$ cd e:是跳转到E文件夹下，如果想进入E:Github文件夹，就需要输入$ cd e:/Github，之后如果想访问Github中的文件夹，直接“cd 文件夹名”即可。

“$ pwd”是查看当前目录

“$ cd ..”是退回到上一级目录下

“$ ls”是查看当前目录下都有哪些文件

“$ mkdir 文件夹名”是在当前目录创建新的文件夹

“$ rm -r 文件夹名”是删除当前目录下的文件夹

“$ touch 文件名.文件格式”在当前目录创建新的文件

“$ rm 文件名.文件格式”是删除当前目录下的文件

如何与新库建立连接:

” <https://lolitasian.blog.csdn.net/article/details/79085301>”

“$ git add -A”提交所有变化到缓存区

“$ git commit -m “修改注释””

给刚才add的东西加一个备注，上传到远程仓库之后，修改的文件后边会显示这个备注

“$ git push -u 仓库名称 分支”

仓库名称：刚才我添加连接的时候，给仓库起名叫origin

分支：你现在只有主分支，所以分支直接写master。以后合作项目的时候，成员之间建了不同的分支，你就可以往你自己的分支上推。

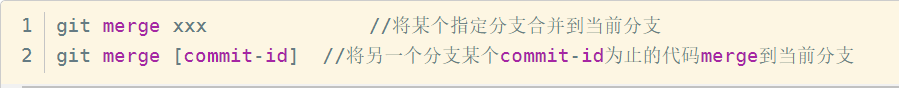
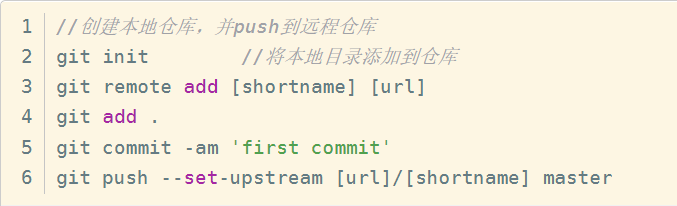
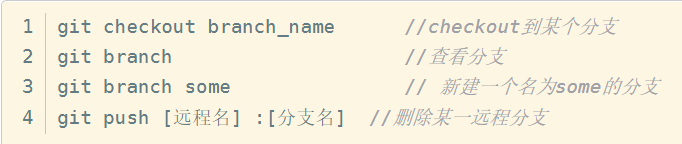
我们第一次推送master分支时，加上 –u参数才会把本地的master分支和远程的master分支关联起来，就是告诉远程仓库的master分支，我的本地仓库和是对着你的哦，不是对着别的分支的哦。

只有第一次推的时候需要加上-u，以后的推送只输入“$ git push 名称 分支”

$ git clone 加上你刚才的地址

仓库是你自己的，你就使用SSH连接，不是你自己的，你没权限你就切换到HTTPS，再复制地址。

它克隆下来是一个文件夹，你想把文件夹放哪里就在哪打开gitbash（前提是本地的仓库要和远程库已经连接上了，如果没有就会复制失败，还记得git remote -v吗？用它看一下你下下来的本地仓库连接上那个远程仓库没）



git stash //保存当前的工作进度。会分别对暂存区和工作区的状态进行保存

git stash save "message..." //这条命令实际上是第一条 git stash 命令的完整版

git stash list //显示进度列表。此命令显然暗示了git stash 可以多次保存工作进度，并用在恢复时候进行选择

git stash pop [--index] [<stash>] //如果不使用任何参数，会恢复最新保存的工作进度，并将恢复的工作进度从存储的工作进度列表中清除。如果提供参数（来自 git stash list 显示的列表），则从该 <stash> 中恢复。恢复完毕也将从进度列表中删除 <stash>。选项--index 除了恢复工作区的文件外，还尝试恢复暂存区。

git stash apply [--index] [<stash>] //除了不删除恢复的进度之外，其余和 git stash pop 命令一样

git stash clear //删除所有存储的进度

git stash show -p //查看最近stash的diff

git stash show -p stash@{1} //查看某个stash的diff



git update-index --assume-unchanged [filename] //假定某个文件没有发生变化，但是切换分支，pull代码会更新index

git update-index --no-assume-unchange [filename] //取消

git update-index --skip-worktree [filename] //将某个文件从git检测中忽略

git update-index --no-skip-worktree [filename] //取消s