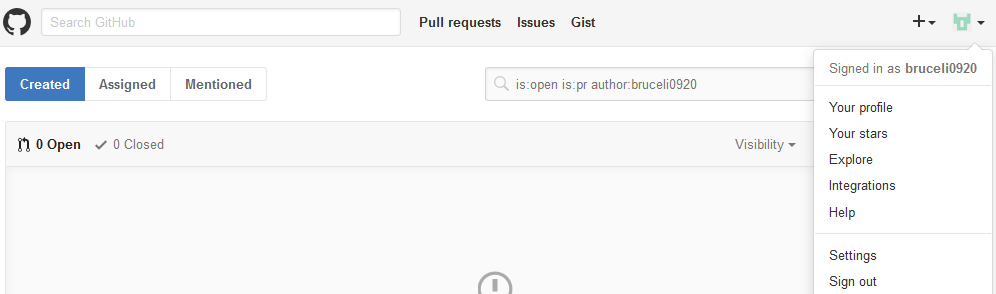
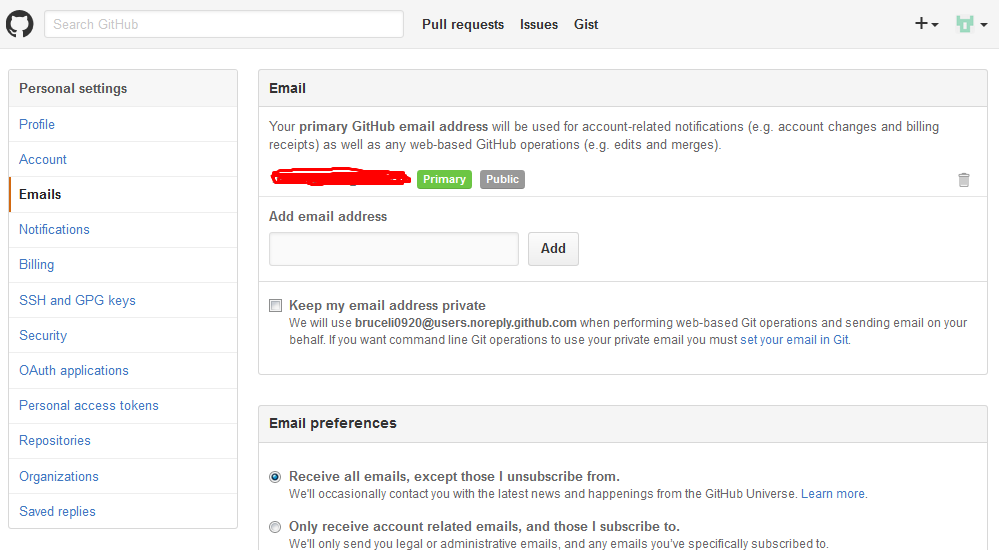
# Git的安装和使用

## 第一步:注册用户名和密码，邮箱；

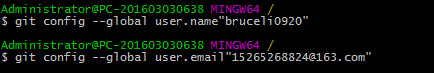


## 第二步：通过邮箱打开链接，表示邮箱已经激活。可以进行配置Git了



## 第三步：配置Git

### 首先输入用户名和邮箱。注意空格

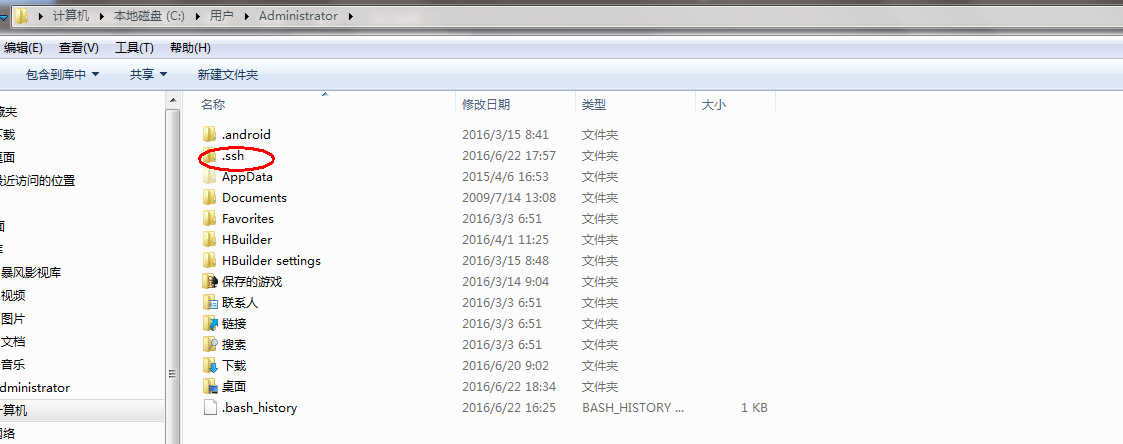


## 建立个人库和git库之间的信任（相当于本地到服务器之间的密码）

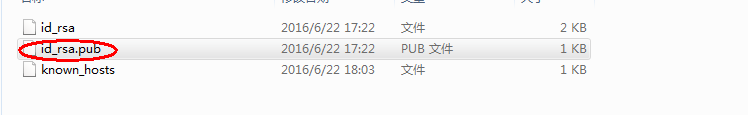


## \*可以设置密码，也可以不设置密码

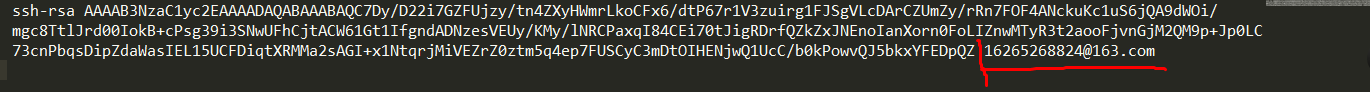
## 通过保存的路径去找到文件



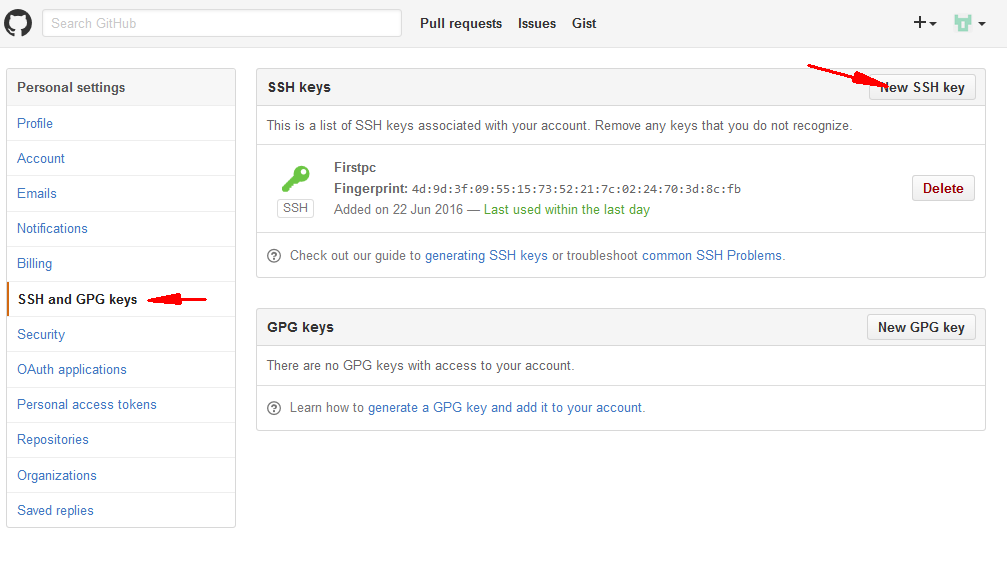
## 将ssh打开后找到id\_rsa.pub文件，用sublime或其他编辑器软件打开



## 然后将打开的文件复制下来，除了邮箱(不除也行)



## 4：在自己的git主页上设置



## 打开SSH and GPG,点击New SSH key,添加一个标题，然后将复制的密匙粘贴到里面刷新就会发现钥匙图案变绿了



## 测试之后钥匙就会变绿

# Git个人仓库和git远程仓库的使用

## 建立自己的个人仓库。提交：

Cmd --》输入指令代码

1. D: 切换到D盘

2.在D盘下输入 mkdir ceshi 之后就会在D盘目录下生成一个ceshi的文件夹

3.dir 显示D盘下的所有东西，加<>的就是文件夹

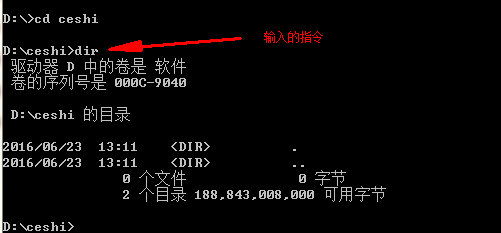
4.dir/w 把文件加出的名字都省略

dir/v 把以v字母开头的文件夹都找出来

dir/a 把隐藏的东西都显示出来

1. cd ceshi cd是指到指定文件夹下，也就是说现在这是在ceshi这个文件夹的下面

6.显示ceshi文件夹下只有2个文件



1. 在ceshi文件夹下输入git指令

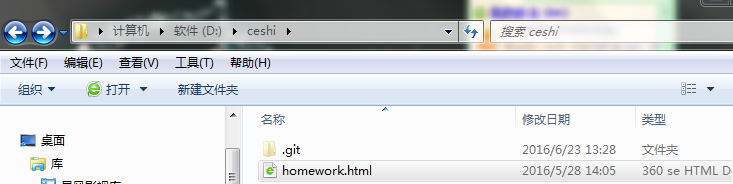
git init



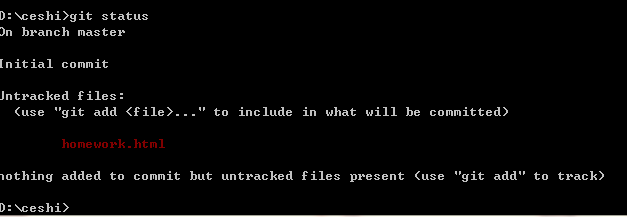
1. 再输入git/a查看隐藏的文件，就会找到一个。git的文件夹



9,之后在将自己要写的代码放到测试文件夹下

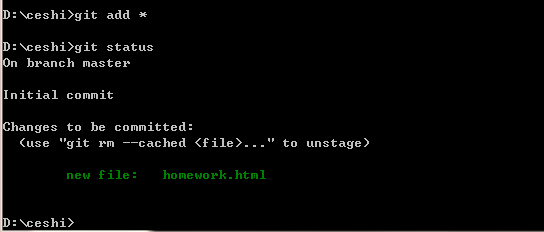


1. 查看一下你现在仓库的状态 git status



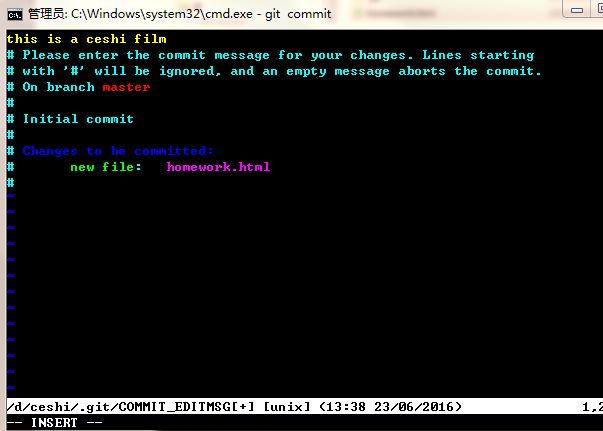
1. 红色的文件就是你放进仓库的文件，但是现在并没有提交

通过git add \* 将文件加入到仓库的缓冲区，然后在查看仓库状态



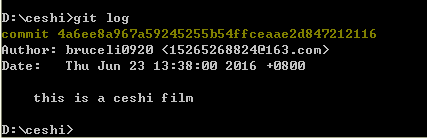
1. 最后一步，将缓冲区的文件夹提交到个人仓库，用 git commit 指令

有可能遇到这种情况：（也可以通过git commit -m”描述的内容” 直接操作，跨过这一步）



这是对提交的文件进行描述；你可以先按i进行编辑描述，之后退出；按ESC键，再按冒号和qw就可以

之后就可以通过git log查看你的提交日志了



详解：

检查本机是否有ssh key设置

$ cd ~/.ssh 或cd .ssh

如果没有则提示： No such file or directory

如果有则进入~/.ssh路径下（ls查看当前路径文件，rm \* 删除所有文件）



2

1、使用Git Bash生成新的ssh key。

$ cd ~  #保证当前路径在”~”下

$ ssh-keygen -t rsa -C "xxxxxx@yy.com"  #建议填写自己真实有效的邮箱地址

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/xxxx\_000/.ssh/id\_rsa):   #不填直接回车

Enter passphrase (empty for no passphrase):   #输入密码（可以为空）

Enter same passphrase again:   #再次确认密码（可以为空）

Your identification has been saved in /c/Users/xxxx\_000/.ssh/id\_rsa.   #生成的密钥

Your public key has been saved in /c/Users/xxxx\_000/.ssh/id\_rsa.pub.  #生成的公钥

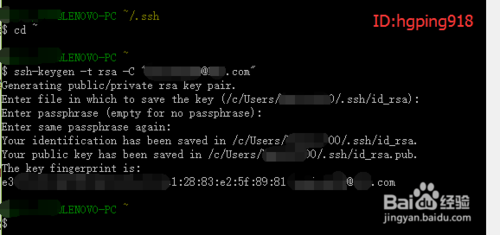
The key fingerprint is:

e3:51:33:xx:xx:xx:xx:xxx:61:28:83:e2:81 xxxxxx@yy.com

\*本机已完成ssh key设置，其存放路径为：c:/Users/xxxx\_000/.ssh/下。

注释：可生成ssh key自定义名称的密钥，默认id\_rsa。

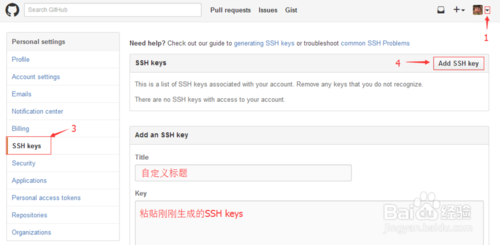
$ ssh-keygen -t rsa -C "邮箱地址" -f ~/.ssh/githug\_blog\_keys #生成ssh key的名称为githug\_blog\_keys，慎用容易出现其它异常。



3

添加ssh key到GItHub

3.1 登录GitHub系统；点击右上角账号头像的“▼”→Settings→SSH kyes→Add SSH key。

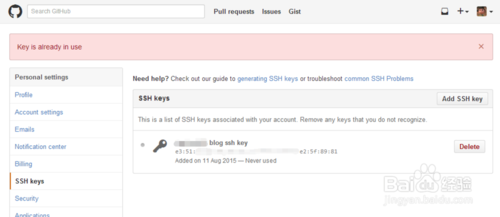


4

3.2 复制id\_rsa.pub的公钥内容。

1) 进入c:/Users/xxxx\_000/.ssh/目录下，打开id\_rsa.pub文件，全选复制公钥内容。

2) Title自定义，将公钥粘贴到GitHub中Add an SSH key的key输入框，最后“Add Key”。

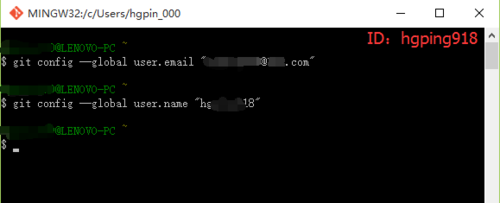


5

配置账户

$ git config --global user.name “your\_username”  #设置用户名

$ git config --global user.email “your\_registered\_github\_Email”  #设置邮箱地址(建议用注册giuhub的邮箱)



6

测试ssh keys是否设置成功。

$ ssh -T git@github.com

The authenticity of host 'github.com (192.30.252.129)' can't be established.

RSA key fingerprint is 16:27:xx:xx:xx:xx:xx:4d:eb:df:a6:48.

Are you sure you want to continue connecting (yes/no)? yes #确认你是否继续联系，输入yes

Warning: Permanently added 'github.com,192.30.252.129' (RSA) to the list of known hosts.

Enter passphrase for key '/c/Users/xxxx\_000/.ssh/id\_rsa':  #生成ssh kye是密码为空则无此项，若设置有密码则有此项且，输入生成ssh key时设置的密码即可。

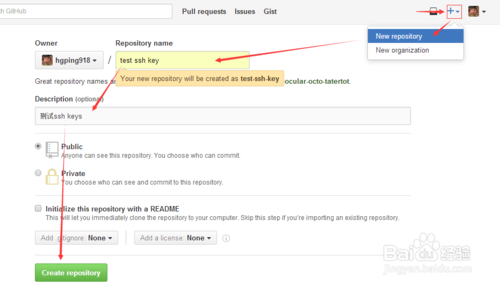
**Hi xxx! You've successfully authenticated, but GitHub does not provide shell access. #出现词句话，说明设置成功。**



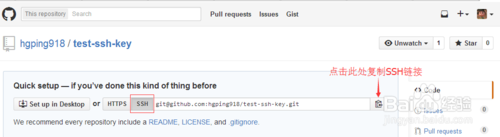
END

## 将本地项目通过SSH push到GitHub

在github上创建一个示例仓库，如：test ssh key。



复制test ssh key的ssh路径。



本地创建项目

1) 创建目录

$ mkdir test

$ cd test

2) 初始化

$ git init

3) 创建hello.md文件

$ echo "这是一次测试test ssh key" > hello.md

4) 提交到本地

若出现如上warning提示则重新提交一次即可。

$ git add .   #提交当前目录下所以文件

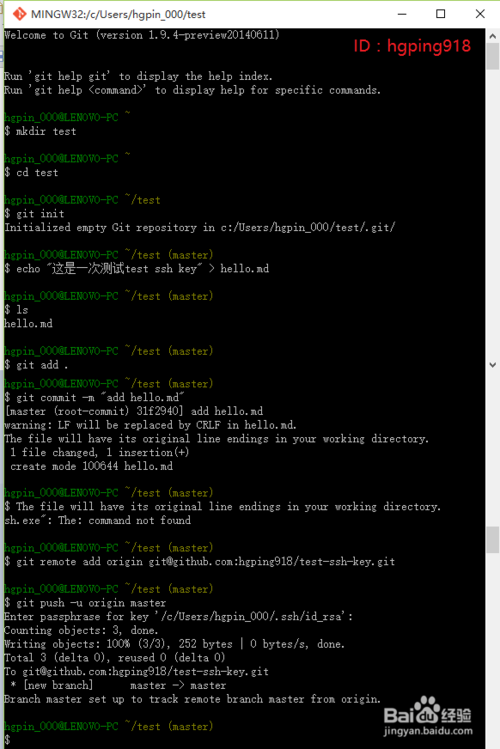
$ git commit -m "add hello.md"   #提交记录说明

5) 提交到github

$ git remote add origin ‘粘贴复制test ssh key的ssh路径’  #

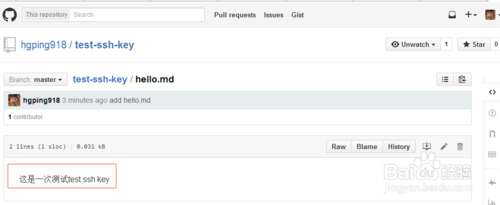
$ git push -u origin master

Enter passphrase for key '/c/Users/hgpin\_000/.ssh/id\_rsa':  #ssh key设置密码故此需要输入密码



4

刷新test ssh key仓库，查看hello.md。（完）



建立关联出现情况：

1、先删除远程 Git 仓库

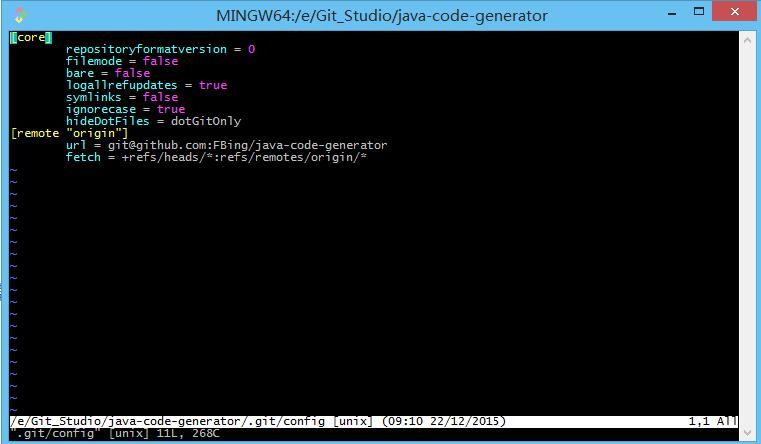
$ git remote rm origin

2、再添加远程 Git 仓库

$ git remote add origin git@github.com:FBing/**[Java](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "http://blog.csdn.net/top_code/article/details/_blank)**-code-generator

如果执行 git remote rm origin 报错的话，我们可以手动修改gitconfig文件的内容

$ vi .git/config



把 [remote “origin”] 那一行删掉就好了。