**一、使用git向GitHub提交代码**

**1、安装git，**

* 官网下载（<https://www.git-scm.com/downloads>）;
* 安装过程（略）；
* git --version检查是否安装成功;

**2、配置git，**

* 打开Git Bash，使用git config命令；
* 最重要的是配置用户名和邮箱；

$ git config --global user.name "antozne"

$ git config --global user.email "ant@163.com"

(上面代码中的用户名和邮箱,自己设置,推荐和gitHub的用户名和邮箱相同)

**3、注册远程登录协议，**

* 使你的本地Git与远程GitHub仓库建立SSH远程登录协议

ssh-keygen -C 'your@email.address' -t rsa

（它会要求你输入密钥文件存储位置,直接敲回车默认即可

再然后要设置密钥口令(两次输入确认即可)。如果你不想在使用密钥时输入口令，将其留空即可。(口令及密码,要记住以后往远程仓库上传项目,和同步远程仓库的项目都需要用到这个密码)）

**4、将SHH公钥绑定到gitHub账户上**

* 找到window用户目录下找到一个.shh文件夹下的id\_rsa.pub文件用记事本打开,复制里面所有的内容,复制的内容；

登录GitHub----settings----SSH and GPG keys----New SSH Key；保存即可（GitHub自行注册）

**5、检查公钥是否正确绑定GitHub,**

* 在Git Bash下输入命令测试刚才的公钥是否认证正确。(输入“yes”；warning 不用理会。)

ssh -T git@github.com

**6、本地仓库和远程仓库(gitHub)的上传与下载**

**6.1、第一种方式:在本地建仓,并初始化,然后上传到远程仓库(GitHub)上的未初始化的仓库**

* GitHub上“New repository(新仓库)”，不勾选仍何其他东西（不创建README.md）；
* 点击 “Clone or download” 获得SSH地址，需要复制新仓库的shh地址，备用；
* 电脑本地任意一个盘,创建一个文件夹，来作为你的本地仓库(已有文件夹也可以)；
* 在该文件夹右键,点击Git bash Here；

$ git init

* 文件夹里多了一个.git文件夹（如果没有.git,可以勾选菜单栏中的“查看”----“显示隐藏文件夹”）；

**6.2、第二种方式,在gitHub上建仓库,并初始化,再将GitHub上的仓库克隆到本地仓库**

* 在GitHub上创建仓库，“New repository(新仓库)”，勾选仍何其他东西（创建README.md）；
* 点击 “Clone or download” 获得SSH地址，需要复制新仓库的ssh地址，备用；
* 在本地任意一个文件夹右键打开Git bash Here

$ git clone git@github.com/SiWanZi/Test.git（”git@github.com/SiWanZi/Test.git“为GitHub上的仓库的ssh地址）

* 本地文件夹中会clone一个和GitHub一样名称的文件夹（内带.git文件夹、README.md）

**7、将项目上传至gitHub远程仓库**

* 将你需要上传到GitHub的项目保存到本地仓库文件夹中；
* 添加到暂存区

$ git add . *(注意点前面有个空格，将工作区的所有文件添加到暂存区)*

* 暂存区的文件提交到本地仓库

$ git commit -m “第一次提交”*(“…”为提交备注，将暂存区的文件提交到本地仓库)*

* 本地仓库和远程仓库建立联系

$ git remote add origin git@github.com:Preci/myFirstRepository.git

（其中 origin 为给远程仓库起的别名，“git@github.com:Preci/myFirstRepository.git ”是你GitHub的ssh地址，让本地仓库和远程仓库建立联系，之后修改时可不用再输）

* 本地修改推送到 GitHub

$ git push origin master(将本地仓库中的内容推(push)到远程仓库中,这时可能要输入密码,密码就是生产ssh密钥时候设置的口令)；

刷新gitHub页面就可以看到上传的项目；

**8、将本地修改推送到 GitHub**

* git push 的一般形式是：

$ git push <远程主机名> <本地分支名>:<远程分支名>

* 几种常见的用法有：

git push origin master

（该形式中远程分支名被省略，表示将本地分支推送到与之存在追踪关系的远程分支（通常二者同名），如果该远程分支不存在，则会被创建。把本地的 master 分支内容推送到远程的 master 分支，还会把本地的 master 分支和远程的 master 关联起来。）

git push origin :refs/for/master

（该形式中本地分支名被省略，表示推送一个空的本地分支到远程分支，删除指定的远程分支）

git push origin

（如果当前分支与远程分支存在追踪关系，则本地分支和远程都可省略。）

git push

（如果当前分支只有一个远程分支，那么主机名都可省略。）

git push -u origin master

（第一次推送 master 分支时，加上 -u 参数，表示将本地的 master 分支推送到 origin 主机，同时指定 origin 为默认主机，之后就可以不加任何参数使用git push了，表达的意思和git push origin master相同。把本地的 master 分支内容推送到远程的 master 分支，还会把本地的 master 分支和远程的 master 关联起来。）

**9、常用命令**

1)、添加远程仓库

$ git remote add origin git@github.com:SiXiWanZi/Test.git

2)、删除远程仓库

$ git remote rm <repName>

3)、修改远程仓库

$ git remote rename <oldName> <newName>

4)、查看远程仓库列表

$ git remote -v