git：版本控制软件
是一个分布式控制软件

本地仓库
每人一份版本库,都在本地

本地工作目录---add---暂存区--commit---版本库
从别人的仓库更新到自己 pull
反过来 ----push

命令：git add :将本地文件，增加到暂存区
git commit:将暂存区的内容提交到本地仓库/版本库/分支
git push: 将本地仓库的内容 推送到远程仓库/别人仓库（远程分支）
git （拉）pull:将远程仓库的内容 推送到 本地仓库（本地分支）

版本库/分支1/分支2.... 默认的主分支叫master

统一的官方托管网站github
去更新和提交密码需要知道你是谁，每次提交都写用户和密码，
所以要配置免密码登陆
先本地生成ssh key

先在本地配置然后发送给远程
配置ssh
现在本地配置，$ ssh-keygen -t rsa -C “mmdgqq@163.com”
-t 指定加密方式
发送给远程
github --settings
SSH and GPG keys title任意
key中输入刚刚在本地生成的ssh,刚刚在本地生成的id_rsa_pub文件内容粘贴到key框

测试连通性：

$ ssh -T git@github.com

如果本地和远程成功通信，

可以在./ssh目录中发现know_hosts文件

如果失败，多尝试几次，检查空格

接下来：　在本地建立git项目，并发送给远程

在项目根目录中，右键 git -bash

输入：git init

此时项目变成了一个git项目

在远程建立git项目

new　-建立项目　--生成唯一标识和https://github.com/mmdgqq/mygit.git

本地项目和远程项目关联

git remote add origin git@git@github.com:mmdgqq/mygit.git

第一次发布项目：

git add .　文件---暂存区　　一个版本

git commit -m "注释内容" //暂存区--本地分支（默认master）　版本库　　两个版本

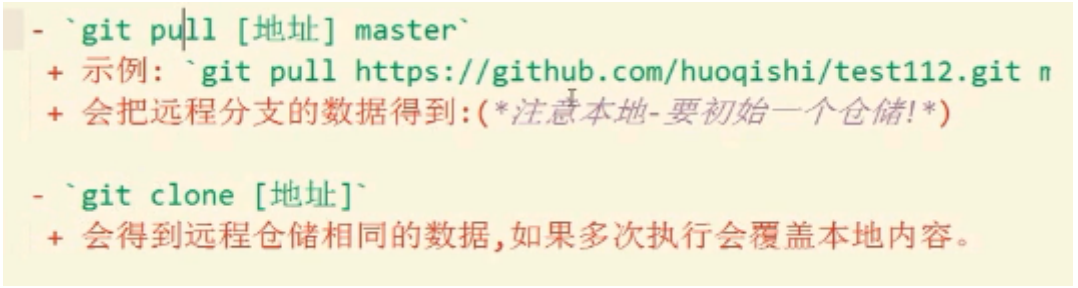
如果不写注释内容的话，会进入一种 vim编辑器的模式，

此时可以在vim编辑器中书写通过 i 进入编辑模式去写注释内容，但是一般不会那么写，

所以，如果进入了那种模式，要按esc退出　:q！强制退出

查看目前提交到暂存器还是仓库的状态 git status

查看日志：提交　git log --oneline



git pull 命令，会与本地做对比，然后把不一样的东西 增加到本地或者删除，去保证与远程仓库的一致性

而git clone会直接覆盖，没有对比的操作

分支的操作：

1：创建分支

git branch dev

2:查看分支：

git branch

3:切换到分支　使用关键字 checkout

git checkout dev

4:git status　//查看当前在那个分支进行操纵

5:git log --oneline　查看提交的记录

6：将分支与主分支的东西合并

　使用 merge关键字

 git merge 分支名字

　eg:git merge dev

7:删除分支

在别的分支下删除

git branch -d 分支名字

head指向最新修改的内容

整个框中代表分支合并时，发生冲突的地方 ，，，，，这时候需要手动更改合并，处理完需要再去提交一次

push和pull的简写方法

此写法针对一个仓库有效，并不是全局有效的

第一次写：git remote add origin 地址

第二次写时候：git push origin master 就不用写地址了

这里的origin 可以理解为一个变量，指向远程仓库的地址

第三次：写这个 git push origin -u master

第四次提交的时候，针对本项目，就可以直接写 git push 了

加上-u之后，git会把当前分支与远程的指定的分支进行关联

即是与远程分支关联的