# Windows下Git安装及基本使用

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 备注 |
| V1 | 2019-09-08 | 谭祖好 | 1. 安装文件：/附件/Git-2.23.0-64-bit.exe. 2. git教程地址l:   <https://www.runoob.com/git/git-tutorial.html> |
| V2 | 2019-09-14 | 谭祖好 | 1. 提交代码 2. 删除github上文件 3. 更新github上文件 4. 同步代码（合并代码） |

## 1.Git简介

Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。Git 与常用的版本控制工具 CVS, Subversion 等不同，它采用了分布式版本库的方式，不必服务器端软件支持。

## 2.Gti和SVN区别

Git 不仅仅是个版本控制系统，它也是个内容管理系统(CMS)，工作管理系统等。如果你是一个具有使用 SVN 背景的人，你需要做一定的思想转换，来适应 Git 提供的一些概念和特征。

Git 与 SVN 区别点：

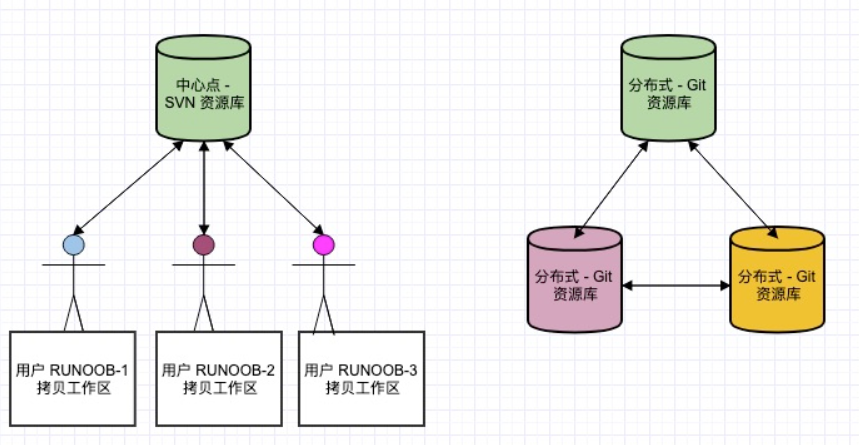
2.1、Git 是分布式的，SVN 不是。这是 Git 和其它非分布式的版本控制系统，例如 SVN，CVS 等，最核心的区别。

2.2、Git 把内容按元数据方式存储，而 SVN 是按文件：所有的资源控制系统都是把文件的元信息隐藏在一个类似 .svn、.cvs 等的文件夹里。

2.3、Git 分支和 SVN 的分支不同：分支在 SVN 中一点都不特别，其实它就是版本库中的另外一个目录。

2.4、Git 没有一个全局的版本号，而 SVN 有：目前为止这是跟 SVN 相比 Git 缺少的最大的一个特征。

2.5、Git 的内容完整性要优于 SVN：Git 的内容存储使用的是 SHA-1 哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏。

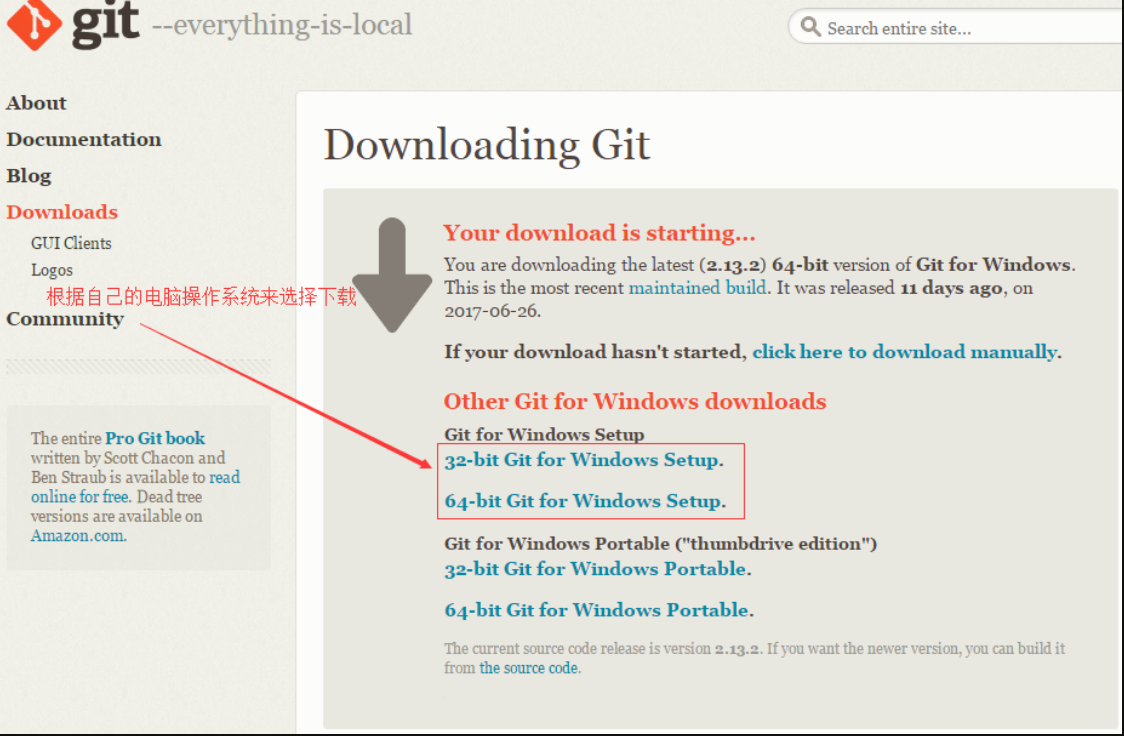


## Git安装

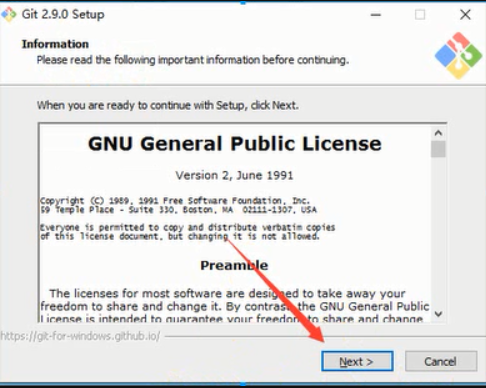
### 3.1.下载Git

使用提供的安装包，其路径为：/附件/Git-2.23.0-64-bit.exe，

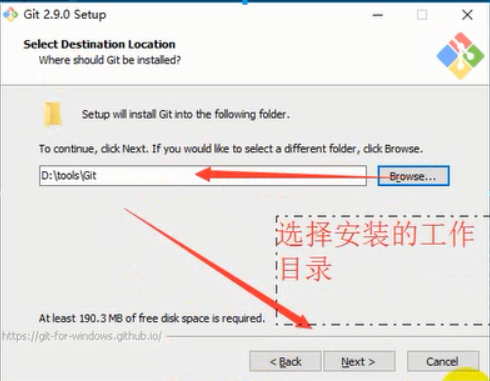
或者去官网下载，其地址为：<https://git-scm.com/download/win>



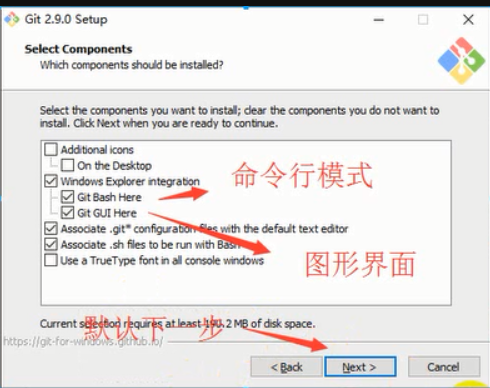
### 下载完之后，双击安装



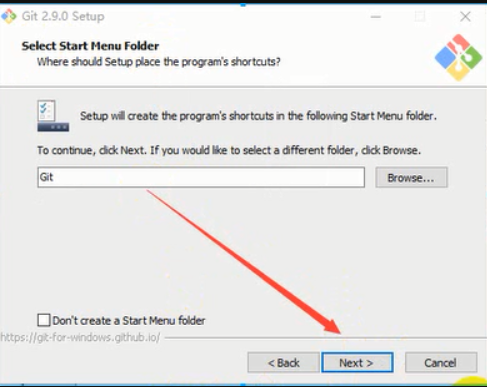
### 选择安装目录



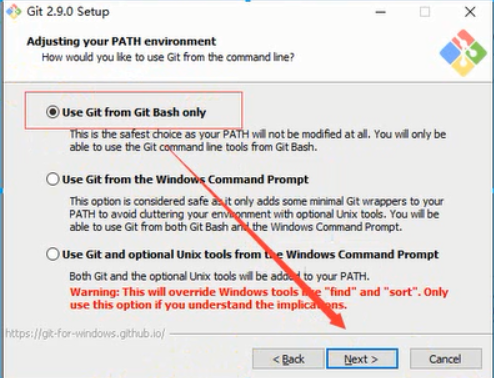
### 选择组件



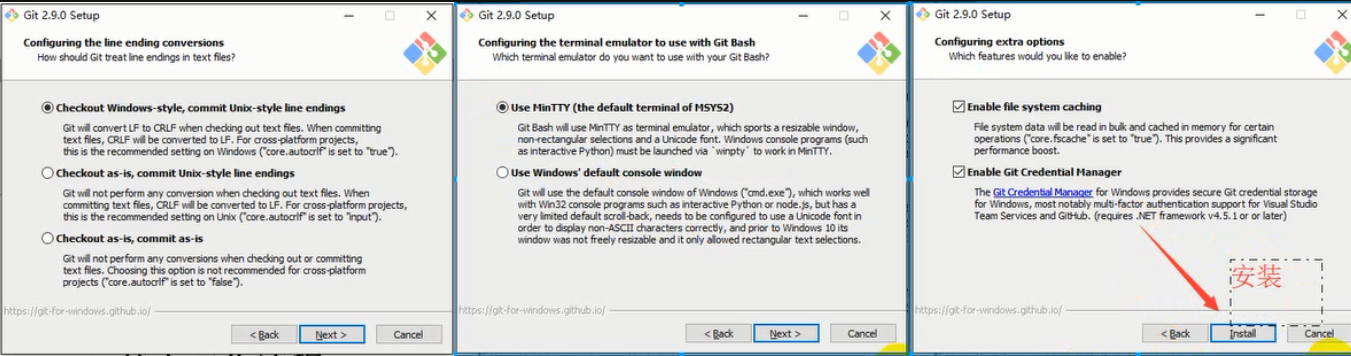
### 开始菜单目录名设置



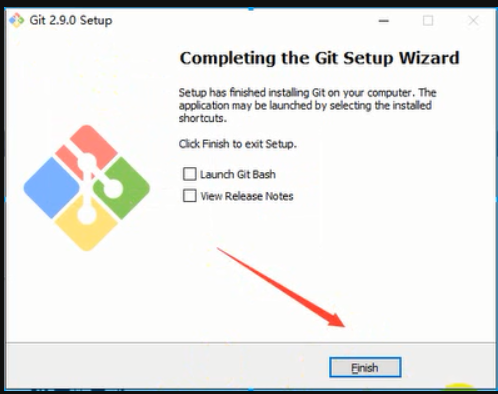
### 选择使用命令行环境



### 以下三步默认，直接点击下一步



### 3.8.安装完成



### 3.8.检验是否安装成功

回到电脑桌面，鼠标右击如果看到有两个git单词则安装成功。

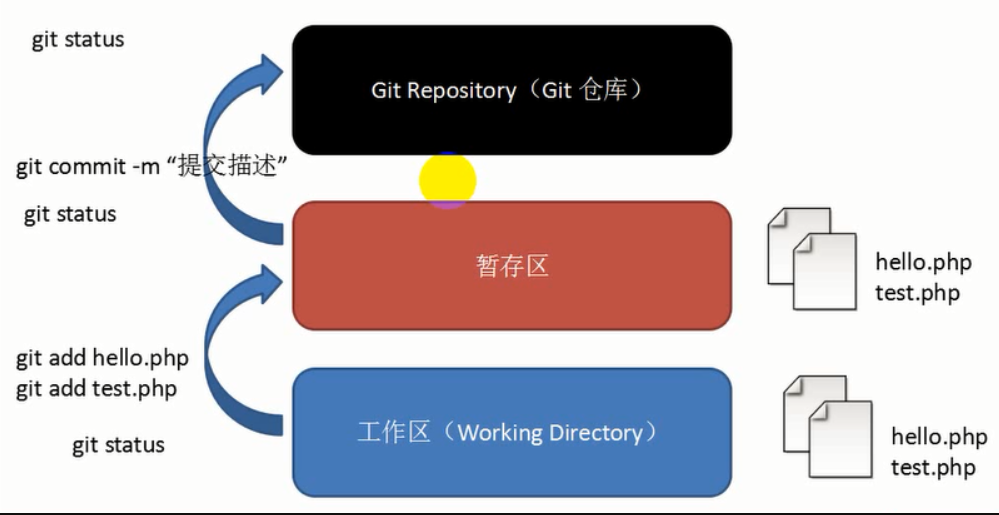


## Git基本工作流程

### 4.1. Git工作区域



### 向仓库中添加文件流程



## 申请Github账号

### 5.1.Github简介

github是一个基于git的代码托管平台，付费用户可以建私人仓库，我们一般的免费用户只能使用公共仓库，也就是代码要公开。

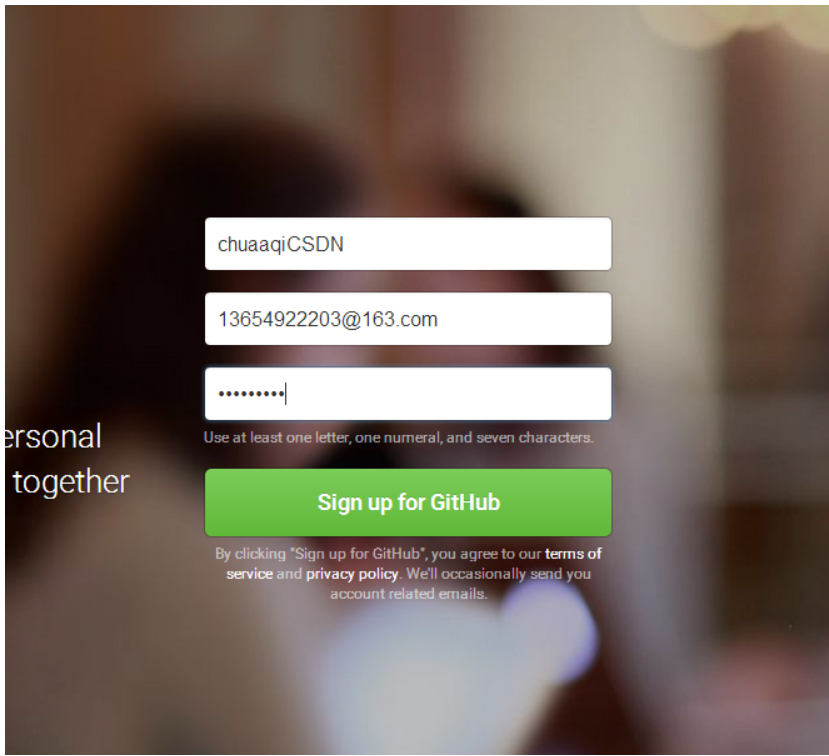
github官网地址：<https://github.com/>

**注意:Github不支持国内浏览器和IE浏览器, 请使用最新版Chrome或Firefox或苹果浏览器。**

### Github账号注册

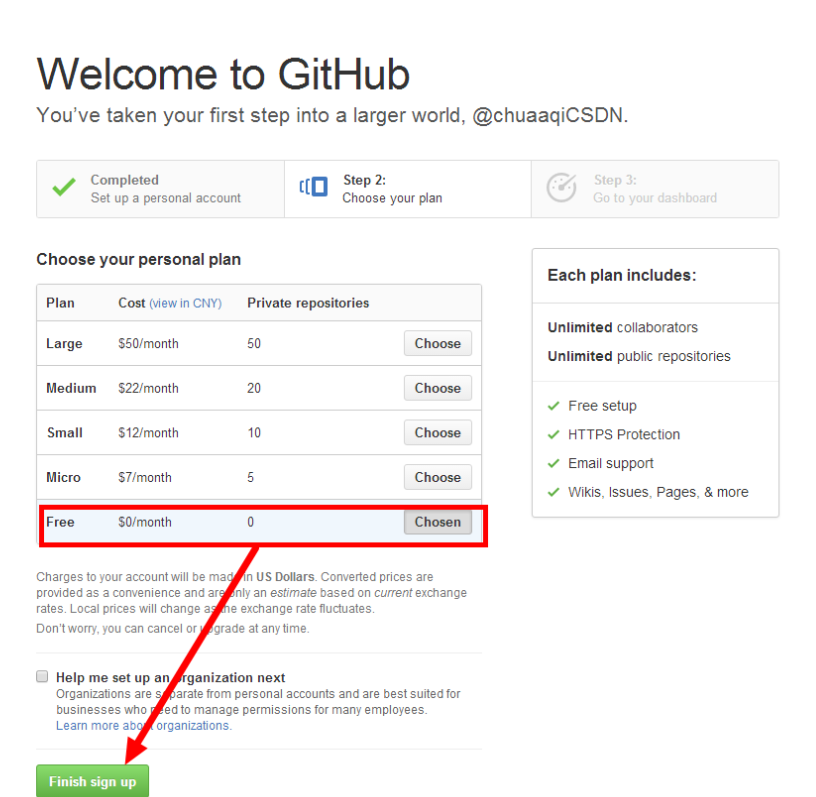
地址: https://github.com/

输入账号、邮箱、密码,然后点击注册按钮。



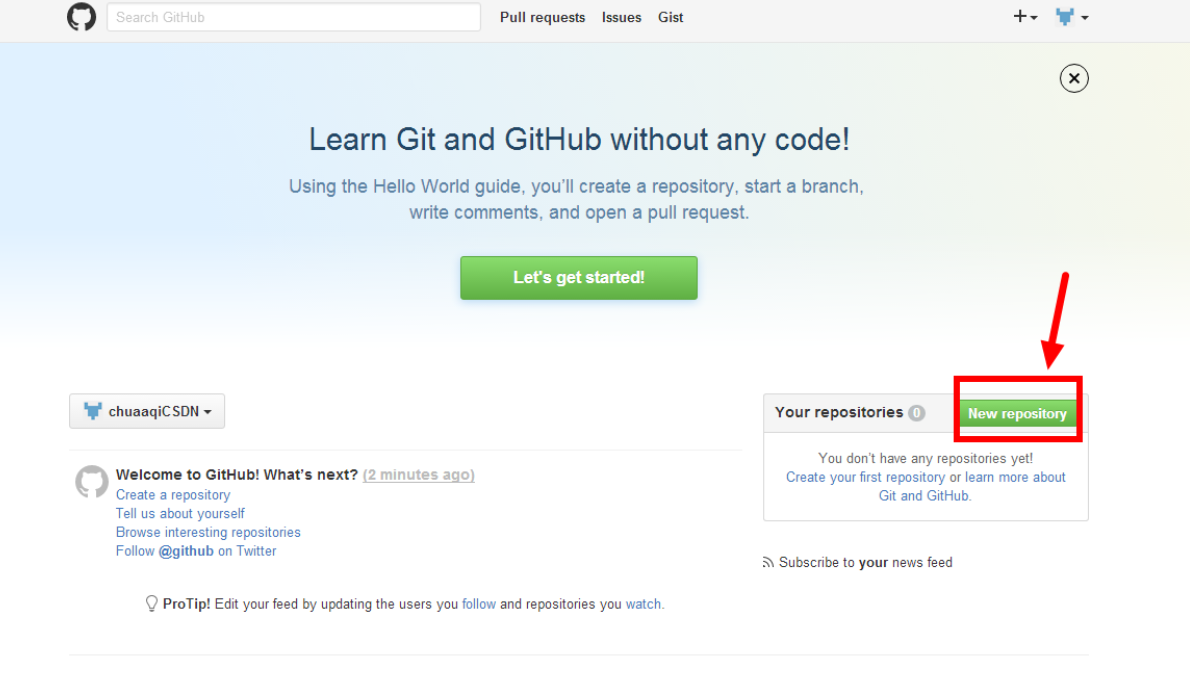
### 初始设置

注册完成后,选择Free免费账号完成设置。

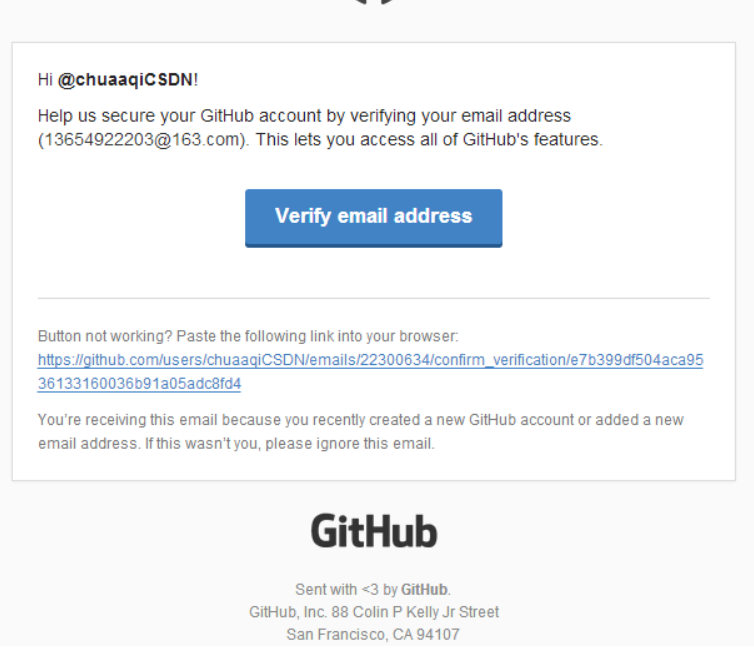


### 验证账号

新建一个仓库

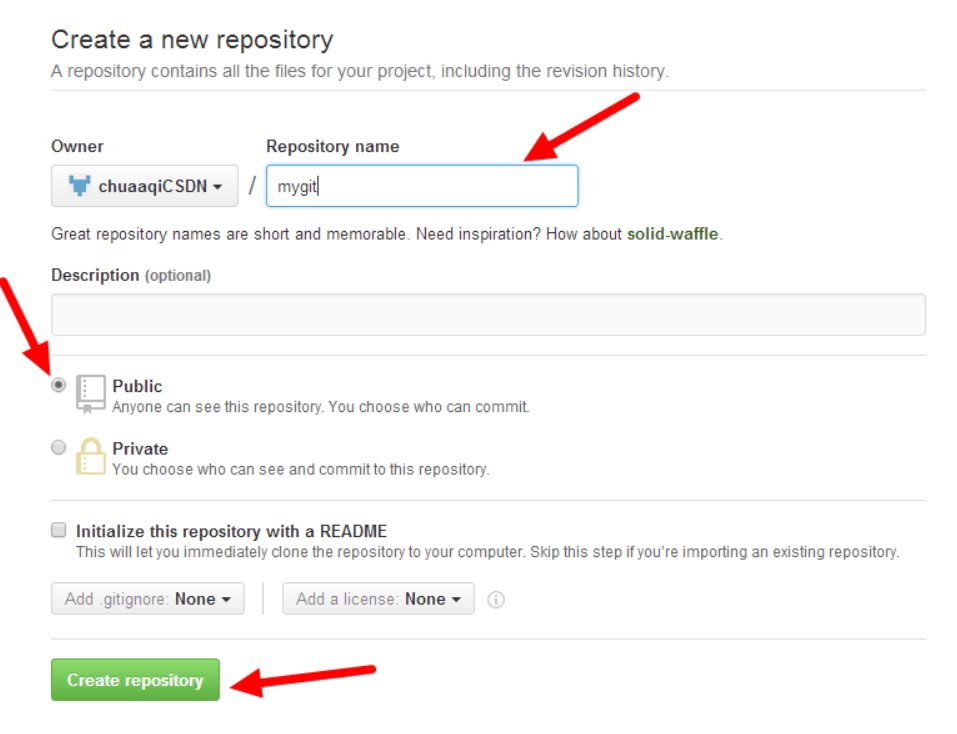


### 登录你注册时的邮箱，验证

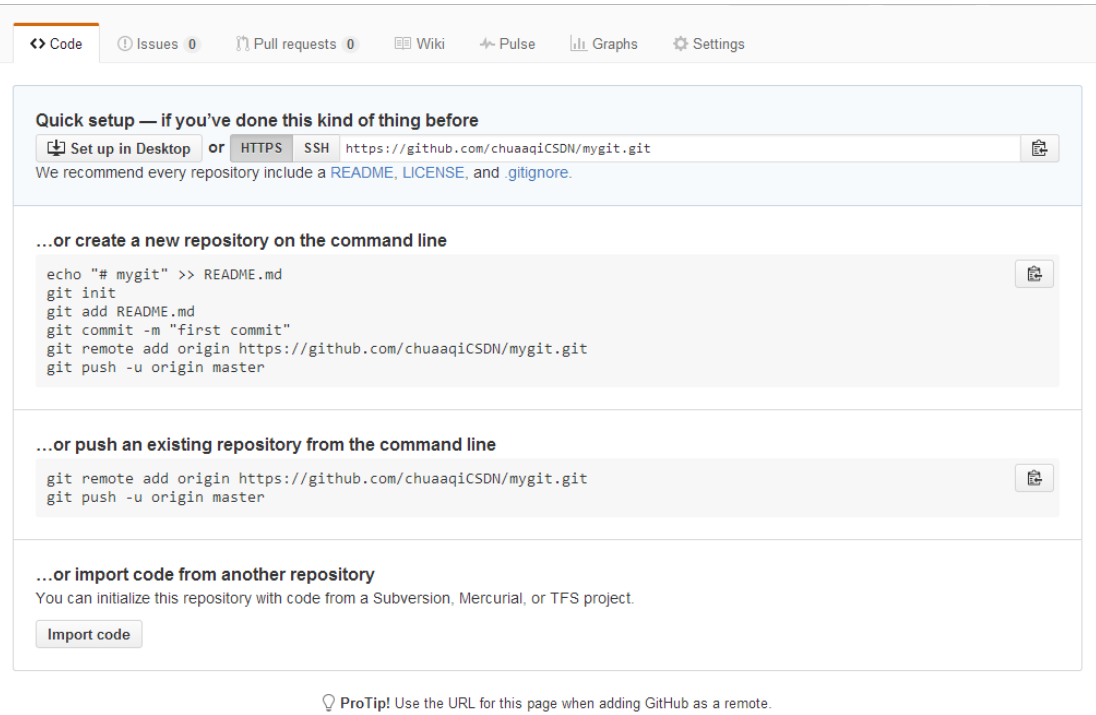


### 新建仓库

输入仓库名，点击创建



仓库创建成功



## 本地仓库上传到github仓库

在D:\git-project\spring-boot-admin-git路径下，有一个项目spring-boot-admin要传至github仓库中。

### 6.1. 文件保存至本地仓库

在D:\git-project\spring-boot-admin-git路径下，鼠标右击选择 Git Bash Here。





### 初始化版本库

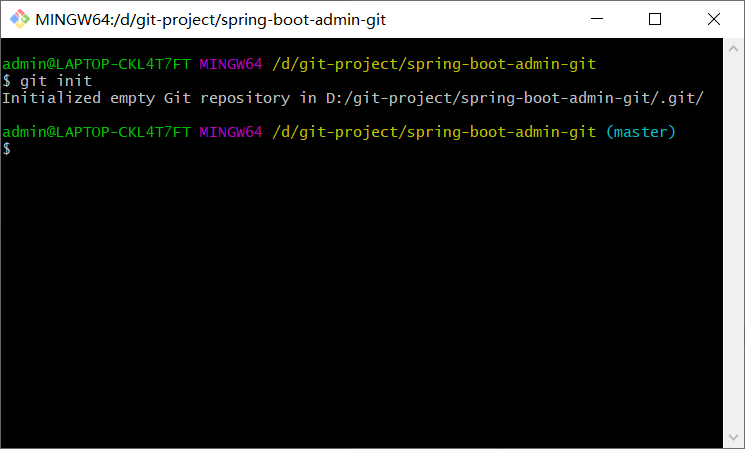
命令: git init

a.创建成功显示：

Initialized empty Git repository in D:/git-project/spring-boot-admin-git/.git/

b.注意：

此时在目标文件下会出现一个.git的文件夹，这是用来追踪版本库的，不要修改。



### 配置用户名和邮箱

a、设置用户名

git config -- global user.name '你在github上注册的用户名'

例如：git config --global user.name 'tanzuhao'

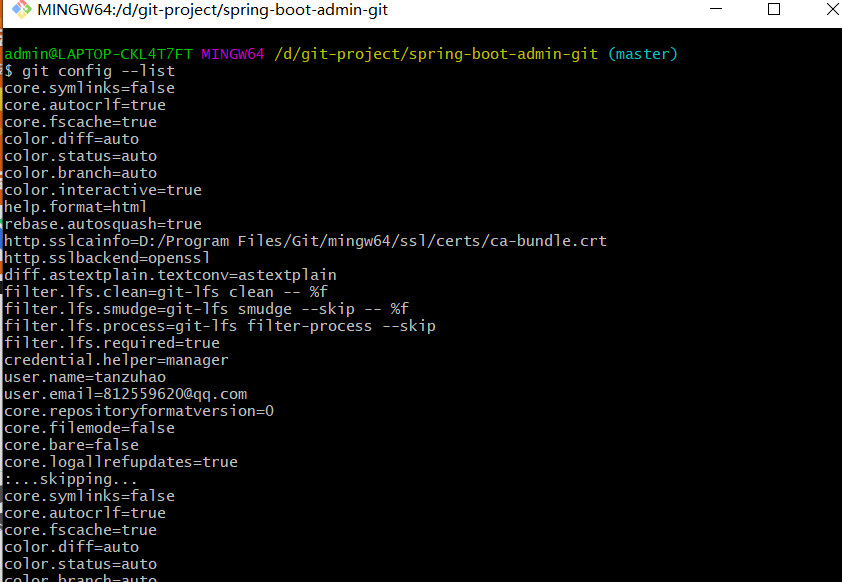
b、设置用户邮箱

git config -- global user.email '注册时候的邮箱'

例如: git config --global user.email '812559620@qq.com'

注意：该配置会在github主页上显示谁提交了该文件

c、查看配置是否成功



注意：git config --global 参数，有了这个参数表示你这台机器上所有的git仓库都会使用这个配置，当然你也可以对某个仓库指定不同的用户名和邮箱。

### 生产SSH密钥

GitHub选择的默认通信方式是SSH ,如果没有配置SSH key就只能clone代码不能push上去（只读），所以要先在Git里面生成SHH Key。如果在当前用户的第一级文件夹下有.ssh文件夹，说明以前可能使用过git，把该文件夹删除（或者先备份再删除，防止意外）。

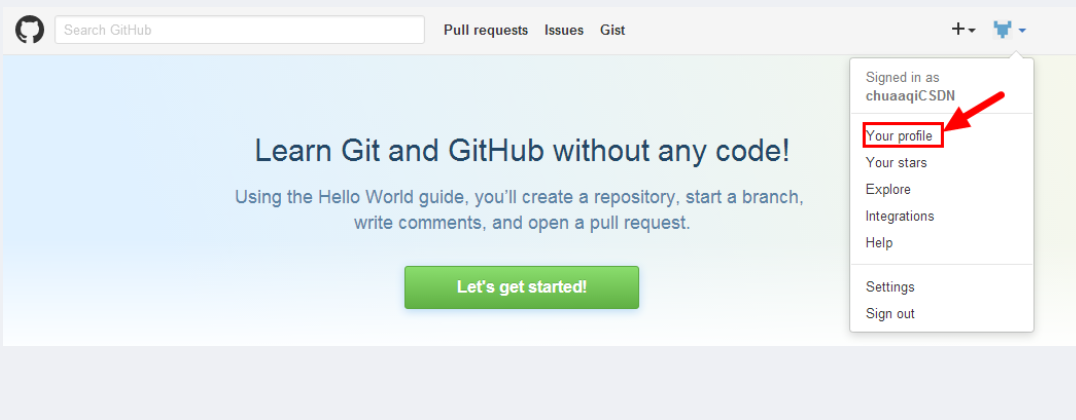
#### 填写email地址

ssh-keygen -t rsa -C "812559620@qq.com"

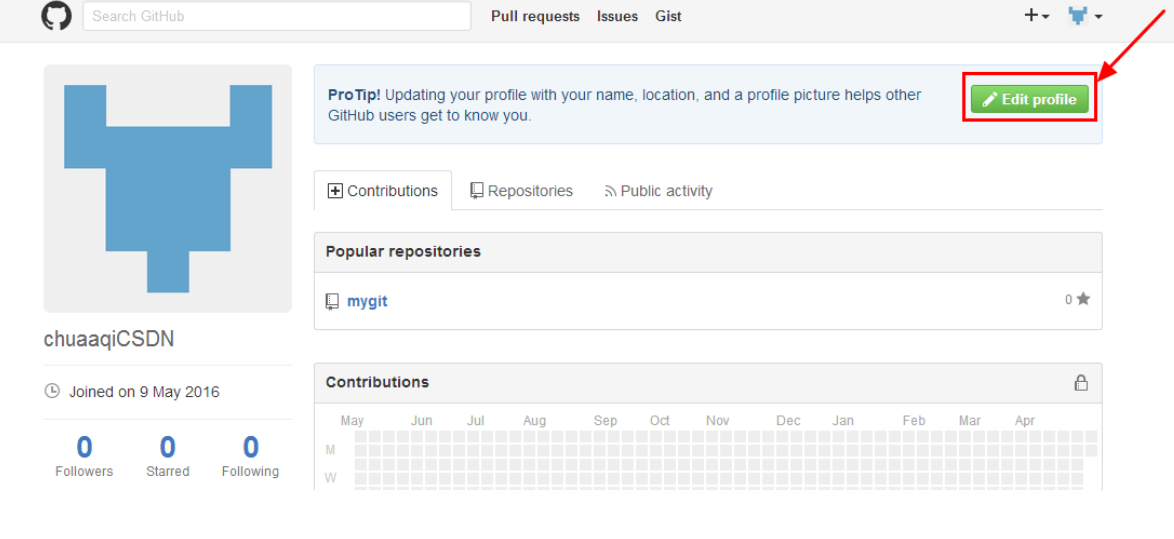
#### 按3个回车，密码为空。

#### Githud配置密钥

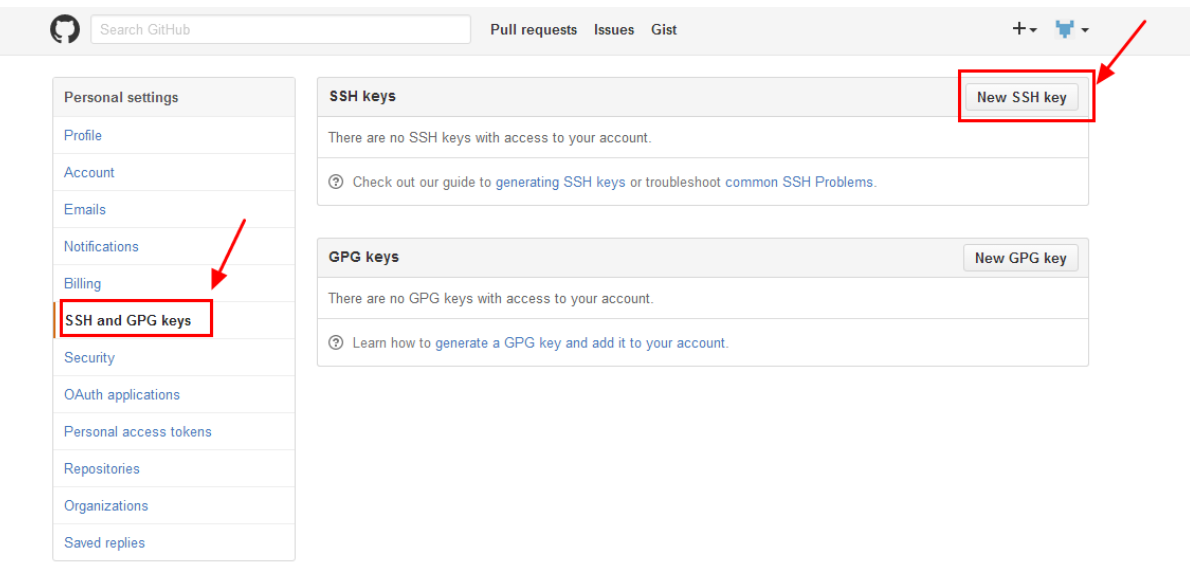
1. 登录你的gihub账号，点击Your profile

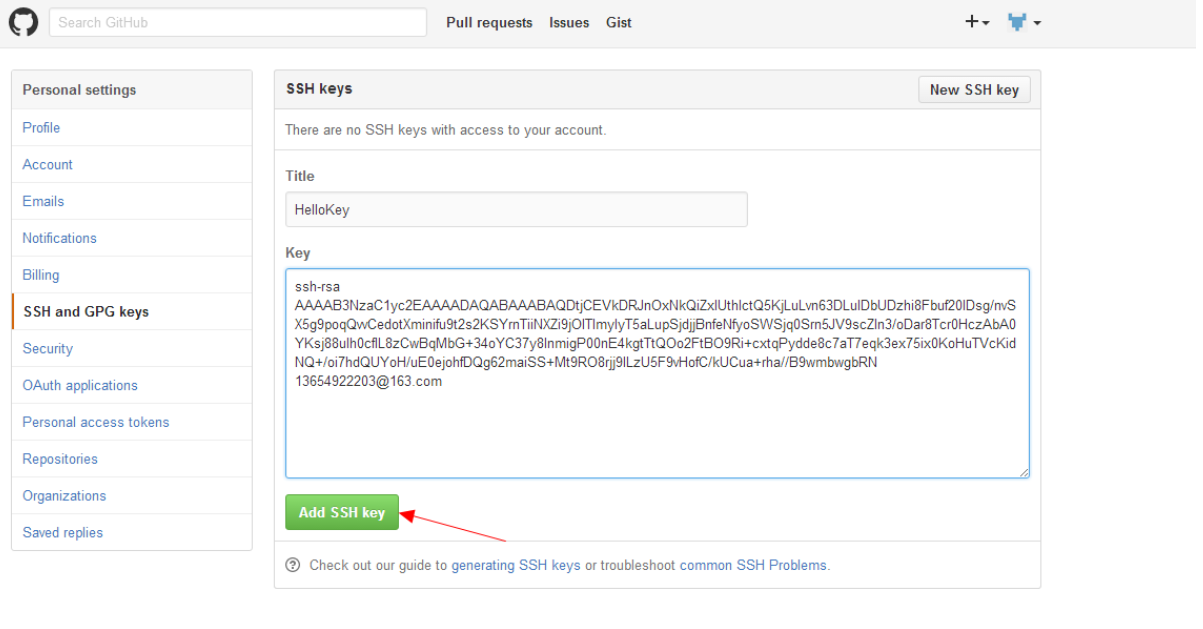


1. 然后点击Edit profile



1. 选择SSH并新建一个SSH Key





其中Title中的名称可以任意填写，将C:\Users\admin\.ssh\目录下id\_rsa.pub复制的所有内容粘贴到Key中，点击Add SSH Key，SSH密钥完成

#### 本地加载密钥

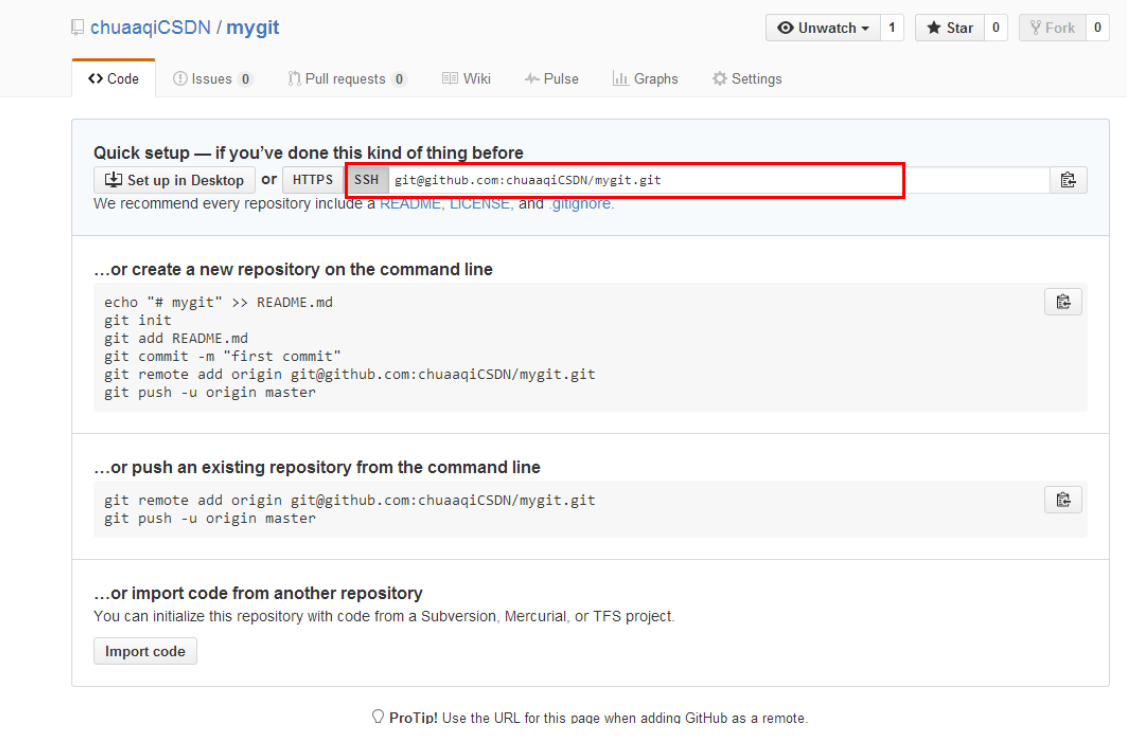
命令: eval "$(ssh-agent -s)"

命令： ssh-add ~/.ssh/id\_rsa

### 6.5.本地Git和Github进行关联

命令: git remote add origin https://github.com/tanzuhao/git-tan.git

其中红色部分的URL时是gihub中的SSH。



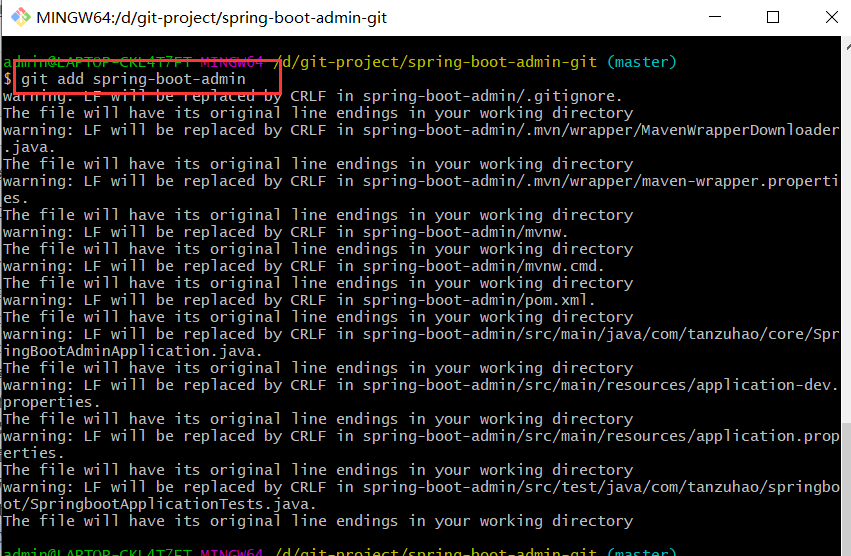
如果命令执行失败，执行:git remote rm origin,再执行

git remote add origin <https://github.com/tanzuhao/git-tan.git>

### 6.6.将文件添加到暂存区

git add filename

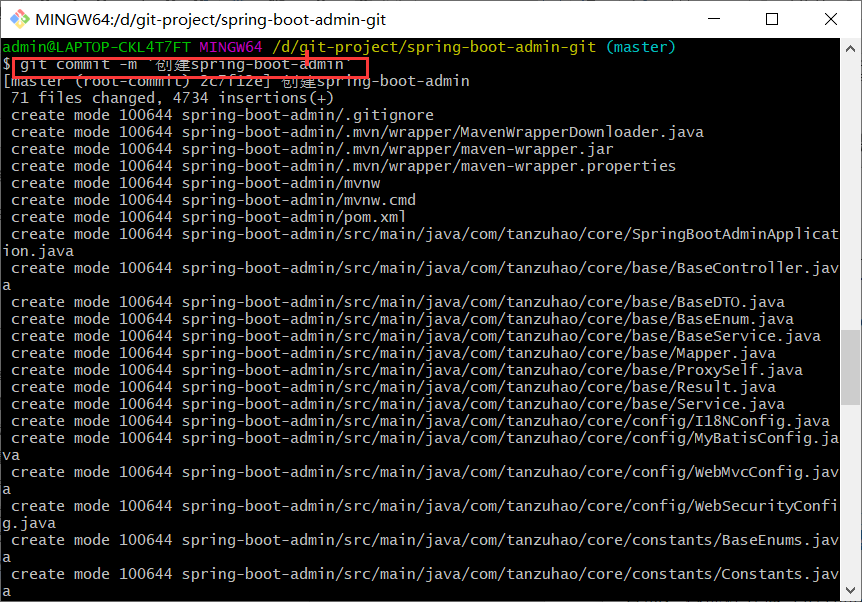
例如: git add spring-boot-admin



### 6.7. 将文件从暂存区提交到仓库

git commit -m '提交注释 '

例如: git commit -m ‘创建spring-boot-admin’



### 6.8.查看状态

git status

### 6.9. 由本地仓库push到远程仓库中

git push origin master

如果失败，显示出错信息：

To github.com:peTzxz/Property-management-system

! [rejected] master -> master (fetch first)

出现这个问题是因为github中的README.md文件不在本地代码目录中，可以通过如下命令进行代码合并。

解决：

1.执行： git pull --rebase origin master

2.再执行: git push origin master

最后可能需要输入Github注册的账号和密码

## Git克隆操作

目的：将远程仓库（github上对应的项目）复制到本地

命令: git clone 仓库地址

## git常用命令

cd xx:目录切换

mkdir XX：创建一个空目录 XX指目录名

pwd：显示当前目录的路径

cat xx：查看xx文件内容

touch: 创建新的文件

git init：吧当前的目录变成可以管理的git仓库，生成隐藏的.git文件

git add xx：把xx文件添加到暂存区

git commit -m “xx”：提交文件 -m后面的是注释

git status：查看仓库状态

git log：查看历史记录

git reset --hard HEAD^：网上回退一个版本

git reflog：查看历史记录的版本号id

git checkout -- xx：把xx文件在工作区的修改全部撤销

git rm xx：删除xx文件 之后要commit

git remote add origin https://github.com/qiuhaifeng01/a.git 关联一个远程库

git push -u（第一次要用-u以后不用）origin master：把当前master分支推送到远程库

git clone https://github.com/xxxxx   从远程库中克隆

git checkout -b dev：创建dev分支 并切换到dev分支上

git branch：查看当前所有的分支

git checkout master：切换回master分支

git merge dev：在当前分支合并dev分支

git branch -d dev：删除dev分支

git branch xxx：创建分支xxx

git remote：查看远程库信息

git remote -v查看远程库的详细信息

git pull origin master 将远程库的更新拉取到本地来

git push origin master：git会把master分支推送到远程库对应的分支上

## 对github上文件进行操作

### 9.1.提交文件到github

git add \*

git commit -m ‘本次更新说明’

git push origin master #将本地分支推到远程分支

### 删除github上文件

git pull origin master #将项目从远程库拉下来

git rm -r --cached xx.txt #删除指定文件，多个文件之间空格隔开

git commit -m '删除xx.txt'

git push -u origin master

### 9.3.更新github上文件到本地git

git pull origin master

### 同步代码（合并代码）

本地git有删除和新增文件，直接 按9.1.操作，git commit提示：

Your branch is up-to-date with 'origin/master'。

a.创建一个分支

git branch newbranch

b.检查分支是否创建成功

git branch

前面的\*代表的是当前你所在的工作分支

c.切换到你的新分支

git checkout newbranch

d.将改动提交到新分支上

git add \*

git commit -m'提交注释'

f.切换到主分支

git checkout master

g.将新分支提交的改动合并到主分支上

git merge newbranch

h.push代码到github

git push -u origin master

最后可删除创建的分支

git branch -D newbranch