Git（分布式版本控制系统）

git init :将当前目录变成git可管理的仓库



工作区就是电脑上看到的目录。

工作区有一个隐藏目录 .git 是版本库。

版本库中的stage是暂存区。

master是创建版本库时git自动创建的唯一一个master分支，分支像可以合并的平行宇宙。



HEAD 指针 指向当前分支,当前分支指向最新的提交，就能确定当前分支及其当前分支的提交点。



**一般往版本库里添文件分两步**：

git add filename ：把要体检的所有修改放到暂存区

git commit -m “xx” ：提交更改，实际上是把暂存区的内容提交到当前分支，-m后面是此次提交修改的说明。

git status ：查看仓库目前的状态

git diff :查看尚未暂存的文件的修改

git log :查看提交历史

git reflog :查看所有的命令历史

**版本转换**

git reset --hard HEAD^ :回到上一个版本

git reset --hard xx ：回到制定版本 xx是commit id 通过git log或git relog可以找到

**撤销修改**

git checkout -- filename :丢弃工作区的修改

git reset HEAD filename :将暂存区的修改退回工作区

**删除文件**

rm filename ：在工作区删除文件

这时候git知道你删除了这个文件，现在有两个选择

1. git rm filename :把文件从版本库中删除 （还需要git commit ）

!!! 要将文件从***版本库***中删除需要两步：首先git rm 然后 git commit

1. git checkout -- filename :用版本中的版本替换工作区的版本（还原)

**分支**

git branch :查看分支

git branch name :创建分支

git checkout name :切换分支

git checkout -b name :创建并切换到该分支

git （branch） merge name :合并某分支到当前分支 || 没有branch

git branch -d name :删除分支 ||少写了一个n

**标签**

发布一个版本时，我们通常在版本库中打一个标签，标签也是版本库的一个快照。

git tag name ：给当前分支新建标签

git tag name commit\_id :给历史版本建标签,commit\_id是历史版本的提交id

git tag -a name -m “xx” ：可以制定标签信息

git tag :查看所有标签

git tag -d name :删除标签

多人协作

本地git仓库和Github仓库之间的传输通过SSH加密，在用户主目录下看有没有.ssh目录，并看看里面有没有id\_rsa和id\_ras.pub文件；如果没有，打开Shell（windows下用Git Bash），创建SSH KEY（输入ssh-keygen之后一直enter），id\_rsa.pub是公钥,添加到Github就可以进行推送了。

Cd到本机想要存放的目录

git clone [git@github.com:user/repository.git](mailto:git@github.com:user/repository.git)

就可以把git上repository仓库复制到本机了

一个教程的网址，讲的很详细

http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000