt 命令可以查看工作区和版本库里面最新版本的区别

如果要提交第二次修改成功的操作：

第一次修改------git add ---------第二次修改------git add ----git commit

撤销修改

1. 丢弃工作区的修改

$ git chechout – readme.txt

Git checkout – file 命令中的 – 很重要，没有 -- ，就变成切换到另一个分支的命令，我们在后面的分支管理中会再次遇到git checkout 命令

1. 撤销后悔的修改

修改了一些内容 --------- 执行了git add ------在没有执行 git commit 之前用git status 查看下，修改只是添加到了暂存区，还没提交------用 git reset HEAD file 可以把暂存区的修改撤销掉，重新放回工作区------然后再用“丢弃工作区的修改操作”

删除文件

1. 删除文件的操作

一般用 rm file 命令进行删除---------这时候git知道你删除了文件，因此工作区和版本库就不一致，用git status 命令查看删除了那些文件--------如确实要从版本库删除该文件，执行命令 git rm 删掉，并进行git commit

如：

$ git rm test.txt

….

$ git commit –m “remove test.txt”

如要是删错了：

执行：$ git checkout --- test.txt

git checkout 其实是用版本里的版本替换工作区的版本，无论工作区是修改还是删除，都是可以“一键还原”

远程仓库

使用GitHub官网获取一个免费的Git远程仓库

1. 再GitHub官网注册一个账号获得Git远程仓库-------创建SSH Key

$ ssh-keygen –t rsa –C [youremail@exanple.com](mailto:youremail@exanple.com)

1. 然后一直回车如一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id\_rsa 和id\_rsa.pub 两个文件，这两个文件就是SSH KEY的密钥对，id\_rsa是私钥，不能泄露，id\_rsa.pub是公钥可以给任何人
2. 登陆GitHub 打开“Account setting”，“SSH Keys”页面，点“Add SSH Key”，填上任意Title，在Key文本框里粘贴id\_rsa.pub；点“Add Key”，可以看到添加的key

添加远程仓库

本地已经有一个git仓库后，又想在GitHub创建一个git仓库，并且要两者进行远程同步

1. 登陆GitHub，然后在右上角找到“Create a new repo”，创建一新的仓库
2. 在Repository name 填入new，其他保持默认就好，点击“Create repository”按钮，就成功创建一个新的Git仓库
3. 目前这个仓库还是空，在本地的new仓库执行命令：

$ git remote add origin [git@github.com:name/new.git](mailto:git@github.com:name/new.git) (name 替换成自己的GitHub账号名)

1. 进行本地库内容推送到远程库

$ git push –u origin master