## 配置git：

1、创建github账户：已有账号—sjtjpuzx

[邮箱—495063950@qq.com](mailto:邮箱—495063950@qq.com)

密码—SJ\*\*\*\*\*\*\*\*\*26.

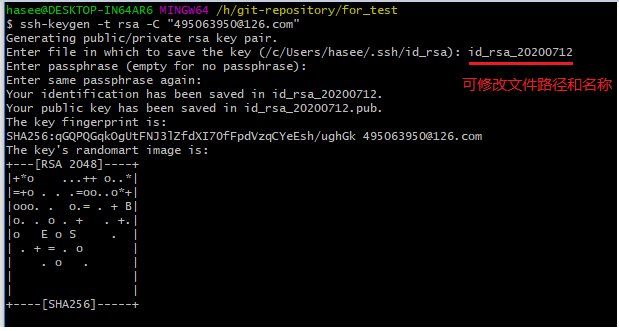
可以选择网页登录（登录地址址：https://github.com/account/unverified-email）或者客户端登录（下载地址：http://windows.github.com）；

2、在本地安装git工具；

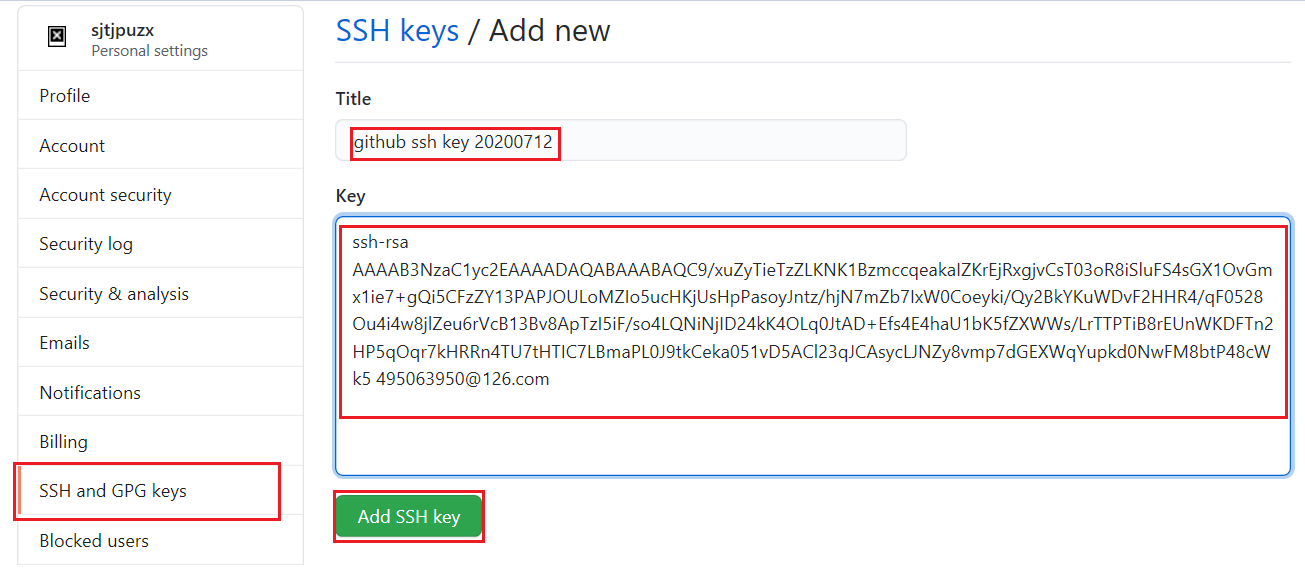
3、配置git，创建ssh key；

a、在本地任意目录打开git bash(必须安装git工具才会有)；

b、生成公钥SSH key；输入ssh-keygen -t rsa -C "495063950@126.com"，邮箱是在github上注册的邮箱。默认会在C:\Users\hasee\(每个电脑有差异，可以在提示中看到)目录下生成.ssh文件夹。如果成功的话，在.ssh文件夹里面包含id\_rsa和id\_rsa.pub两个文件，其中id\_rsa.pub是公钥；只有在github上正确添加SSH key，才能在github上下载和上传仓库。



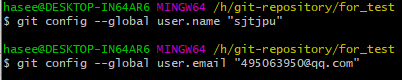
c、将公钥添加到github；打开id\_rsa.pub，复制里面的所有内容，回到github上，进入Account🡪Settings，左边选择SSH Keys，New SSH key 🡪 Add SSH Key，title随便填，粘贴在电脑上生成的key；



d、验证是否添加成功；在git bash下输入ssh -T git@github.com，如果是第一次的会提示是否continue，输入yes就会看到：You've successfully authenticated, but GitHub does not provide shell access 。这就表示已成功连上github。(第一次连接会报失败，是因为没有known\_hosts文件，输入yes后系统会自动创建一个known\_hosts文件，第二次连接后，即可成功)。



e、配置username和email；因为每次commit要记录它们，用来区分是谁操作的。

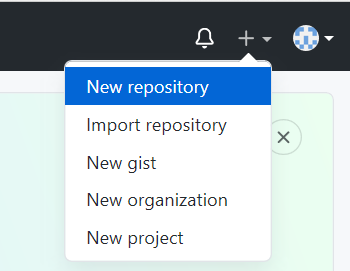


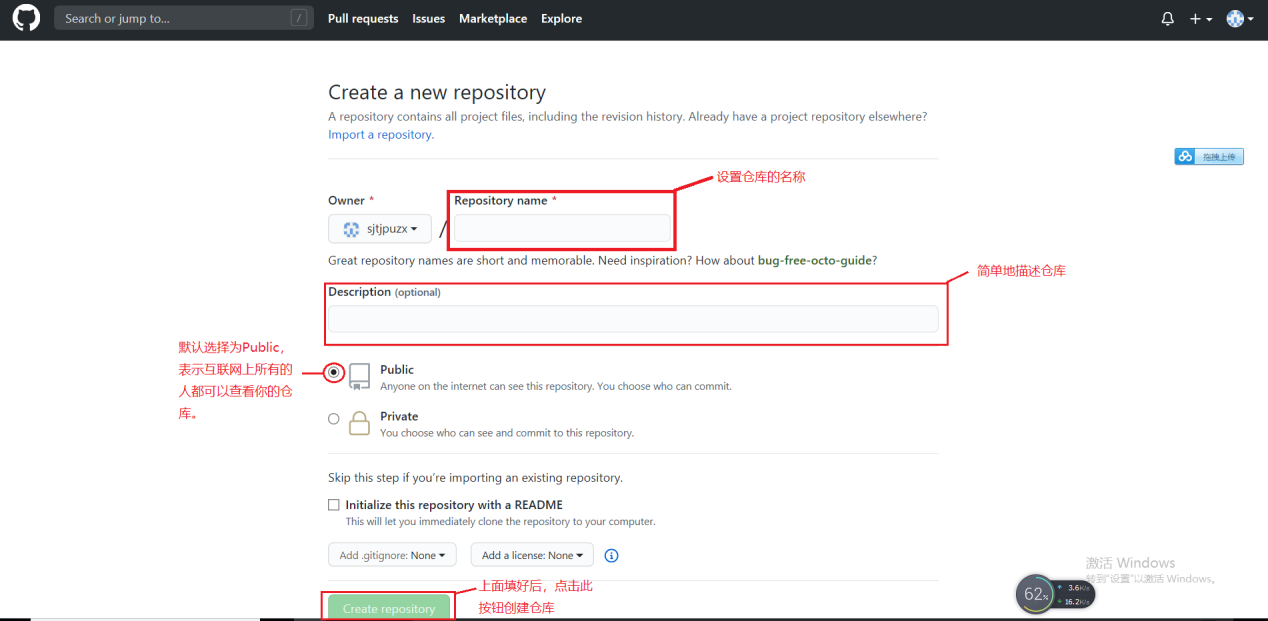
## github创建仓库：

一个仓库通常用来组织一个单独的项目。该仓库可以包含你项目中的所有文件，如文件夹、文件、图片、视频、电子表格、数据集等。我们建议包含一个README文件，或者你自定义的用来记录项目信息的文件。

1、登录<https://github.com>；

2、在页面的最右上角，你的头像的左边，点击“＋”，在弹出的下拉选项中选择“New repository”；





**注意**：如果需要将本地仓库上传到github，则不要勾选“Initialize this repository with a README”；如果是在github上创建仓库，本地来clone远程仓库，则可以勾选“Initialize this repository with a README”。

## 创建本地仓库：

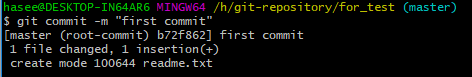
1、进入到要创建仓库的本地目录；

2、git init；这样会在本地生成一个空的git仓库，可以往仓库中添加代码或者文件；



3、git add .；这个是将项目上所有的文件添加到仓库中的意思，如果想添加某个特定的文件，只需把.换成这个特定的文件名即可。

4、git commit -m "first commit"；输入git commit -m "first commit"，表示你对这次提交的注释，双引号里面的内容可以根据个人的需要改。



## 上传本地仓库到github

1、首先需要创建本地仓库；

2、关联远程库：git remote add origin(可修改) branch\_Name(为空时默认为master) url；url 为你的远程仓库的 url，可采用 http 协议或 ssh（git）协议，一般采用ssh协议；



**说明**：

git remote add [shortname] [url]

在添加远程仓库中，可以设置**简单的名字**，方便以后引用这个原创仓库；在github创建仓库时，默认的shortname就是origin。

比如，将本地仓库的master分支推送到远程仓库；

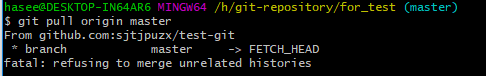
git push origin master

综上，你的远程仓库的名字是demo1，origin是远程仓库的简写或者引用，在本地使用origin就代表操作的是远程仓库demo1。

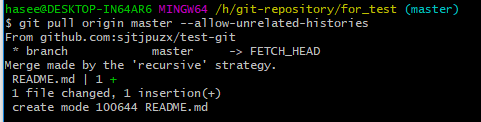
3、从远程仓库拉取所有更新（每次上传项目都要操作）；

第一次使用git pull需要指定远程库的名称和分支，之后可以直接使用git pull从远程仓库拉取更新：

git pull <remote> <branch>



有可能出现如上图所示的错误，原因是两个分支是两个不同的版本，具有不同的提交历史，添加一句：--allow-unrelated-histories，可以允许不相关历史提，强制合并。



4、提交本地仓库的修改至github远程仓库：

git push -u origin master

第一次需要指定远程仓库名称和分支，之后可以直接使用git push提交修改。

