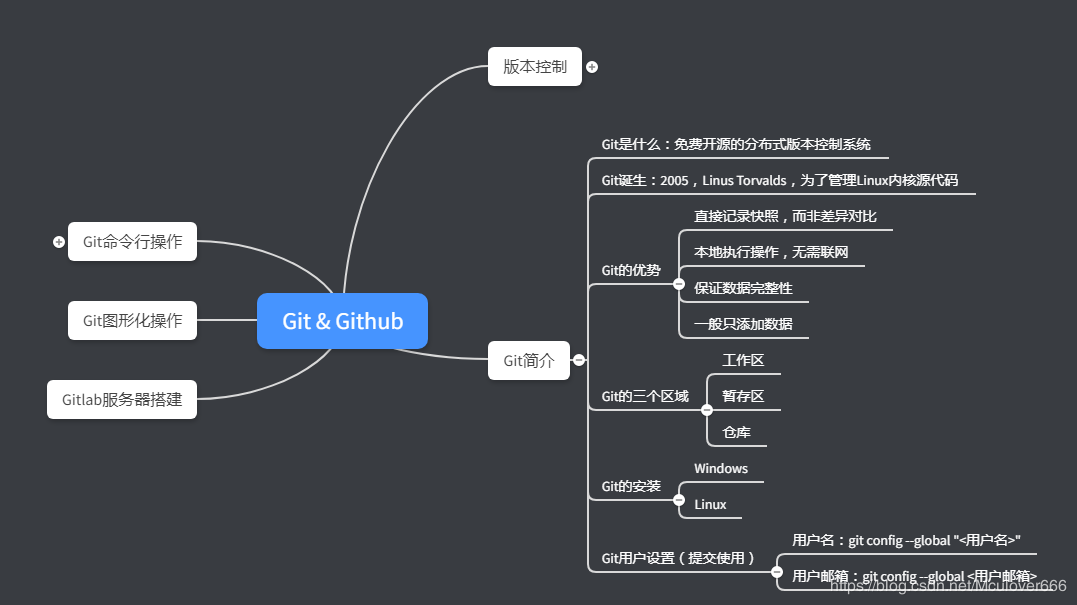
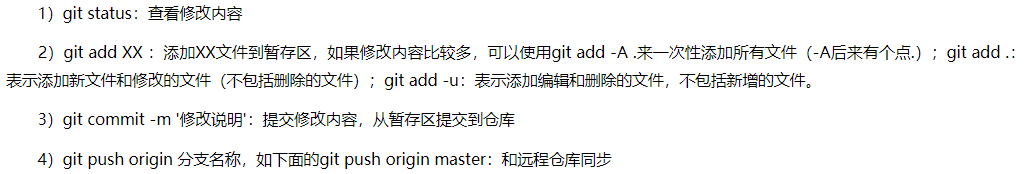
# 我只完成了2~3，后面4~10都没仔细看



## **git提交全部文件的基本步骤：**



## 2. 安装

官网<https://git-scm.com/>

下载地址<https://git-scm.com/downloads>

官网安装教程

<https://git-scm.com/book/zh/v2/%E8%B5%B7%E6%AD%A5-%E5%AE%89%E8%A3%85-Git>





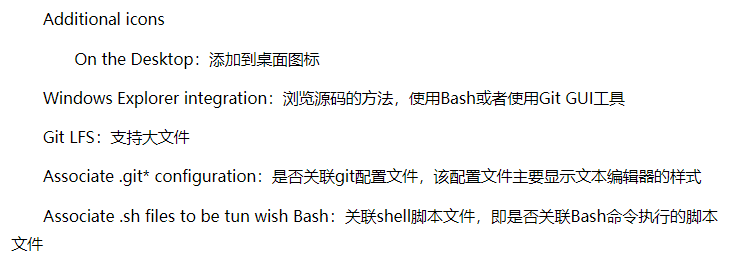
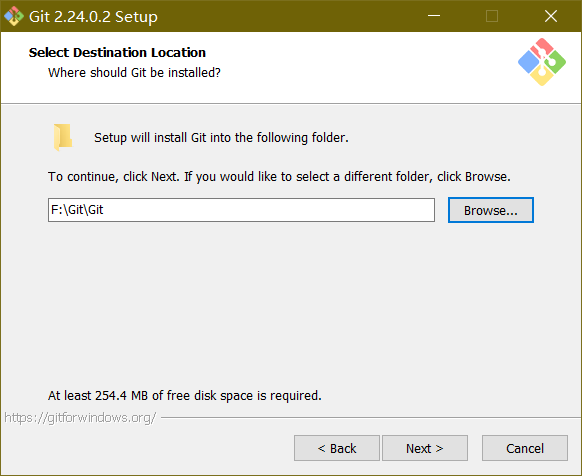
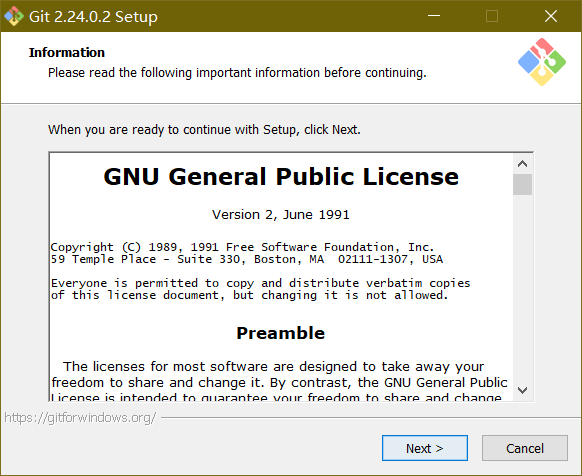
1. <https://git-scm.com/downloads>

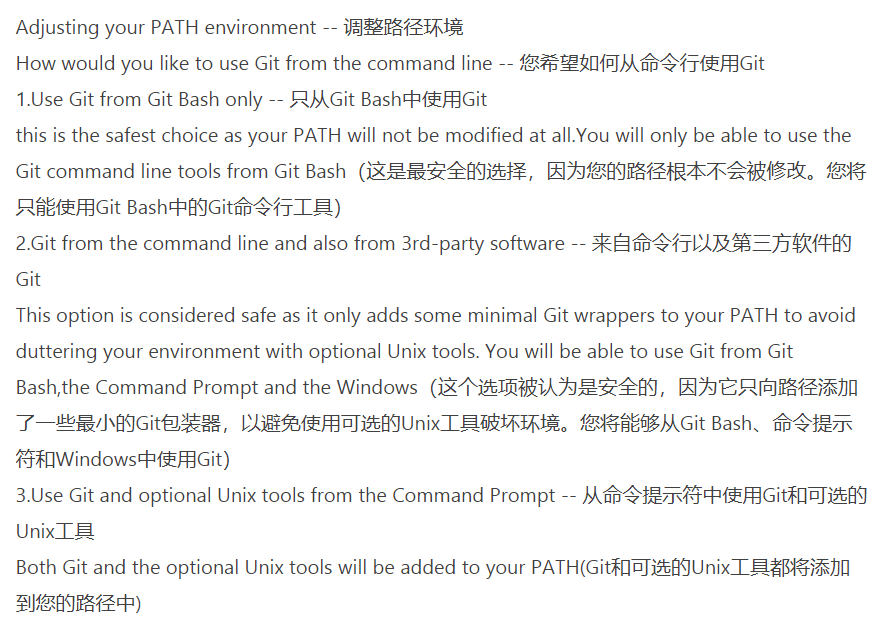
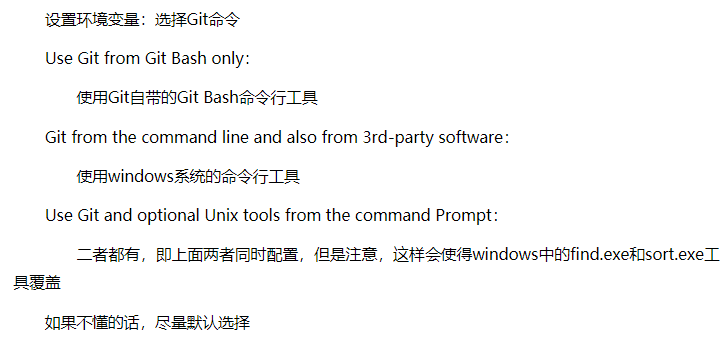
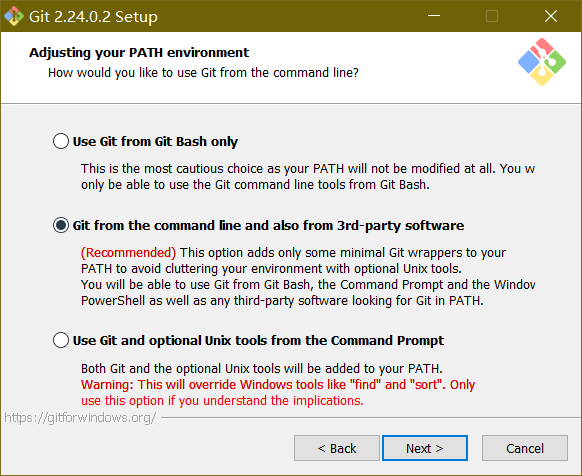
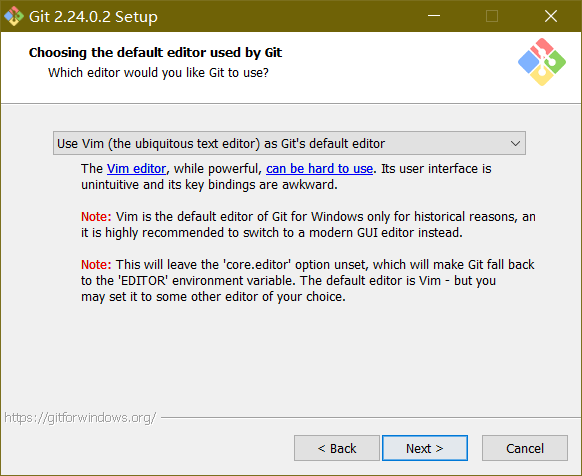
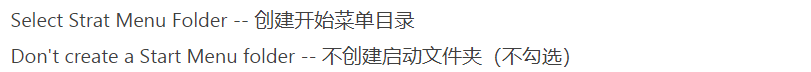
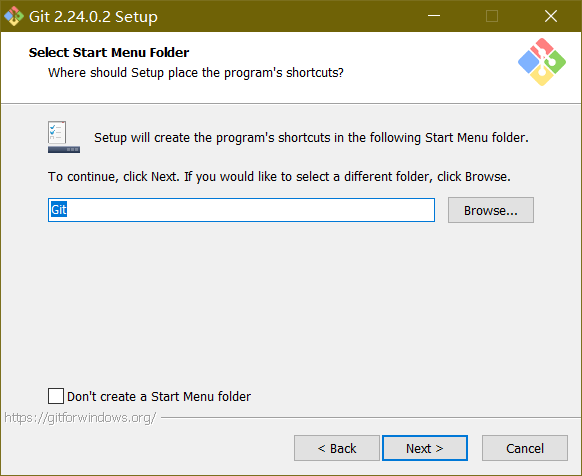
下载

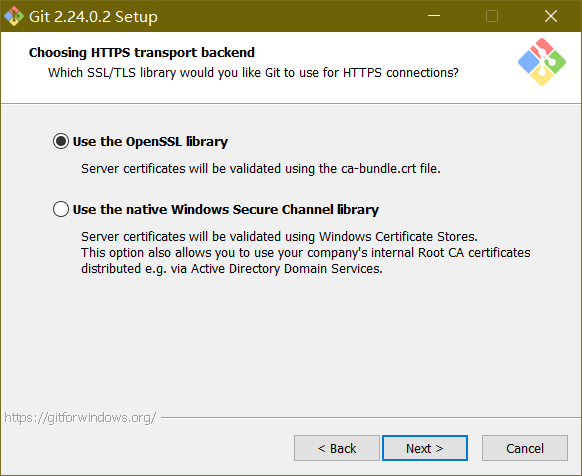


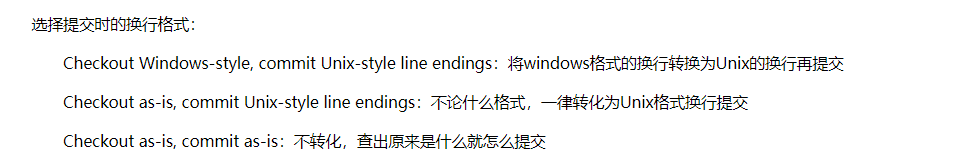
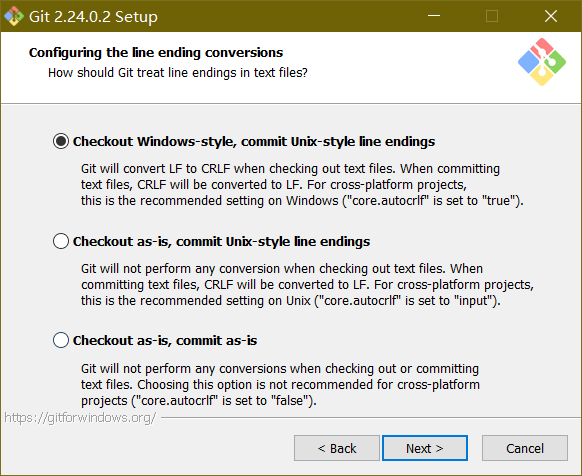
1. 安装程序如下

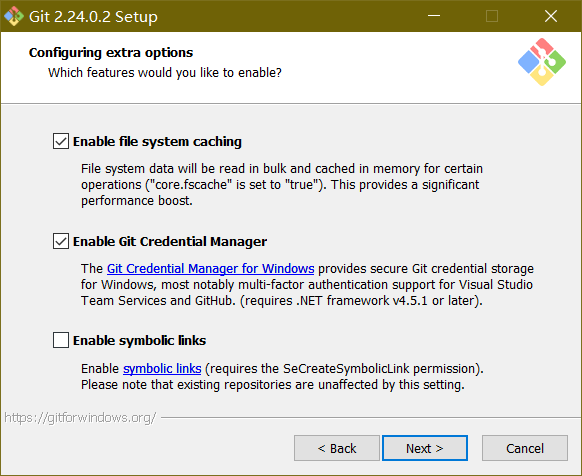
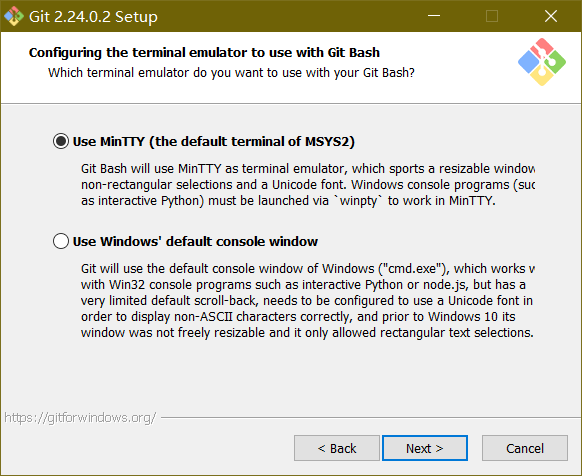




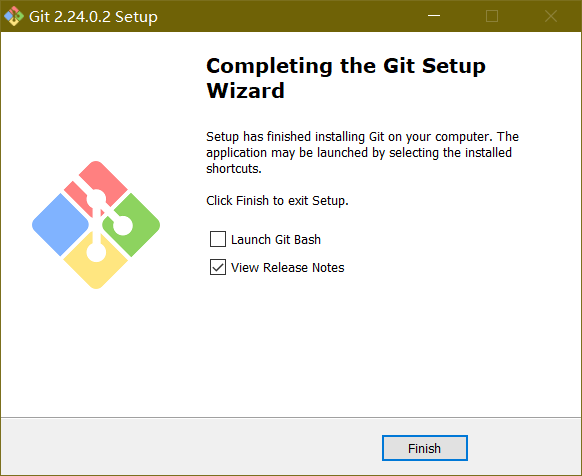
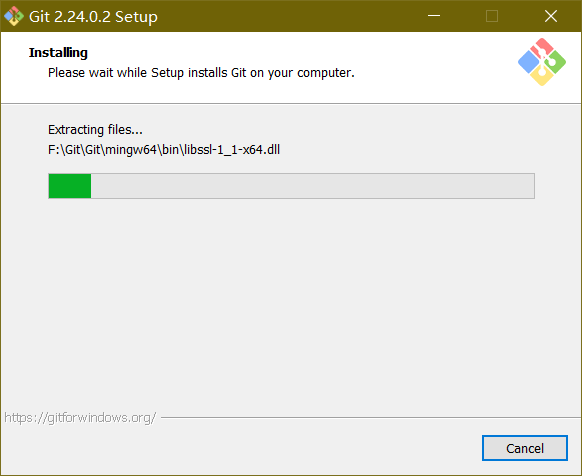








默认不勾选



Release Notes

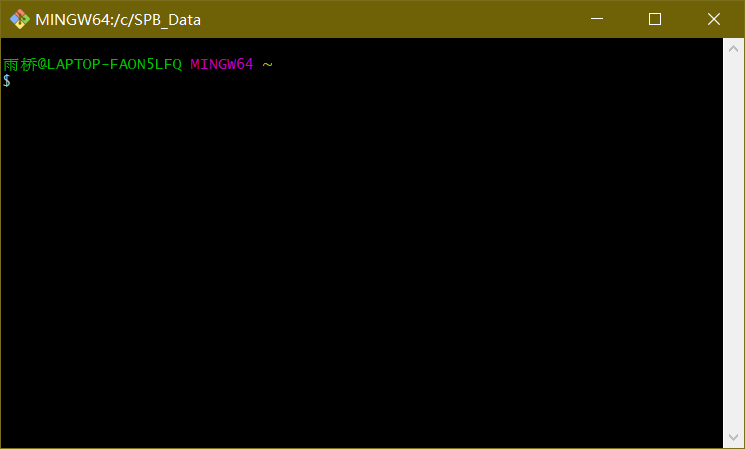
<file:///F:/Git/Git/ReleaseNotes.html>

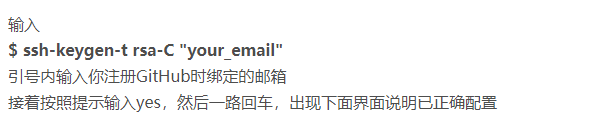
## **Git绑定GitHub账号**

### Git Bash与Github关联

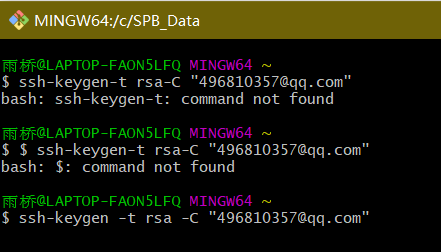


打开Git Bash

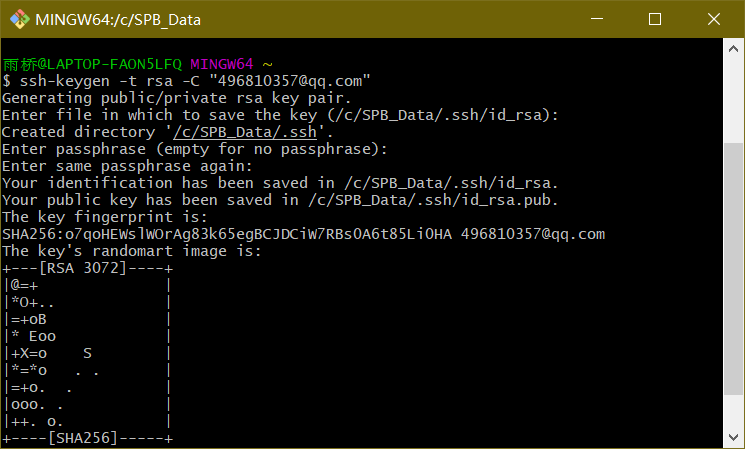




**$ ssh-keygen -t rsa-C "496810357@qq.com"**

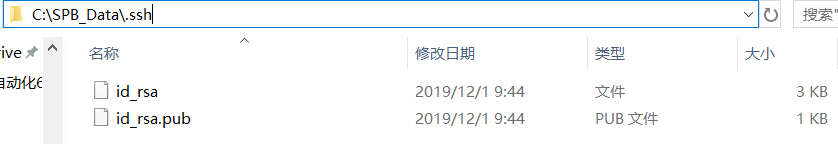


回车



由内容可以看到和教程不同，默认存储到了

C:\SPB\_Data\.ssh



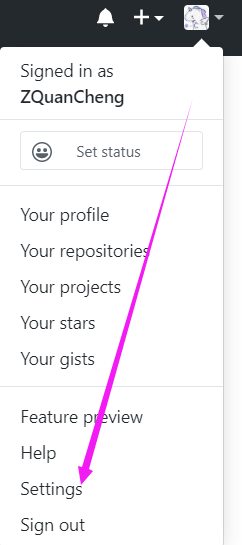
接着使用记事本打开id\_rsa.pub文件，内容便是创建的密钥

ssh-rsa  496810357@qq.com

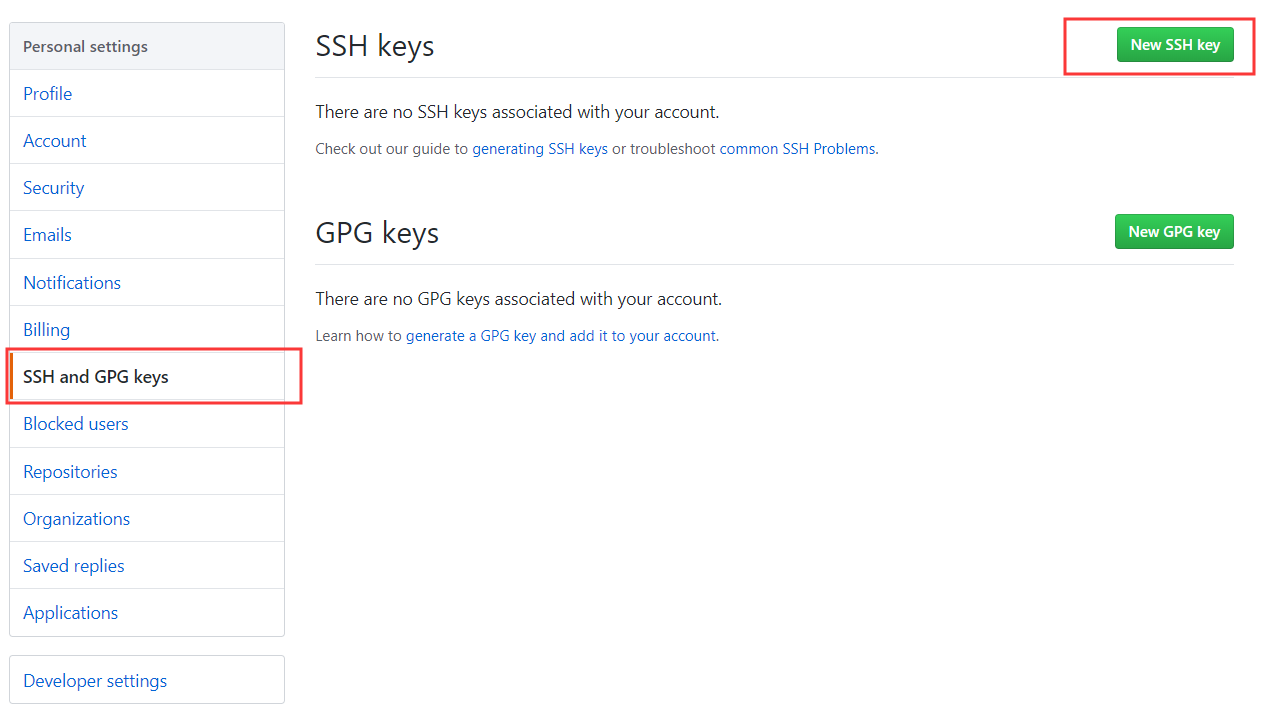
### Github

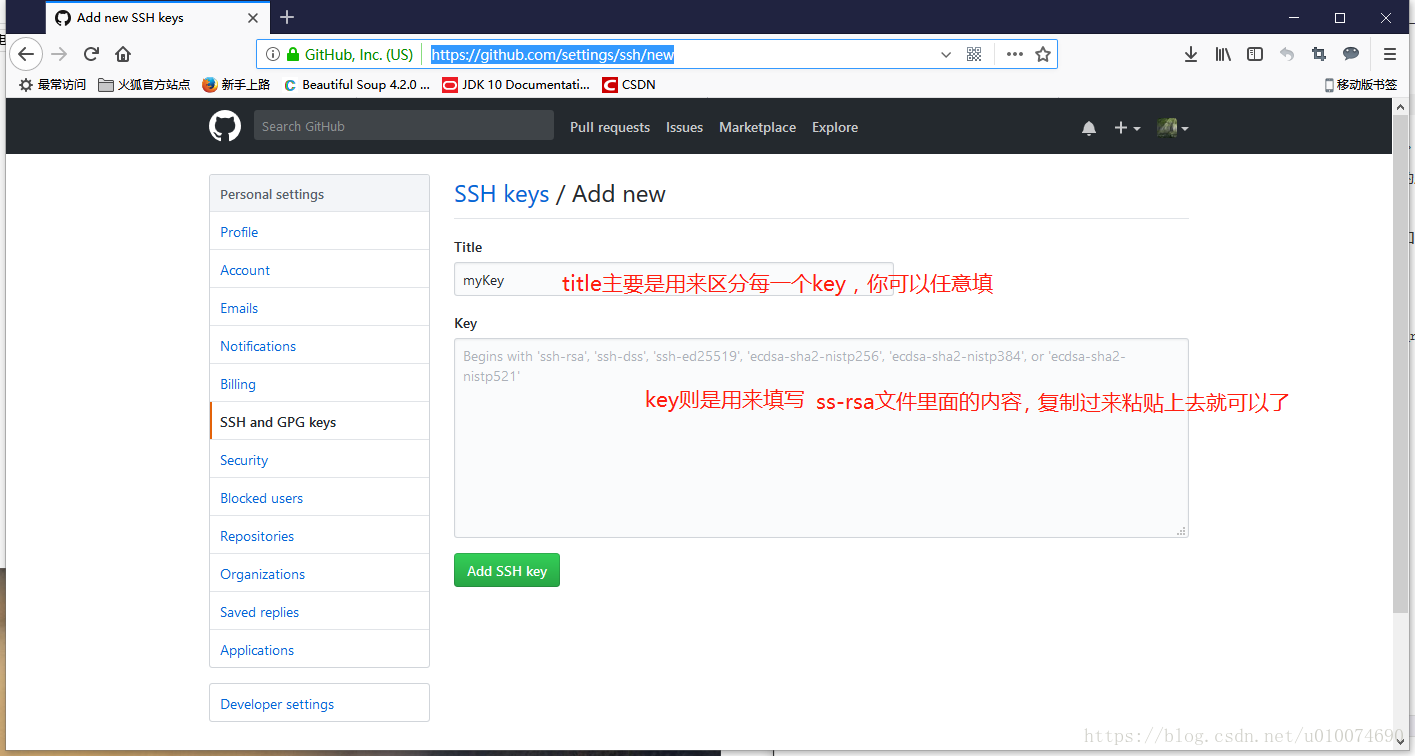
登录你的github,然后访问：https://github.com/settings/ssh/new

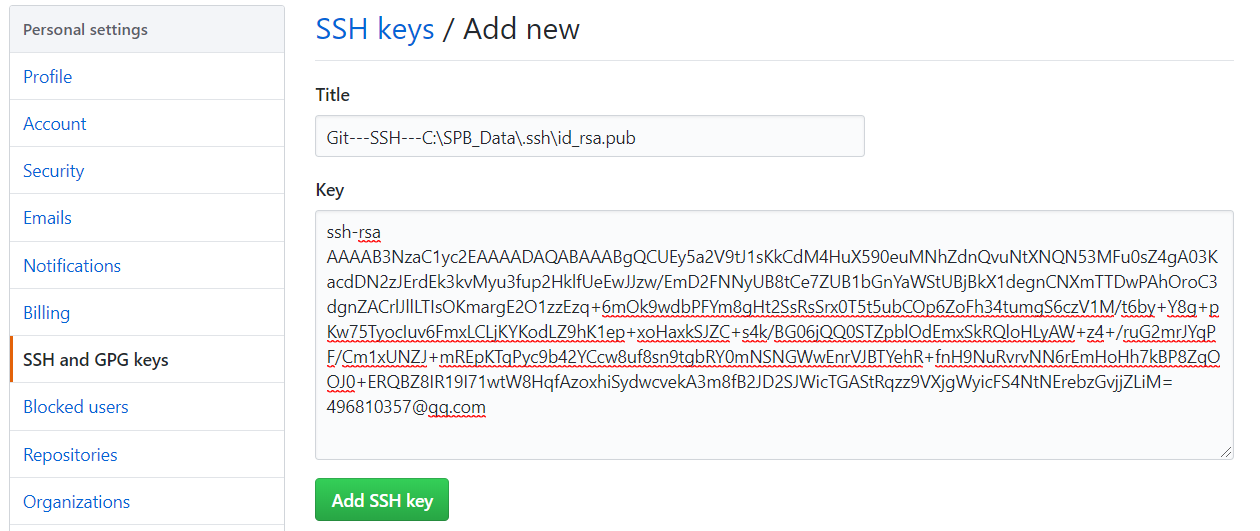
点击settings



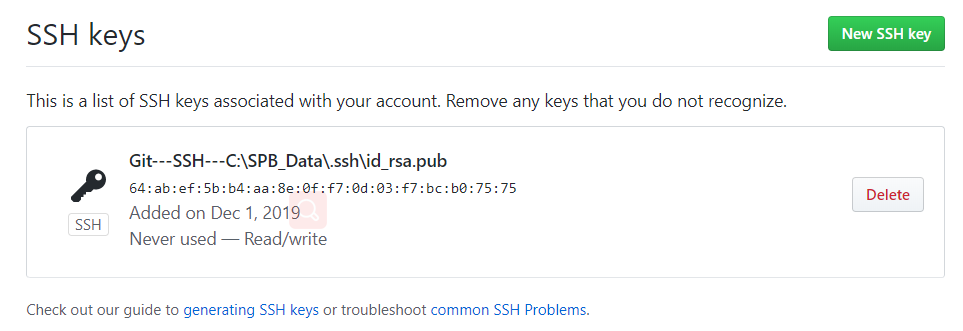
点击New SSH key







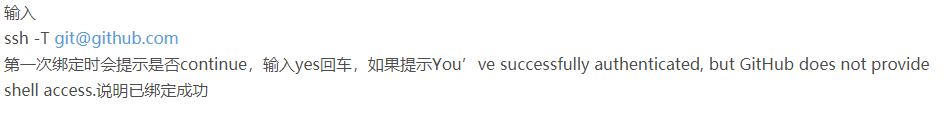
Add SSH key



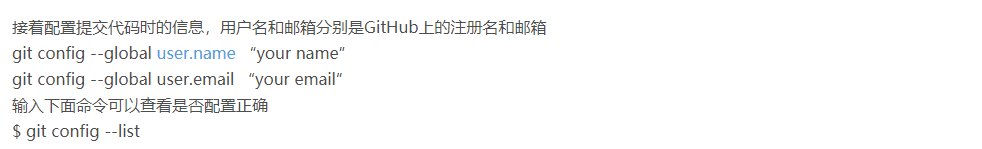
添加完SSH Keys之后,本地Git便与远程Github建立了连接

**然后我们变开始我们的Git的使用：**

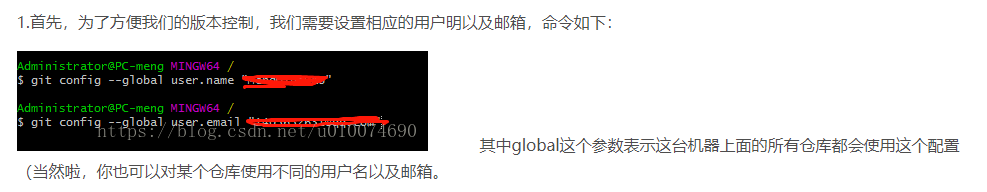
### 回到Git Bash





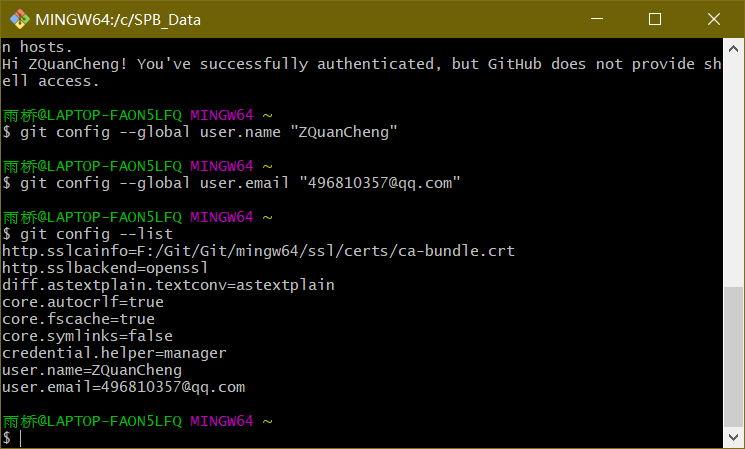


为了方便我们的版本控制，我们需要设置相应的用户明以及邮箱，命令如下：



git config --global [user.name](http://user.name/) “ZQuanCheng”

git config --global user.email “496810357@qq.com”

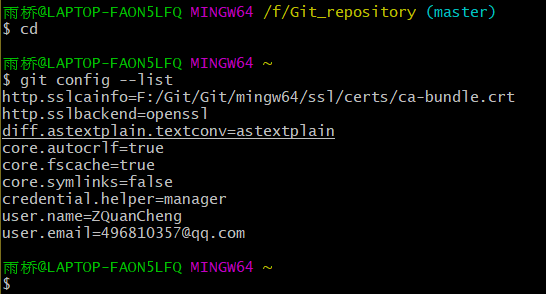


配置正确

### 把GitHub的资源clone到本地计算机仓库

#### 检查连接

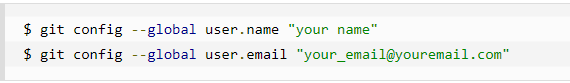
git config –-list



确认username和email

如不对

之前还需要设置username和email，因为github每次commit都会记录他们。



git config --global [user.name](http://user.name/) “ZQuanCheng”

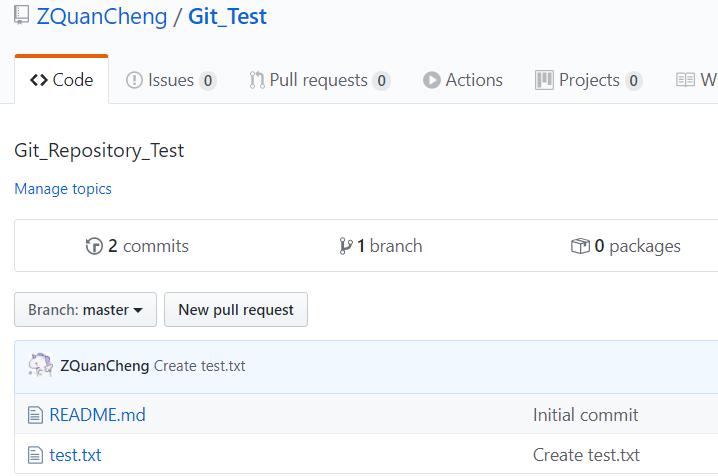
git config --global user.email “496810357@qq.com”

检查是对的，不要重新设置

#### 开始操作

创建一个新的Repositories（勾选with a README）

添加一个txt文件



不创建也行，只把最原始的Repository Clone下来

新建一个文件夹F:\Git\_test

在文件夹中打开Git Bash





复制Web URL

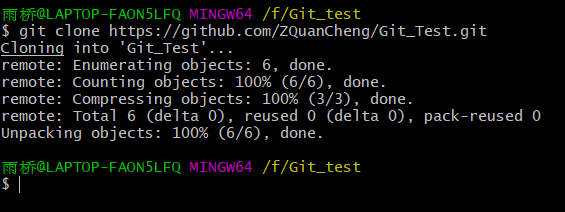
https://github.com/ZQuanCheng/Git\_Test.git



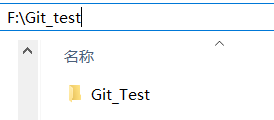
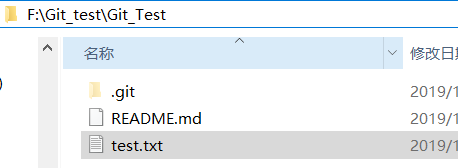


Git clone https://github.com/ZQuanCheng/Git\_Test.git

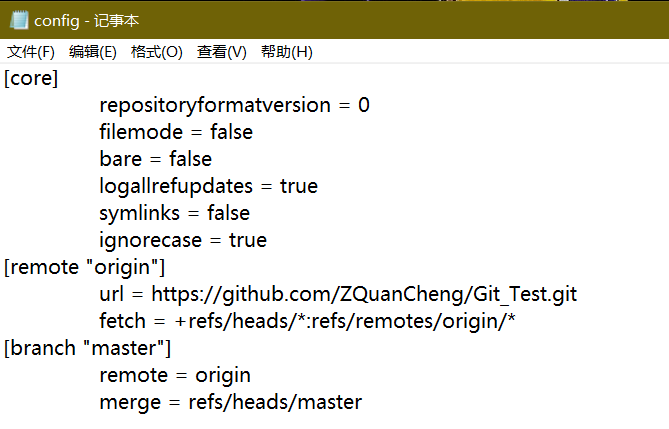
 



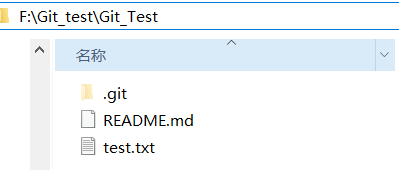
可以用Ctrl+C强行停止

\Git\_Test\.git\ config打开

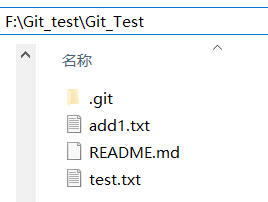


### 更新clone的仓库的内容到Github



#### 添加文件

添加一个文件add1.txt



但是这个文件在仓库中没有记录

每次添加文件都必须记录

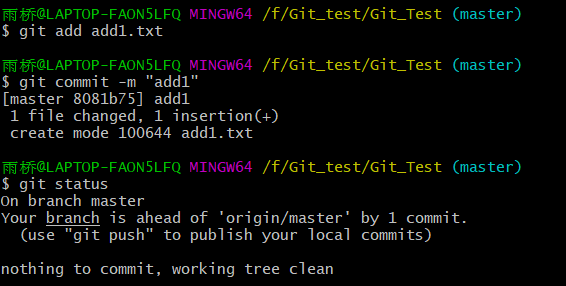
然后

打开 Git Bash

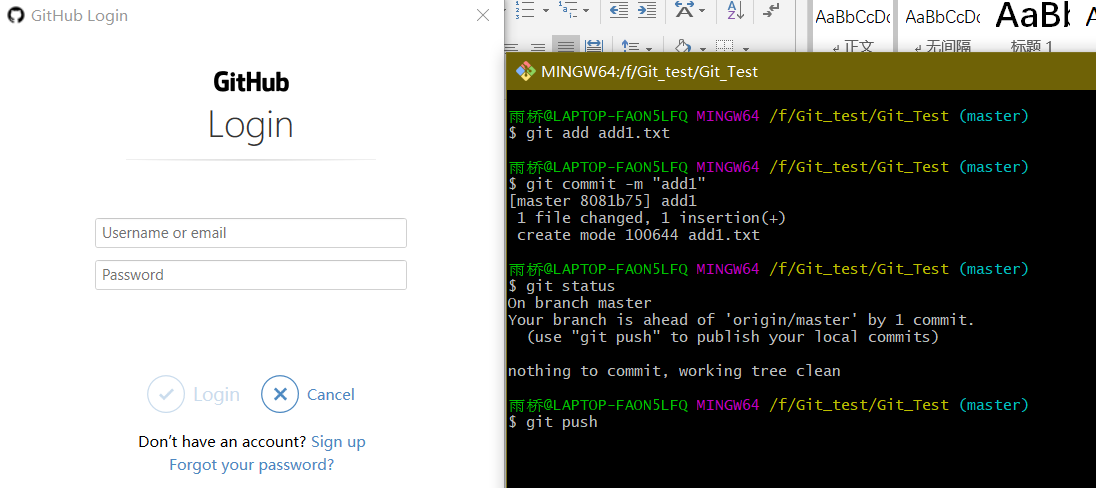
git add <filename>

git commit -m ‘添加说明‘

git status（可没有，只是用来查看仓库状态）



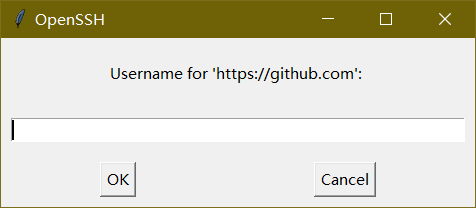
git push 同步到远程仓库



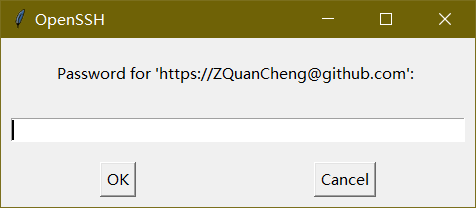
弹出登陆界面

输入名称和密码，Login

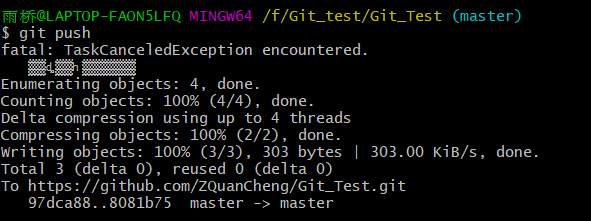
有弹出



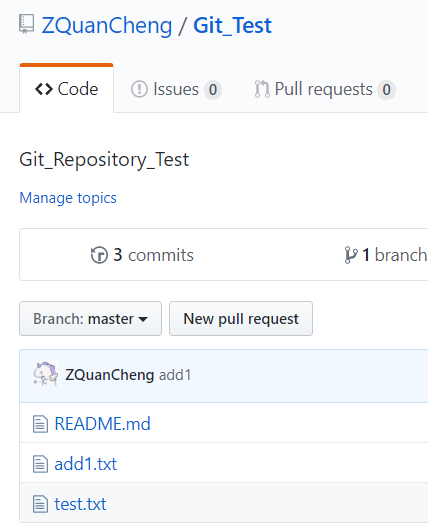
ZQuanCheng

  
输入密码

有了

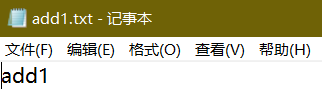
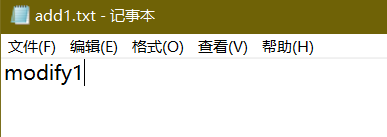


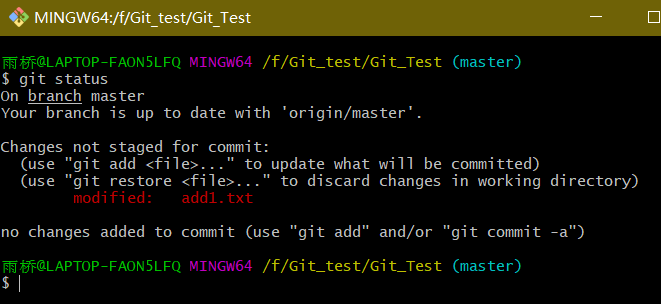
同时在Github上会有显示commit信息



成功

#### 修改文件

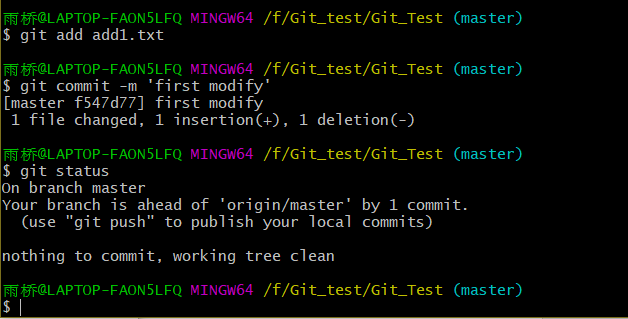


查看状态，被修改了，但是仓库还没有记录修改

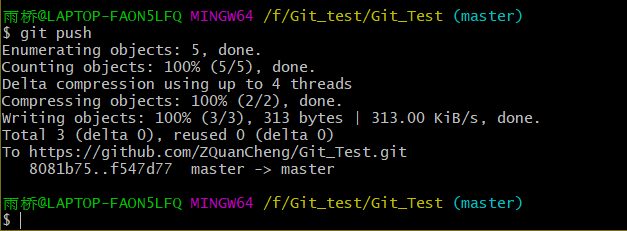
git add <filename>

git commit -m ‘添加说明‘

git status（可没有，只是用来查看仓库状态）



git push同步到远程仓库

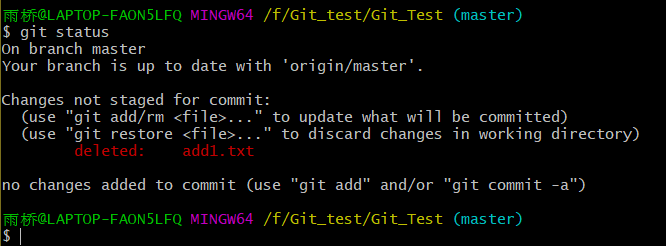


同时在Github上会有显示commit信息



#### 删除文件

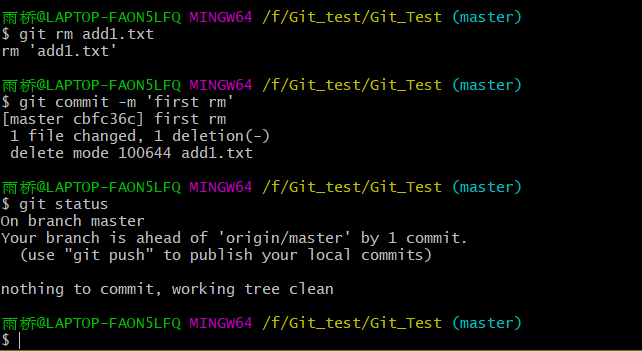


查看状态，被删除了，但是仓库还没有记录删除

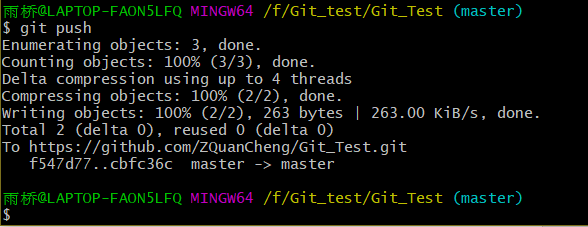
git rm <filename>

git commit -m ‘添加说明‘

git status（可没有，只是用来查看仓库状态）



Git push 同步到远程仓库



同时在Github上会有显示commit信息

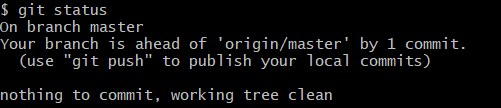


#### 同步到Github远程仓库

每次git commit -m ‘添加说明‘

然后git status 查看仓库状态

会提醒你



Git push命令 将本地仓库内容同步到Github远程仓库

### **在指定文件目录并通过init命令创建仓库**

#### 创建仓库

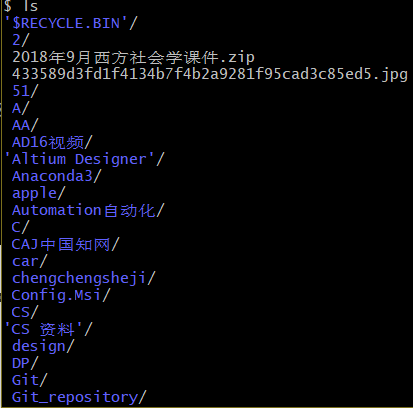


ls命令是linux下最常用的命令之一，ls跟dos下的dir命令是一样的都是用来列出目录下的文件。

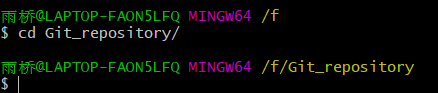
实际操作

创建F:\Git\_repository文件夹

再次ls



输入 cd Git\_repository/ 打开Git\_repository文件夹

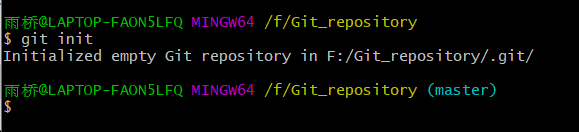


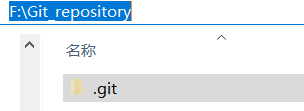
也可以从要创建仓库的文件夹进入Git Bash





输入git init

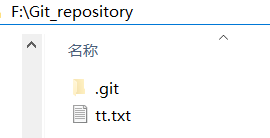




首先要明确下，所有的版本控制系统，只能跟踪文本文件的改动，比如txt文件，网页，所有程序的代码等，Git也不列外，版本控制系统可以告诉你每次的改动，但是图片，视频这些二进制文件，虽能也能由版本控制系统管理，但没法跟踪文件的变化，只能把二进制文件每次改动串起来，也就是知道图片从1kb变成2kb，但是到底改了啥，版本控制也不知道。

完成好以上步骤的话，一个Git的环境就基本搭建完毕了

#### 添加文件



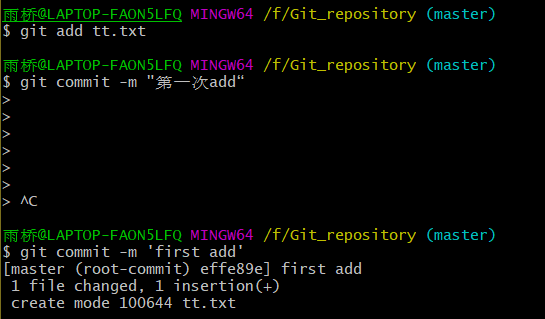
但是这个文件在仓库中没有记录

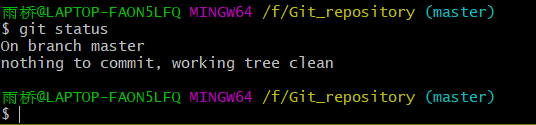
每次添加文件都必须

git add <filename>

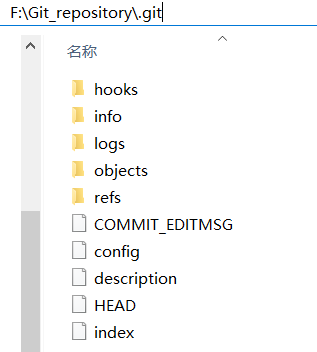
git commit -m ‘添加说明‘

git status（可没有，只是用来查看仓库状态）

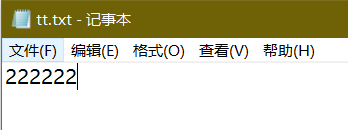
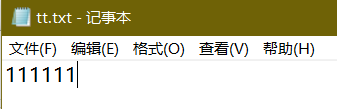


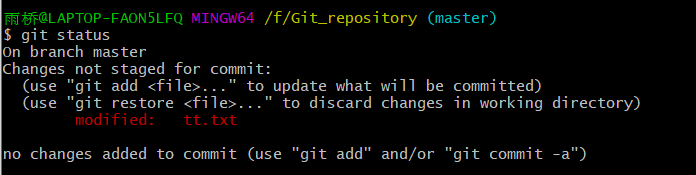


第一次添加commit后.git文件夹也变了



#### 修改文件



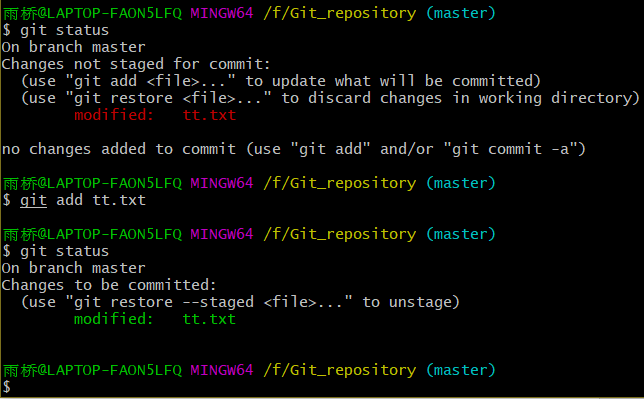


查看状态，被修改了，但是仓库还没有记录修改

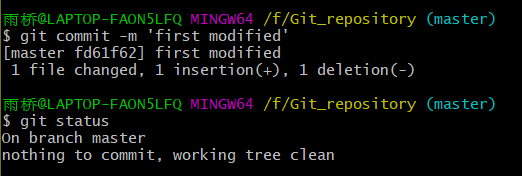
git add <filename>

git commit -m “添加说明”

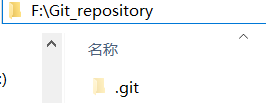
git status（可没有，只是用来查看仓库状态）



已经记录修改



#### 删除文件

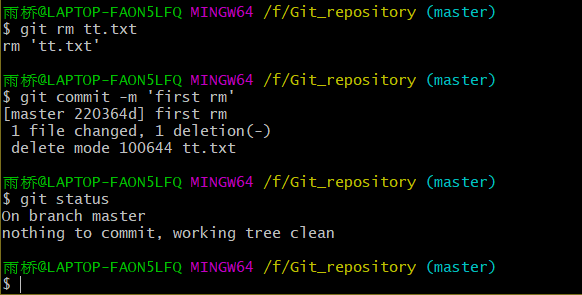


查看状态，被删除了，但是仓库还没有记录删除

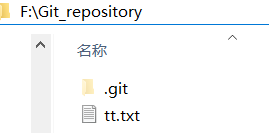
git rm <filename>

git commit -m “添加说明”

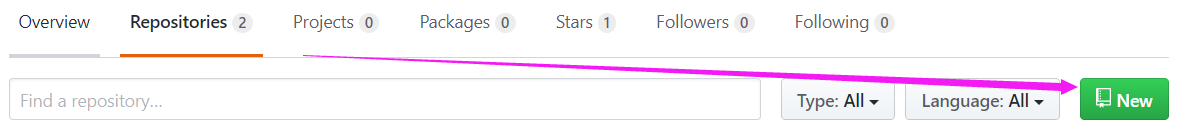
git status（可没有，只是用来查看仓库状态）

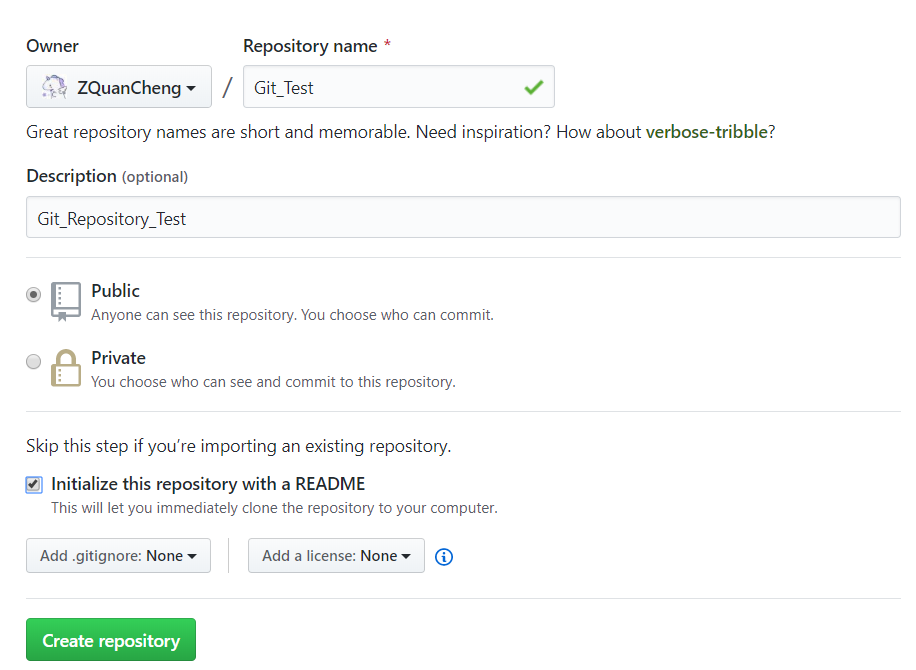


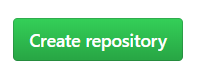
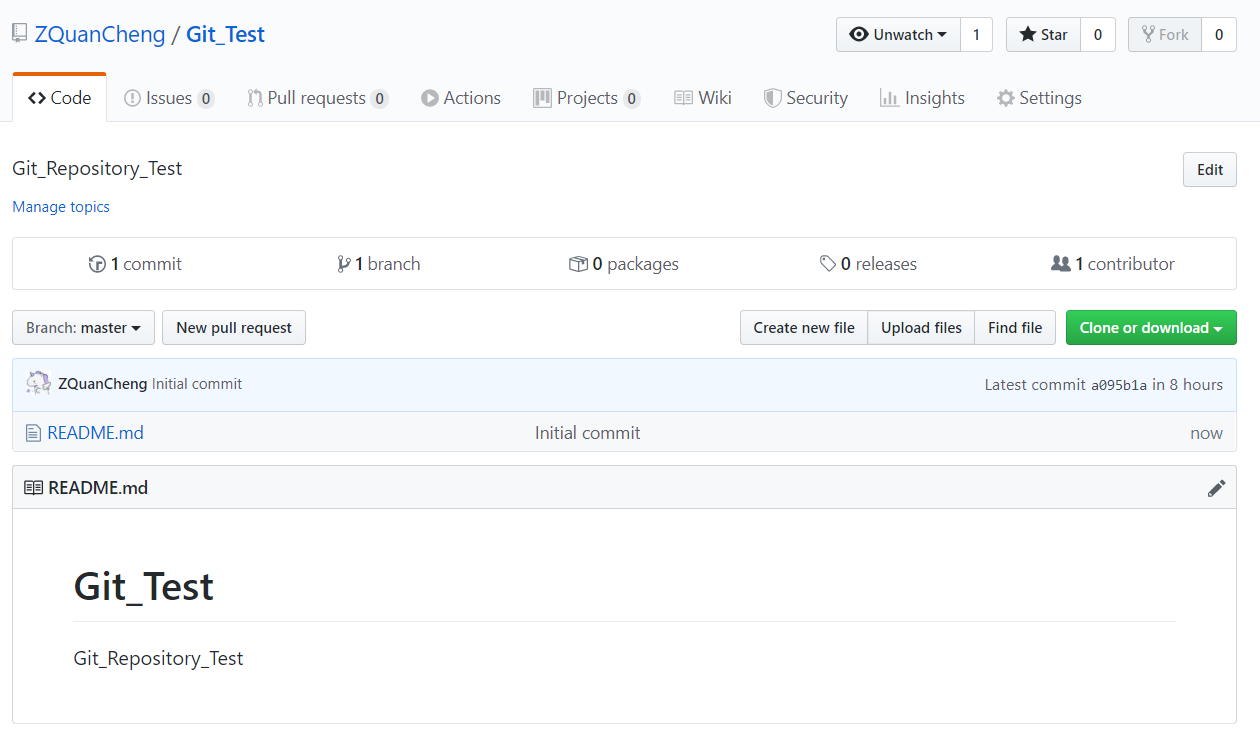
### 将本地计算机的仓库内容上传到指定的Github的 Repository（在此之前没有关联）

为要上传的本地仓库创建一个新的Repositories（勾选with a README）（之前依旧创建好一个仓库了F:\Git\_repository）





创建成功

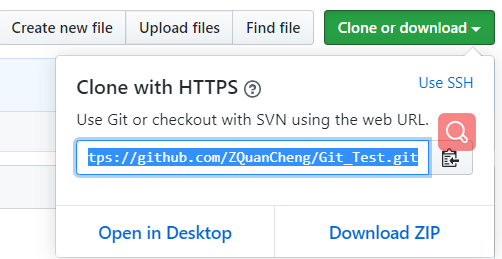


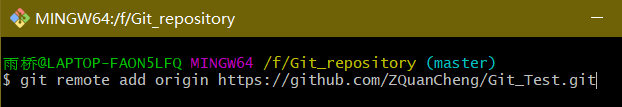
打开Git Bash

#### 方法一（成功）

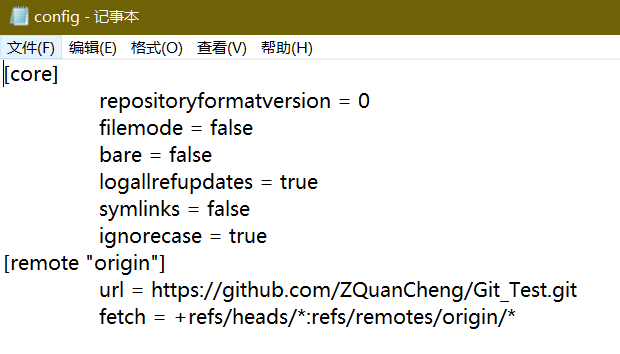
在本地仓库打开Git Bash

Git remote add origin <远程仓库地址>



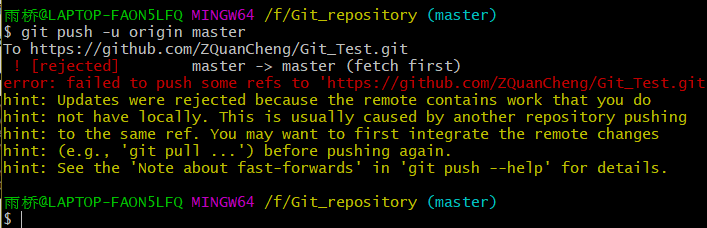


打开F:\Git\_repository\.git\ config



建立了连接

git push -u origin master（因为是第一次所以加了-u参数，以后push就不用加了）



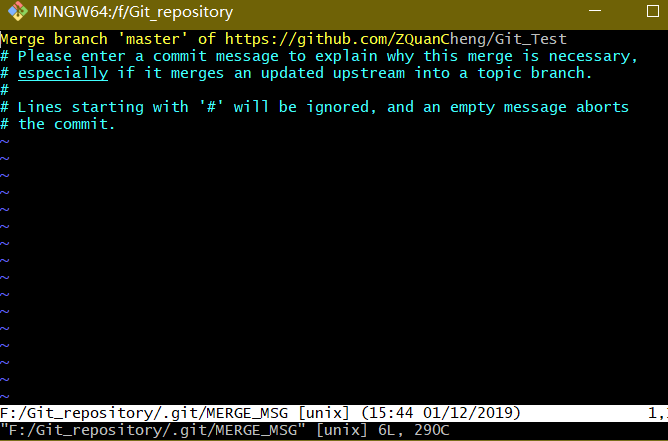
报错

git pull origin master --allow-unrelated-histories

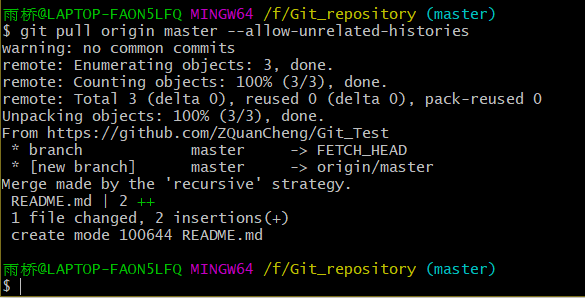


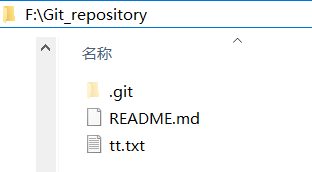
SSH远程连接错误

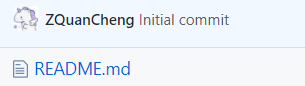
再来git push origin master --allow-unrelated-histories



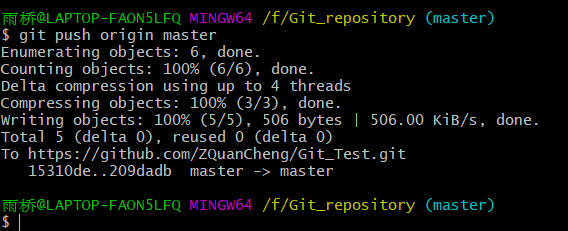
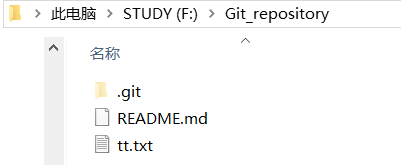
键盘按下冒号：，然后q，回车

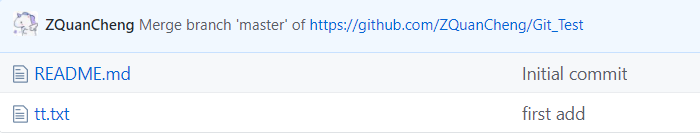




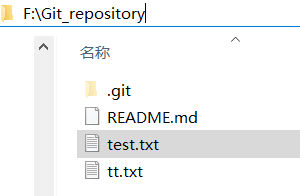


git push origin master（不是第一次，所以没有-u参数）

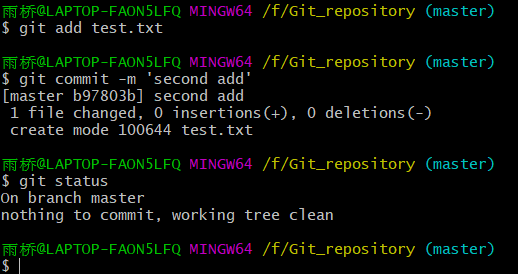
  




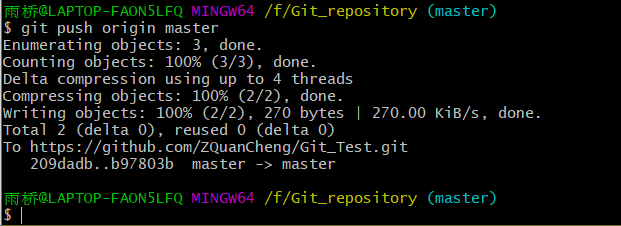
接下来在本地仓库里添加一些文件，比如test.txt

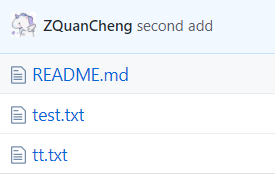


然后在命令行输入以下命令



然后





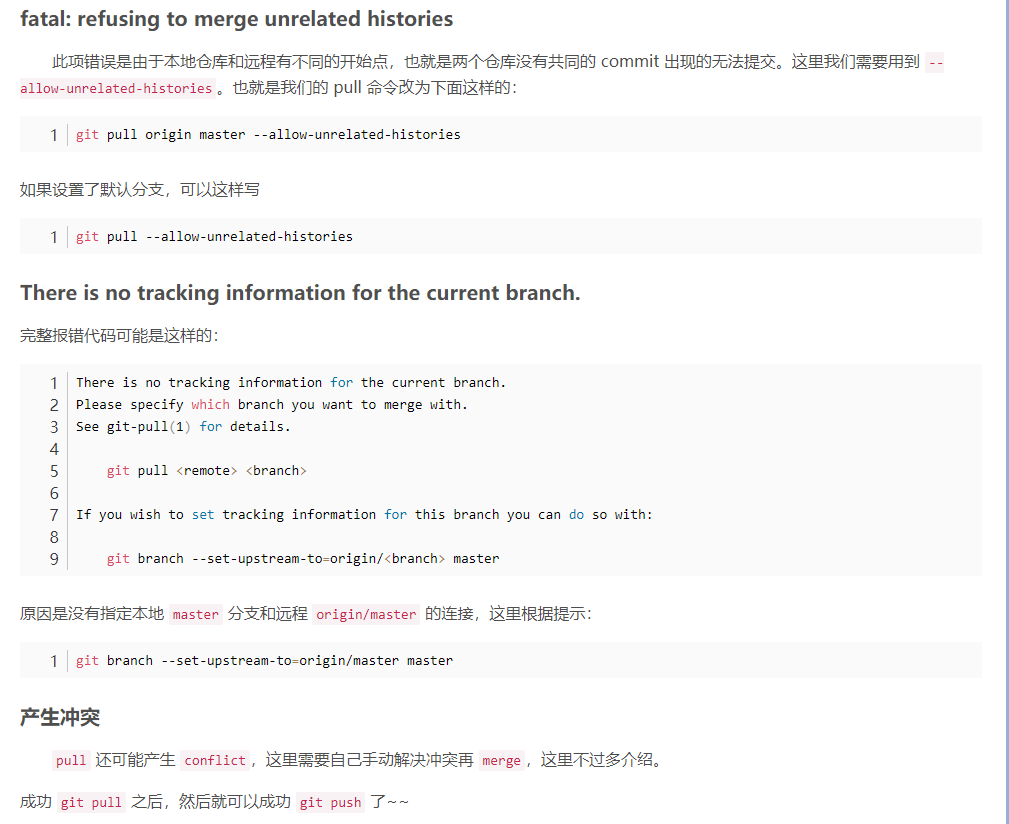
git push命令会将本地仓库推送到远程服务器。

git pull命令则相反。

注：首次提交，先git pull下，修改完代码后，使用git status可以查看文件的差别，使用git add 添加要commit的文件。

#### 问题分析

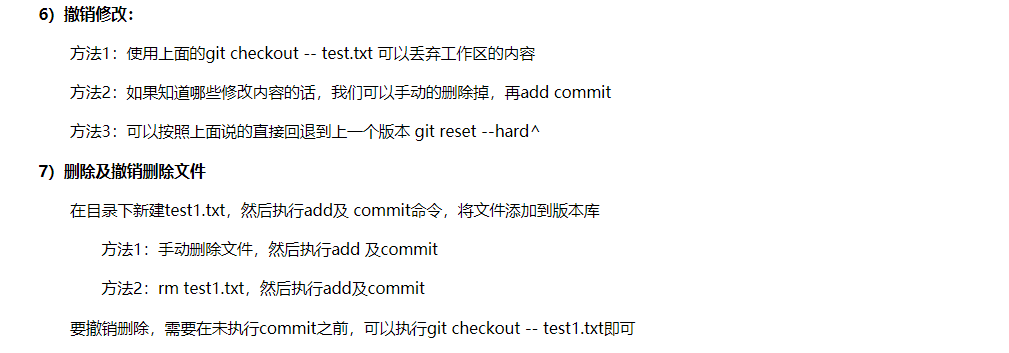
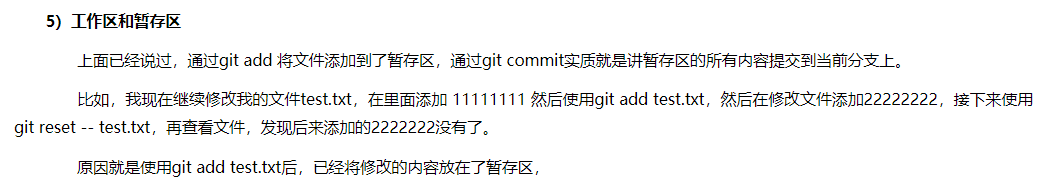


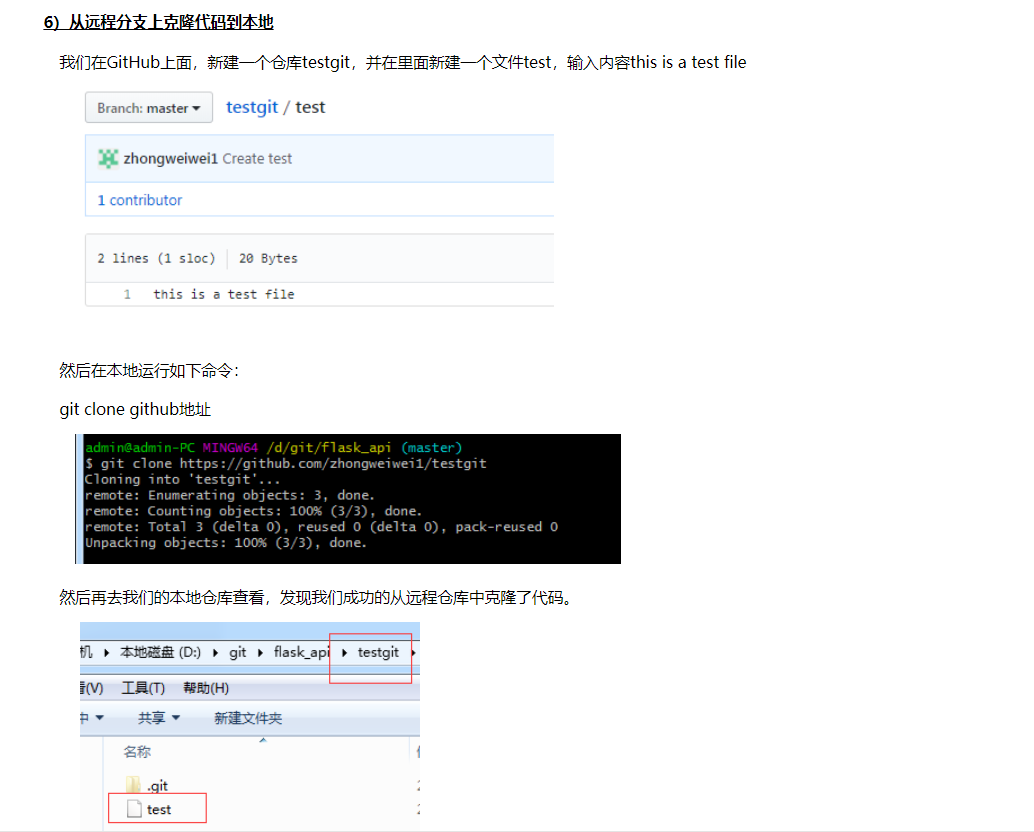
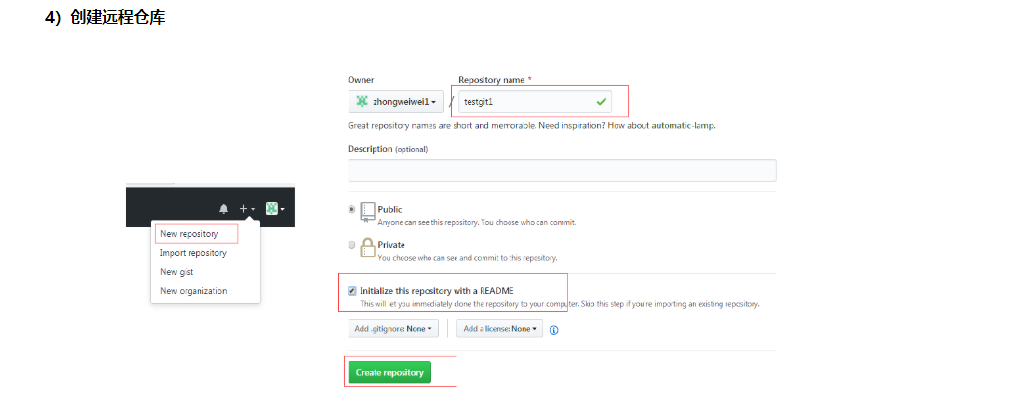
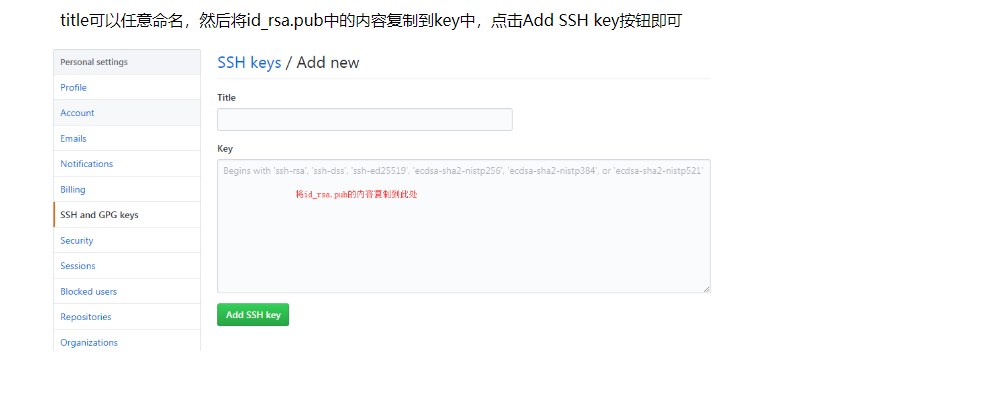
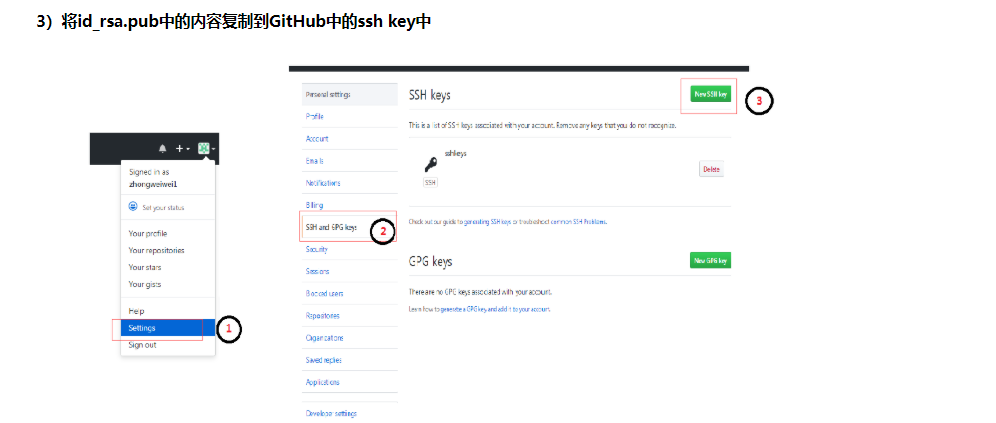
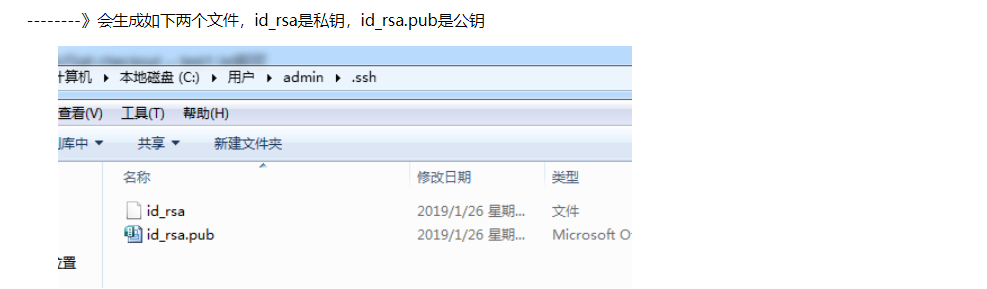
# **以下是每个要点的详细讲解**

## **4、设置用户名和邮箱**

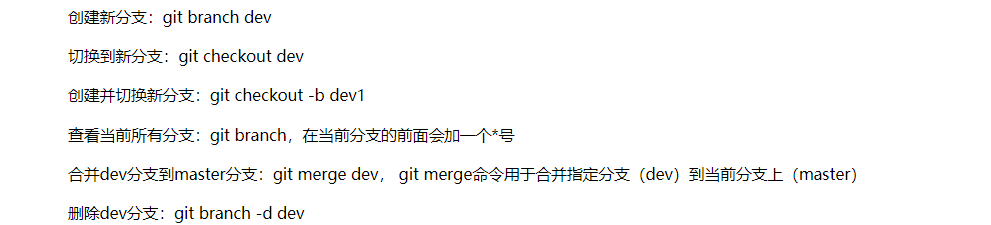
## **5、使用本地仓库**



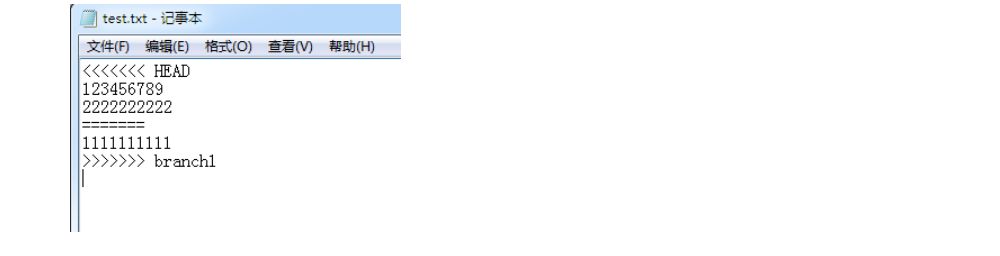
## **6、使用远程仓库**



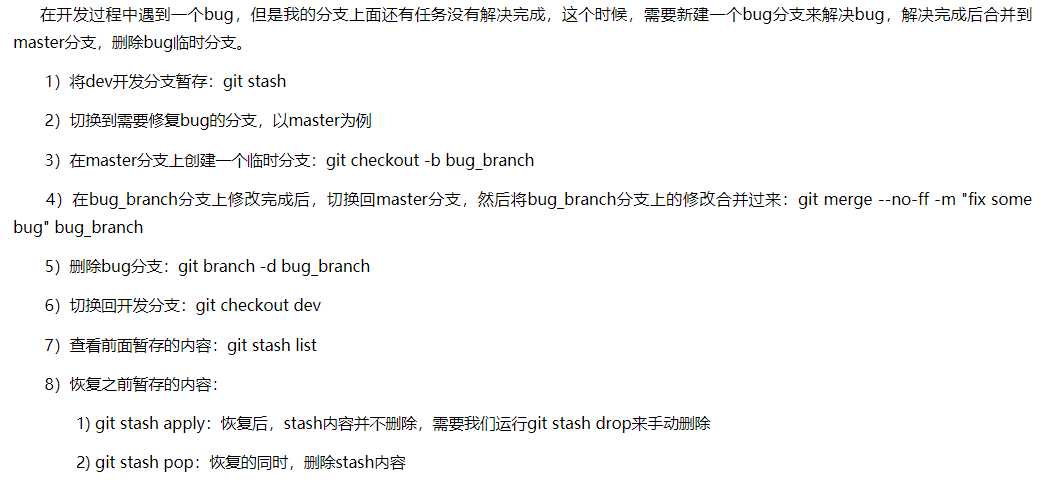
## 7、创建与合并分支



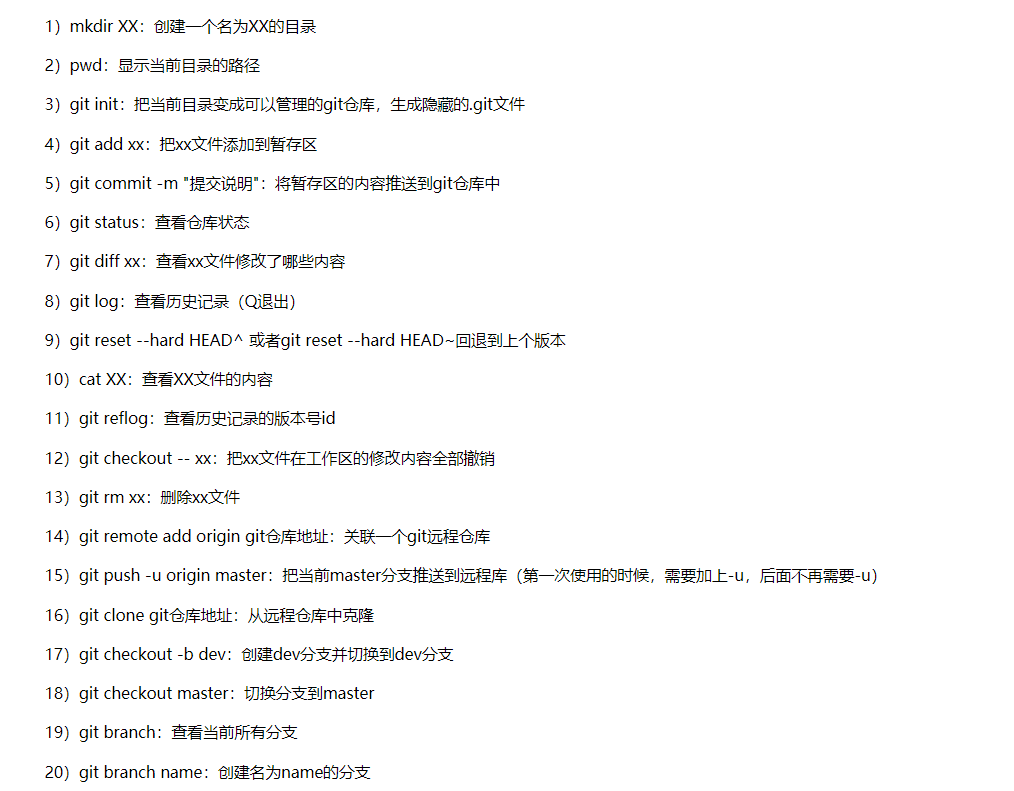
## 8、解决冲突



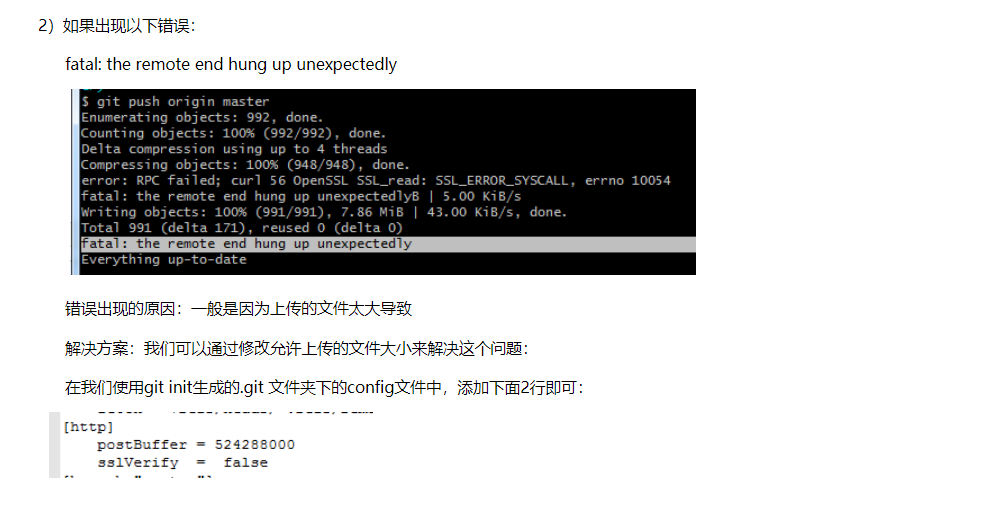
## 9、bug分支



## 10、Git常用命令：

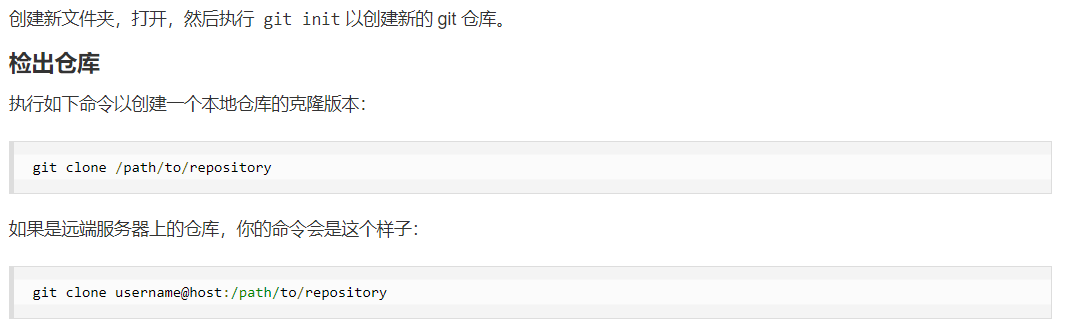
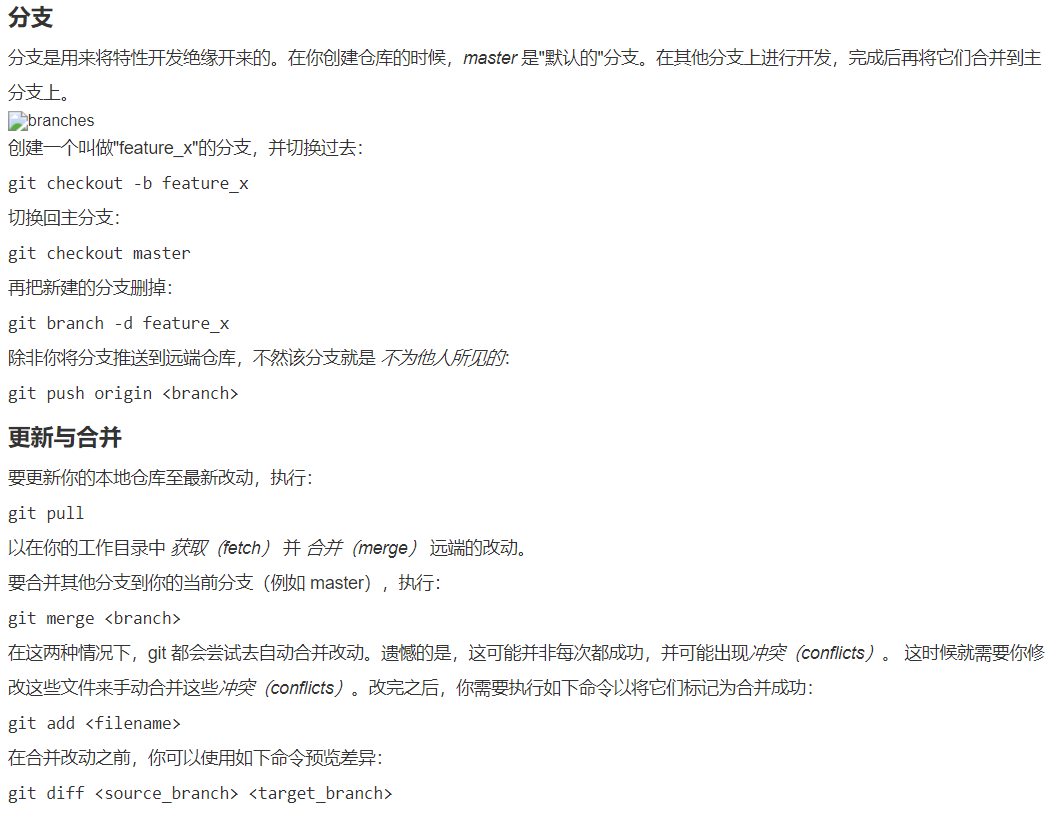


## 11、常见错误：



## 12. 一个简明教程

<https://www.runoob.com/w3cnote/git-guide.html>

Github 简明指南：<http://rogerdudler.github.io/git-guide/index.zh.html>

如何高效利用GitHub:<http://www.yangzhiping.com/tech/github.html>