 以前不知道使用代码管理工具，最后写的一些东西都没有了，由于硬盘坏了或者不小心格式化了之类的，后来使用了Git

和Github来托管自己的代码和读书笔记方便了不少，到哪里只要有网就可以把自己的东西拷贝下来继续使用。

我这里简单的记录一下我使用的过程，最简单的使用都是，高级的功能我一直没有使用到，虽然买一本《Git权威指南》

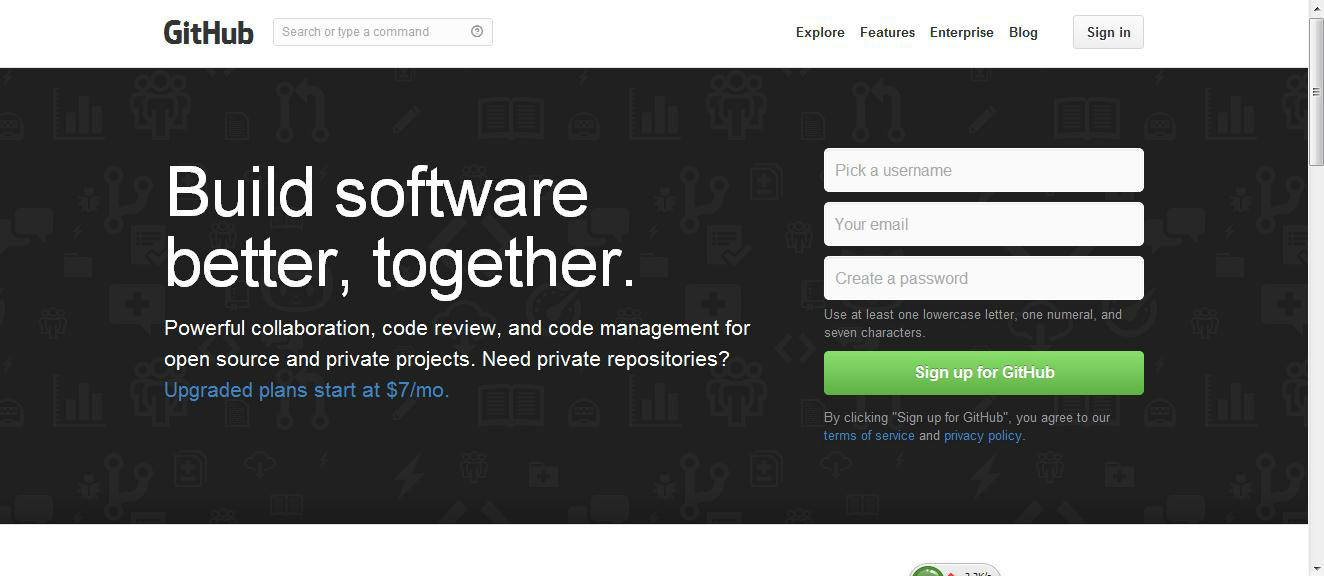
但是很多东西用不到就不能够真的会。下面开始简单介绍我使用的方法，我这个是在windows上使用的。我使用分两种情况，

因为我的代码都是在Linux下写的，所以在linux下主要是托管代码用，在windows下主要是托管笔记使用的，比如一些PDF

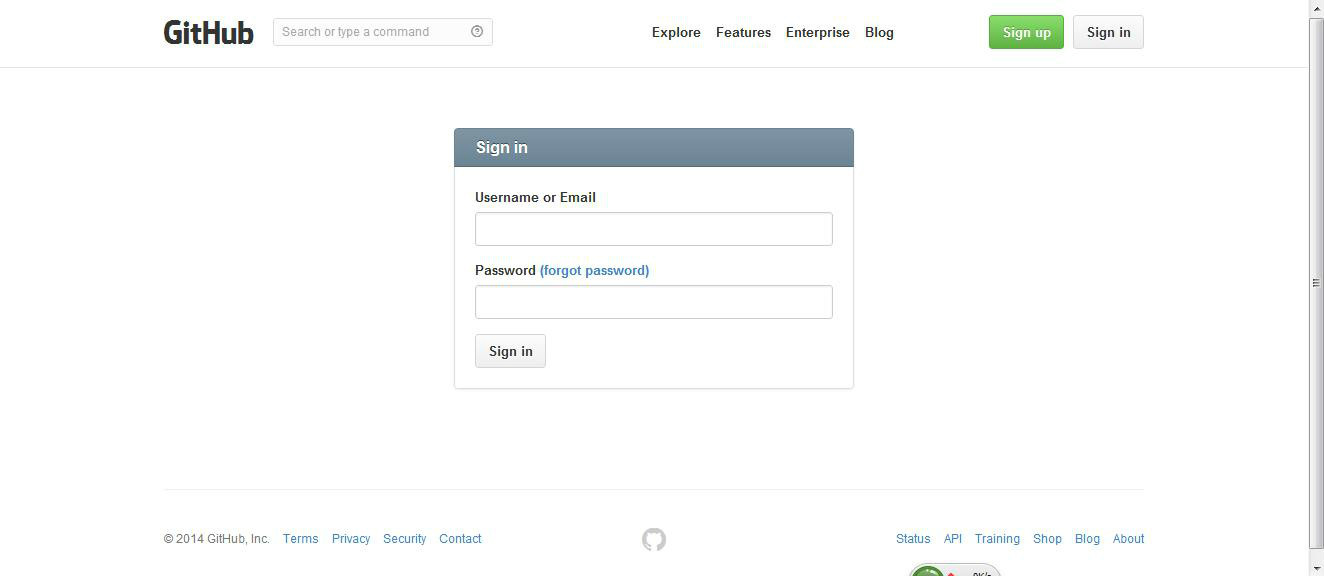
文档，我在看的时候会加入自己的注释，这样使用托管功能，在哪里都可以接着注释，不用总是拷贝或者总是需要复制。

先注册github.com的账号官方网站: [https://github.com/](https://github.com/" \t "https://my.oschina.net/bxxfighting/blog/_blank)

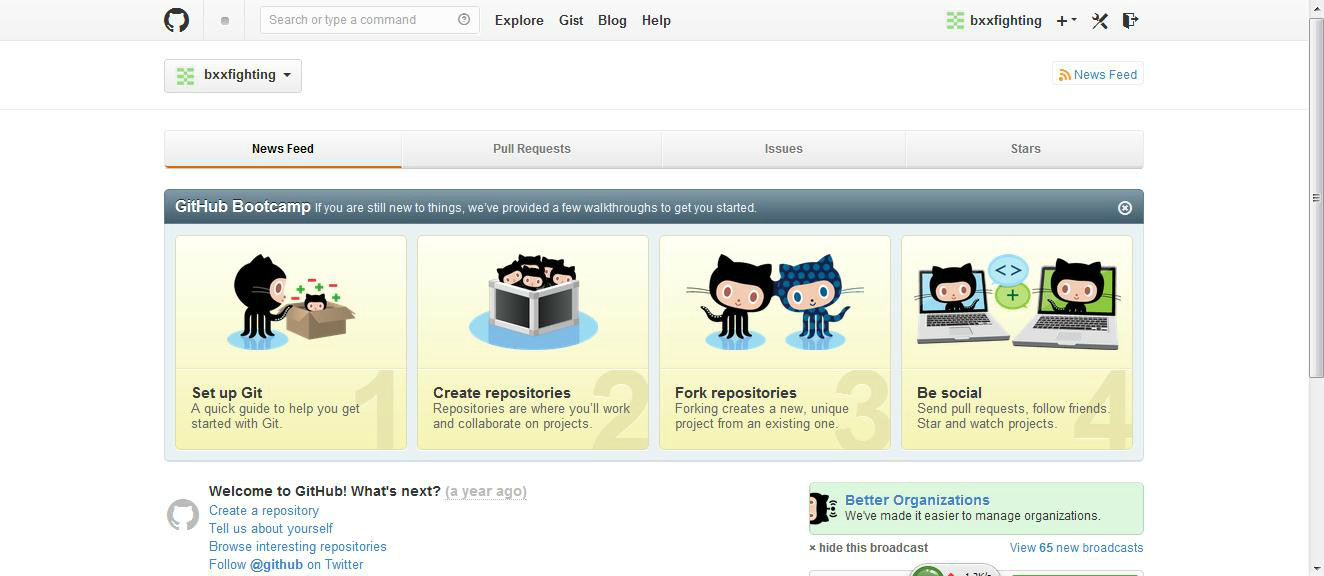
**注册界面，第一个用户名，以后会用到，我的是bxxfighting**

****

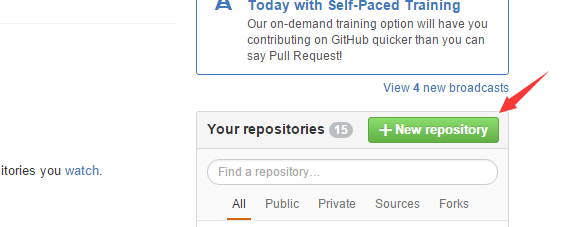
**2.      登录界面**

****

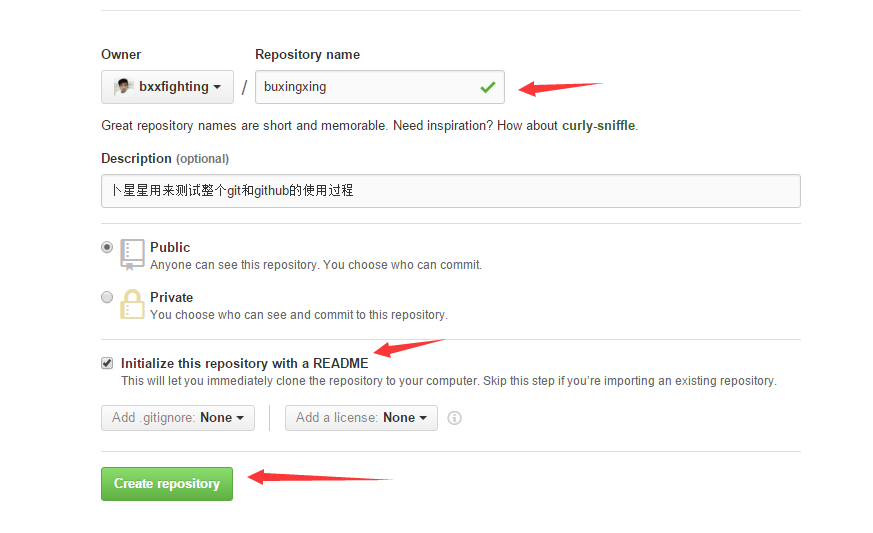
**3.      登录成功后界面**

****

4.**创建仓库,我现在创建一个测试仓库叫buxingxing,仓库分公开的和私有的,公开的是免费的,私有的是收费的,我使用的是公开的仓库,如下创建方式**

****

点击New repository按钮,弹出如下界面,第一行填仓库名,就叫个buxingxing,第二行是对这个仓库的描述,之后那个Public就是公共仓库的意思,接下来的README就是在仓库里创建一个README文件,可以往里写一些介绍你这个项目的功能之类的东西,再下面那个Add gitignore按钮,可以选择你这个项目是用什么语言之类的,后面那个License我没有选,点击”Create repository”



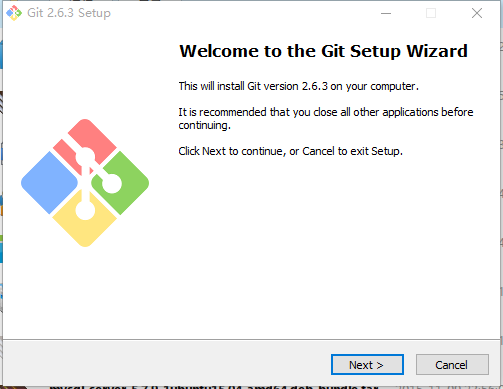
**5.      创建仓库成功后,界面如下显示,可以点击README.md来编译这个文件**

****

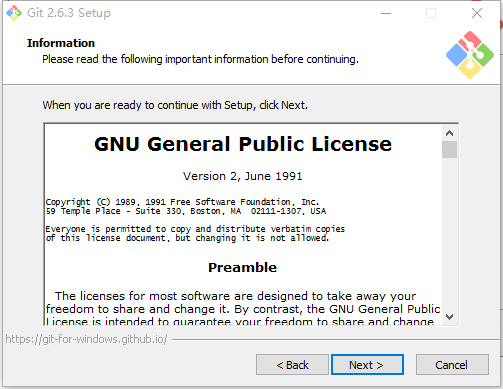
# 在windows下安装Git

**1.      下载网址: http://git-scm.com/download/**

**2.      下载完毕,打开安装,点击”next”**

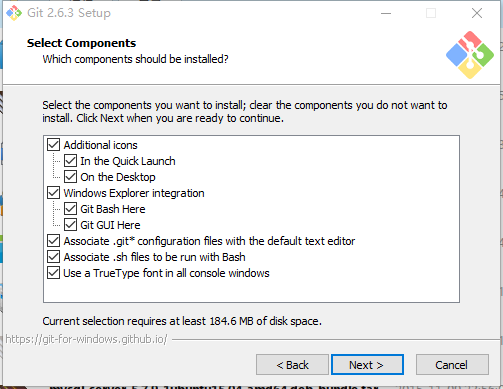


**3.      点击”next”**

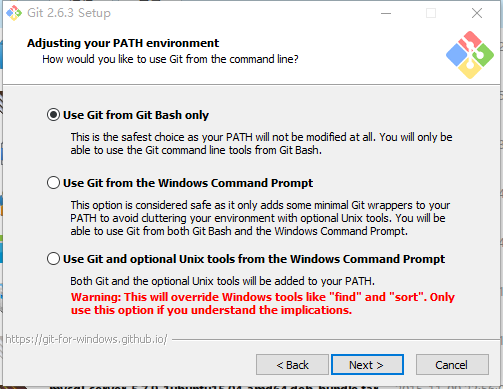
****

**4.      可以更改安装路径后,点击”next”,我没有更改，就是使用的默认路径。**

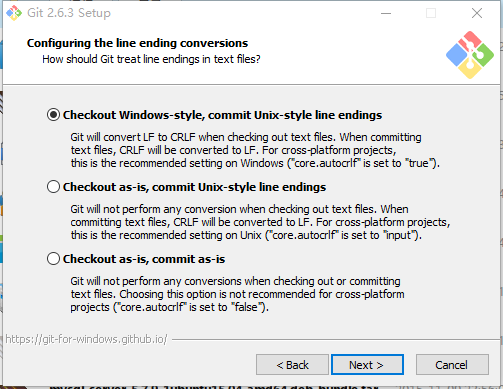
**5.      这里我把所有的选项都选中了。**

****

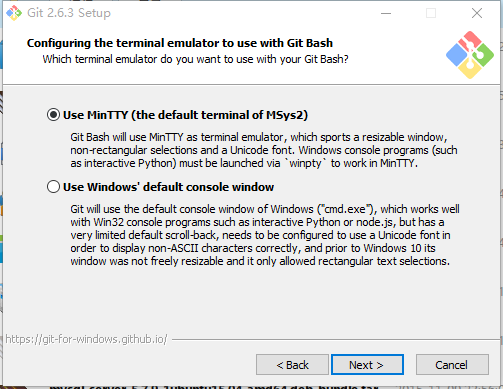
**6.      直接默认，点next.**

****

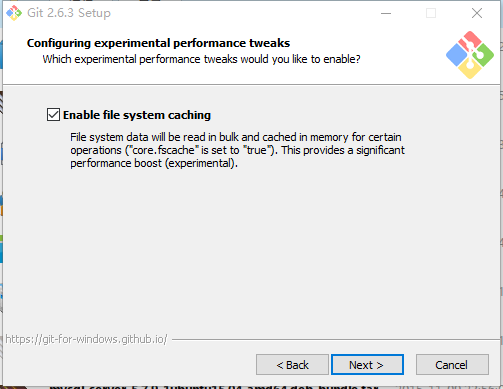
**7.      默认设置，点Next**

****

**8.      默认设置，点击next。**

****

**9.     默认设置，点击”next”**

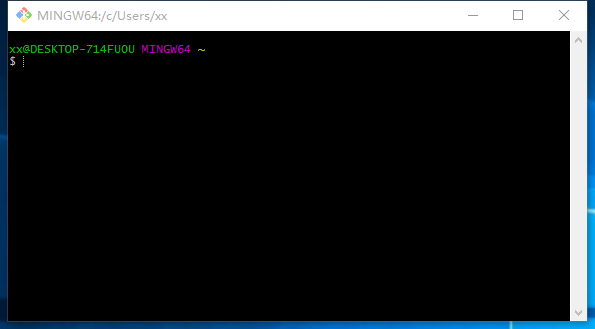
****

# 使用git和github托管项目代码

**1.      双击图标”Git Bash”**

****

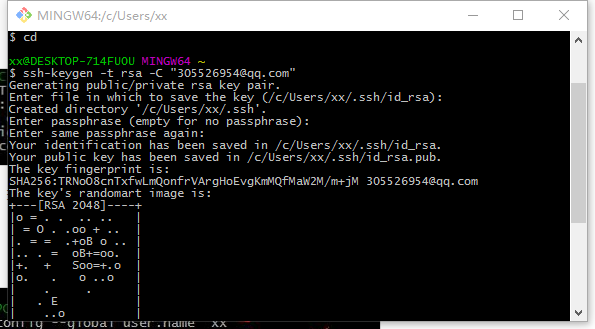
**2.      打开界面如下**



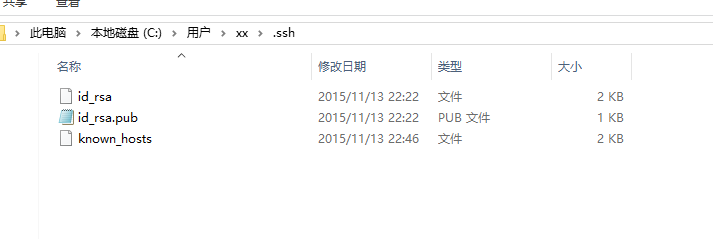
**3.      配置Git,图示如下:**

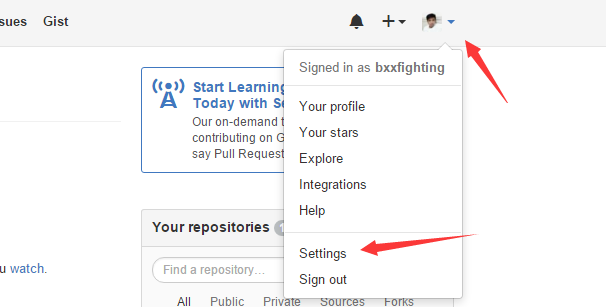
a)        先输入ssh-keygen –t rsa –C “邮箱地址”,注意ssh-keygen之间是没有空格的,其他的之间是有空格的

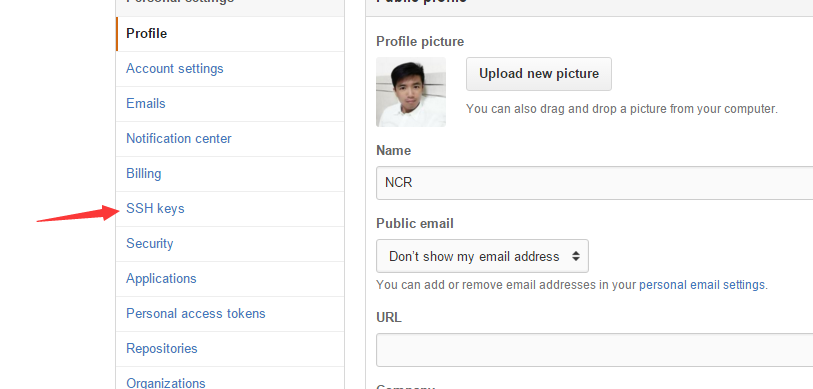
b)        回车之后,会出现一行,让你输入一个保存密钥的地方,括号里面是它默认的位置，这里会让你输入几次内容，都不用输入，直接回车就可以了，可以看到如图的效果（这里最好都不要输入，直接回车，我第一次更改了保存的路径，就出了问题，少了一个文件）：

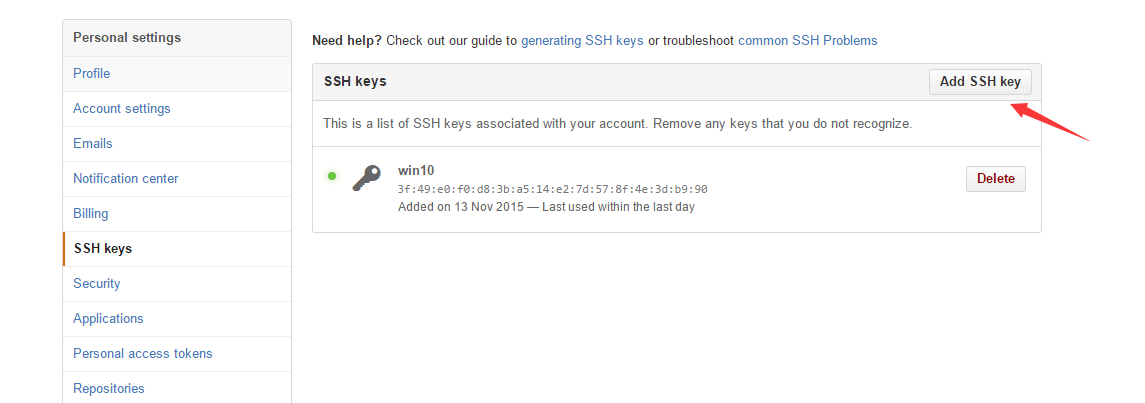


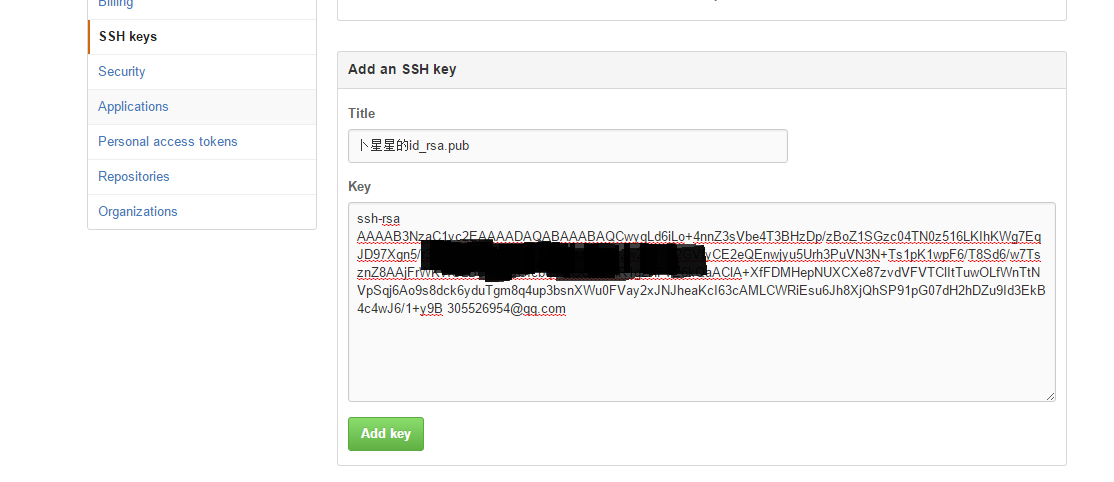
c)        回车之后,这样密钥就生成了,可以打开id\_rsa.pub（位置根据你的电脑来看）来查看,我使用的是记事本直接打开的这个文件,里面的所有内容就是这个密钥,一会需要使用的时候,就直接全选复制就可以了

**d)        现在转到github网站上去配置一下ssh key,点击箭头指示的三角图标，选择Settings，然后点击左侧的SSH Keys，之后点击右侧的Add SSH Key，这样就会出现添加SSH Key的界面，在Title这一栏填一个名字，名字随意起，之后打开刚才生成的那个文件id\_rsa.pub，全选复制里面的内容到Key这一栏中，点击Add Key按钮完成操作，这时你填的邮箱会收到一封确认的邮件，不用管它**

****



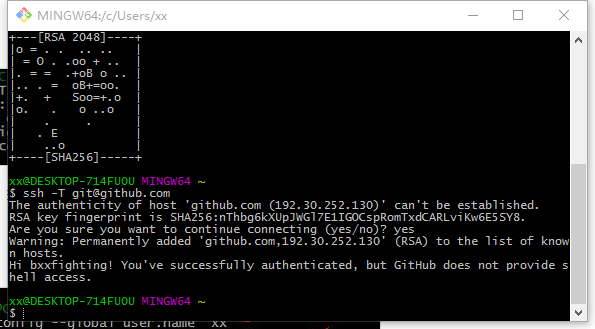




**e)        验证一下是否设置成功,在git bash下输入如下命令：**

ssh –T [git@github.com](mailto:git@github.com" \t "https://my.oschina.net/bxxfighting/blog/_blank)

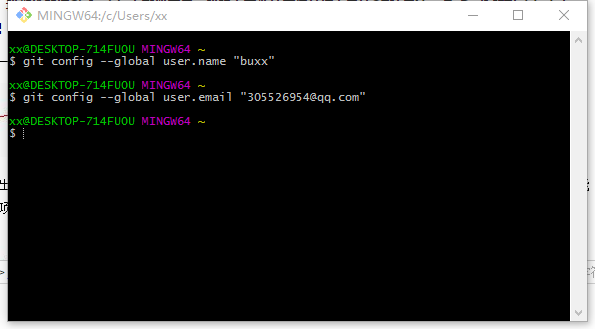
如果你是第一次，会让你输入yes或no,这时输入yes就可以了，其它显示就和我这个是一样的。如果你的是出现不是这些内容，有可能是显示权限问题什么的，就应该是我上面提到的那种情况，你看一下你生成密钥时是否操作正确，目录下是否有那个known\_hosts这个文件

****

**f)        现在配置一下用户名和邮箱：**

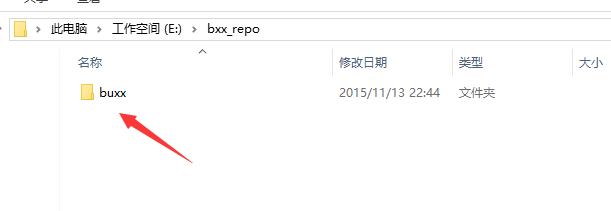
git config –global user.name “用户名”

git config –global user.email “邮箱”



**4.      到现在为止，我们就算把Git和github配置完了，现在就来托管我们的项目吧，刚才我们已经在github上面创建了一个叫buxingxing的仓库，那么我们现在就在本地创建一个目录，来管理这个仓库。**

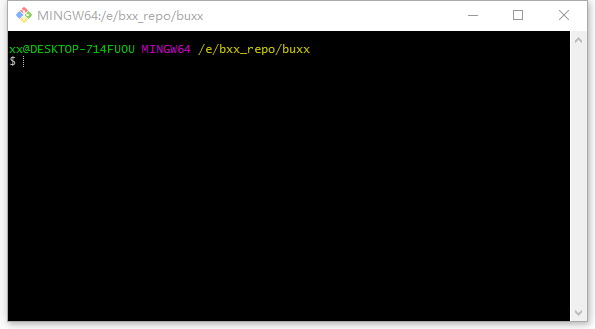
a)        随意创建了一个目录叫buxx



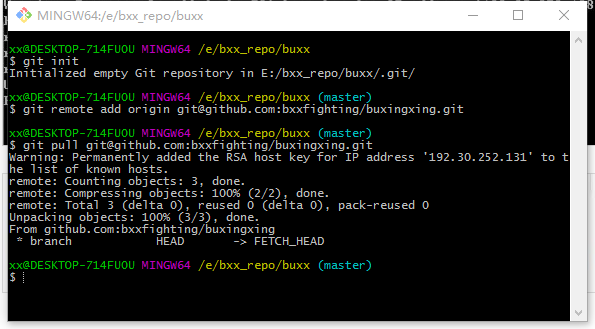
b)        右击目录，出现的菜单中有Git Bash Here，点击它。

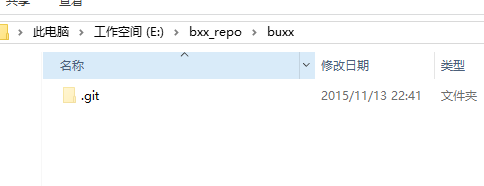


c) 这时候就在这个目录上打开了我们的终端。



d) 这时候输入 git init，来完成初始化工作。这时候目录里面就多了一个.git的目录了。注意中一共有三个命令，init/remote/pull均在上面显示，下面几步效果都在这张图中显示出来了。





现在继续在终端上输入如下命令，增加对我们github上创建的buxingxing仓库的管理。

git remote add origin git@github.com:bxxfighting/buxingxing.git

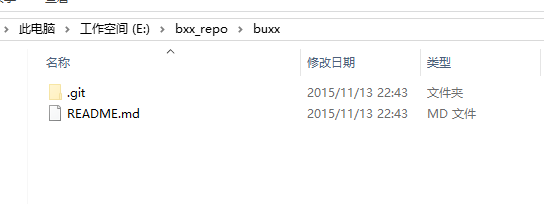
其中bxxfighting是我在网站上注册时使用的用户名，buxingxing.git是我为这个项目建立的仓库名，在网站上显示是这样的：



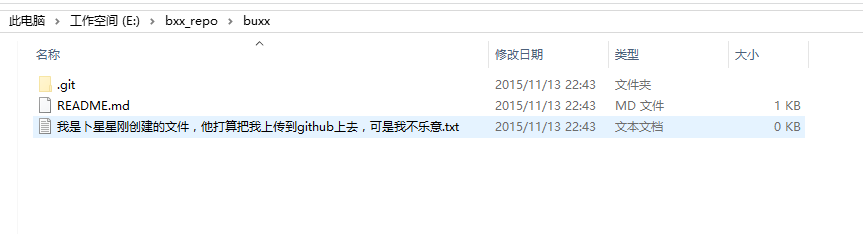
e)        由于我建立仓库的时候创建README.md之时，已经算一次提交了，我需要先在本地同步一下仓库的内容,命令如下：

git pull git@github.com:bxxfighting/buxingxing.git

完成的效果如下图，并且本地目录下多出了README.md文件

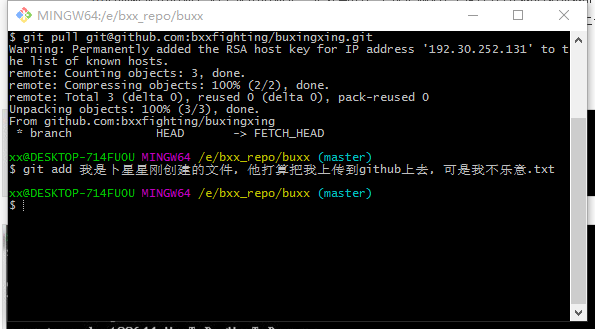


现在我们在本地目录创建我们的文件，比如下图是我创建的文件：



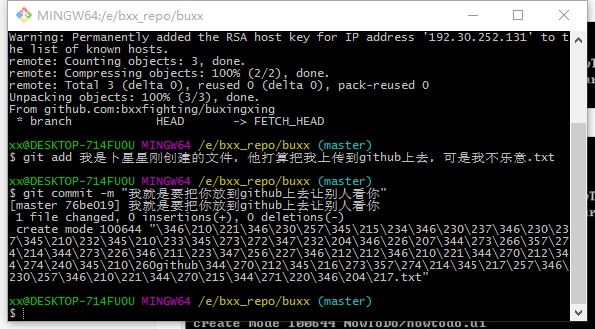
f)        下面就要把我刚创建的文件上传到到仓库上去了，首先执行增加命令，如下：

git add .    （这后面是一个英文的句号）



add后面加了一个点，是想要提交所有文件，如果想提交指定的文件，可以写文件名，执行完增加命令后，要执行提交命令，如下：

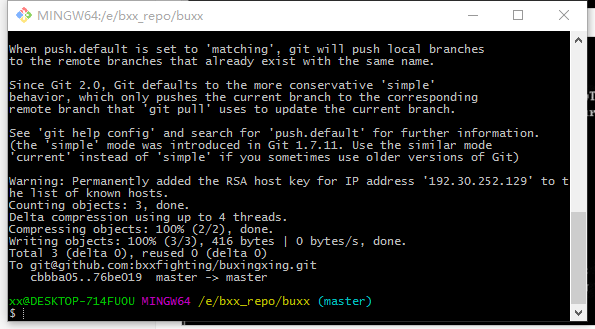
git commit –m “这里写下你自己的记录本次提交内容的信息”



-m后面跟提示信息，这个提示信息是一定要写的，不仅是规则，同时也方便我们记录我们提交的过程，写清晰为什么提交或修改了什么是非常有用的，提交完成后，我们就要把它推送到远程仓库上去了，命令如下：

git push [git@github.com:bxxfighting/buxingxing.git](mailto:git@github.com:bxxfighting/NowToDo.git" \t "https://my.oschina.net/bxxfighting/blog/_blank)

这样就完成了我们要做的所有任务





现在就基本上可以使用了，每次增加了新文件就先add，然后commit，如果只是改了文件的内容，只执行commit就行了，当然最后一步都是要执行push，把所以改变推送到我们的github上去托管。

其实，这里有一处提醒，就是最后push那一步，提醒中说，已经有更牛逼的办法了。所以，你自己去根据提示去看看什么牛逼的办法吧，如果懒，就这么用就行倒是。

到这就差不多了，我每次用其实都来看看，不是总是用，有点记不住，第一次这样就过就记下了，方便以后使用。其实

这真是非常方便的，可以使用熟练，管理自己的代码和笔记很好的东西，以前管理代码，最近发现每次看电子书，里面写上笔记，但是同步费劲，现在决定用它来管理了，回去在自己的笔记本上也这样来用，就可以同步自己看的进度了。