

## 参考资料

github简单使用教程

<http://wuyuans.com/2012/05/github-simple-tutorial/>

手把手教你最简单的开源项目托管GitHub入门教程

<http://jingyan.baidu.com/article/f7ff0bfc7181492e27bb1360.html>

windows下使用git及github仓库管理项目 入门

[http://blog.sina.com.cn/s/blog\\_700aa8830101kdp3.html](http://blog.sina.com.cn/s/blog_700aa8830101kdp3.html)

Fork和Pull-request的视频教程

<http://happycasts.net/episodes/37>

如何在github上fork一个项目来贡献代码以及同步原作者的修改

[http://www.360doc.com/content/13/0410/18/2569758\\_277424931.shtml](http://www.360doc.com/content/13/0410/18/2569758_277424931.shtml)

Pull Request的正确打开方式（如何在GitHub上贡献开源项目）

<http://blog.csdn.net/zhangdaiscott/article/details/17438153>

GitHub将Fork来的Project合并到原Project - Fork和Pull Request模式

<http://www.tuicool.com/articles/ZnERVn>

Git workflow指南: Pull Request workflow

<http://blog.jobbole.com/76854/>

## 1、注册github账号

打开浏览器，在地址栏输入地址：<https://github.com>

填写用户名、邮箱、密码

点击Sign up即可简单地注册

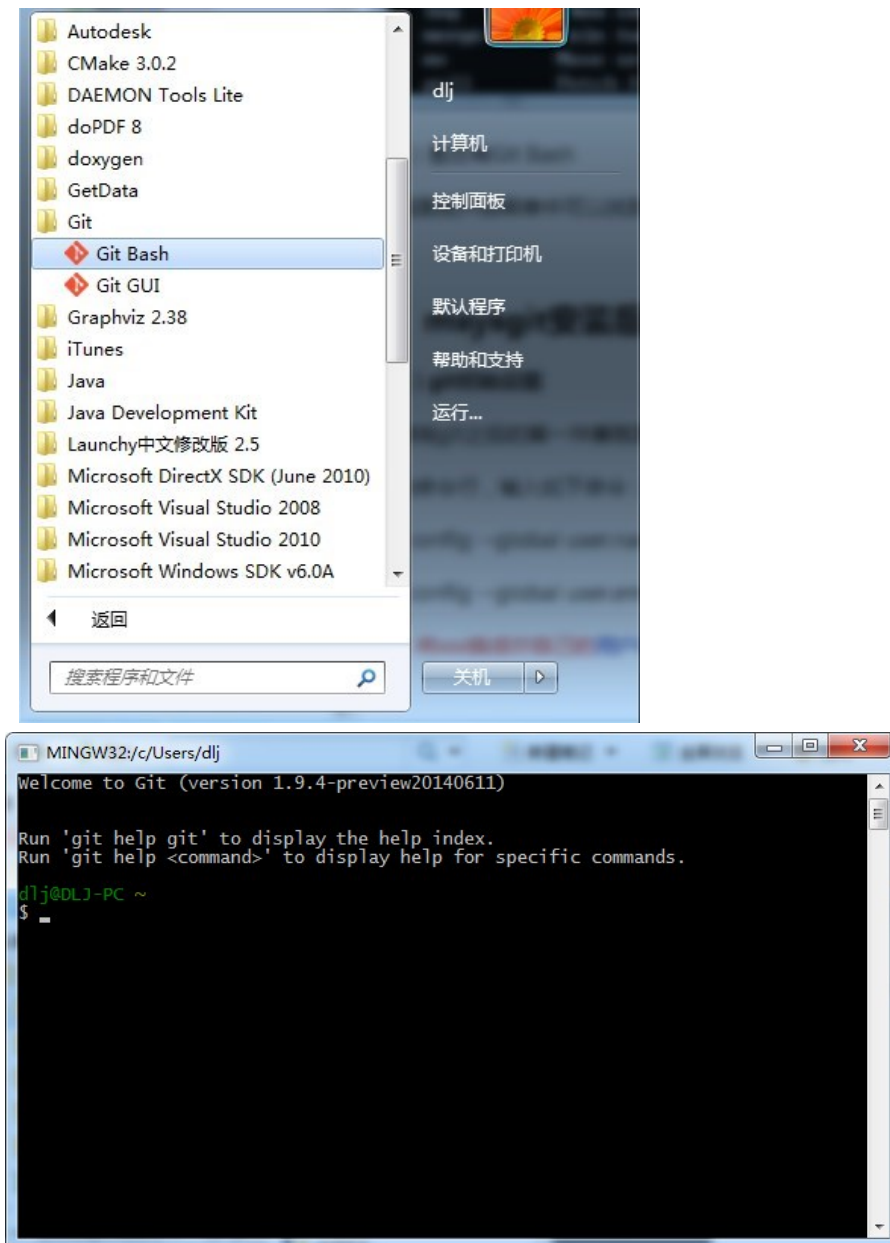
## 2、安装客户端msysgit

github是一个网站，属于服务端，要想在自己电脑上使用git我们还需要一个git客户端，我选用的是msysgit，它只提供了git的核心功能，而且是基于命令行的。

具体的安装及配置方法，参见[“msysgit安装及配置.pdf”](#)

## 3、生成SSH Key（可选）

打开Git bash（在桌面或开始菜单中可以找到Git Bash的快捷方式）



开始菜单-->Git-->Git Bash  
启动Git Bash

首先在本机创建ssh key;

```
ssh-keygen -t rsa -C "your_email@youremail.com"
```

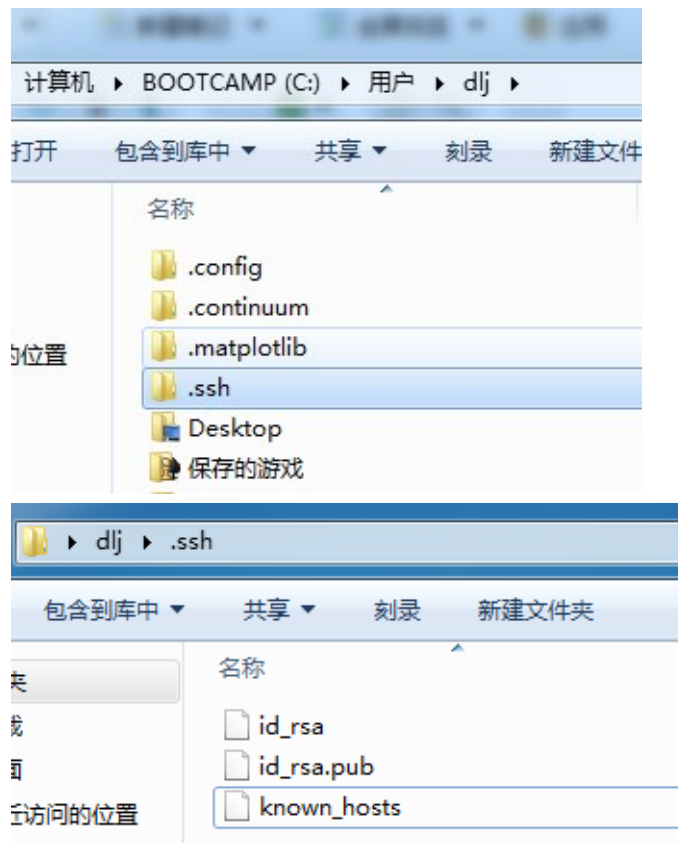
将`your_email@youremail.com`改为你的邮箱，之后会要求确认路径和输入密码，我们这使用默认值，一路回车就行。

成功的话会在用户的个人文件夹下生成 `.ssh` 文件夹，同时生成2个新的文件：

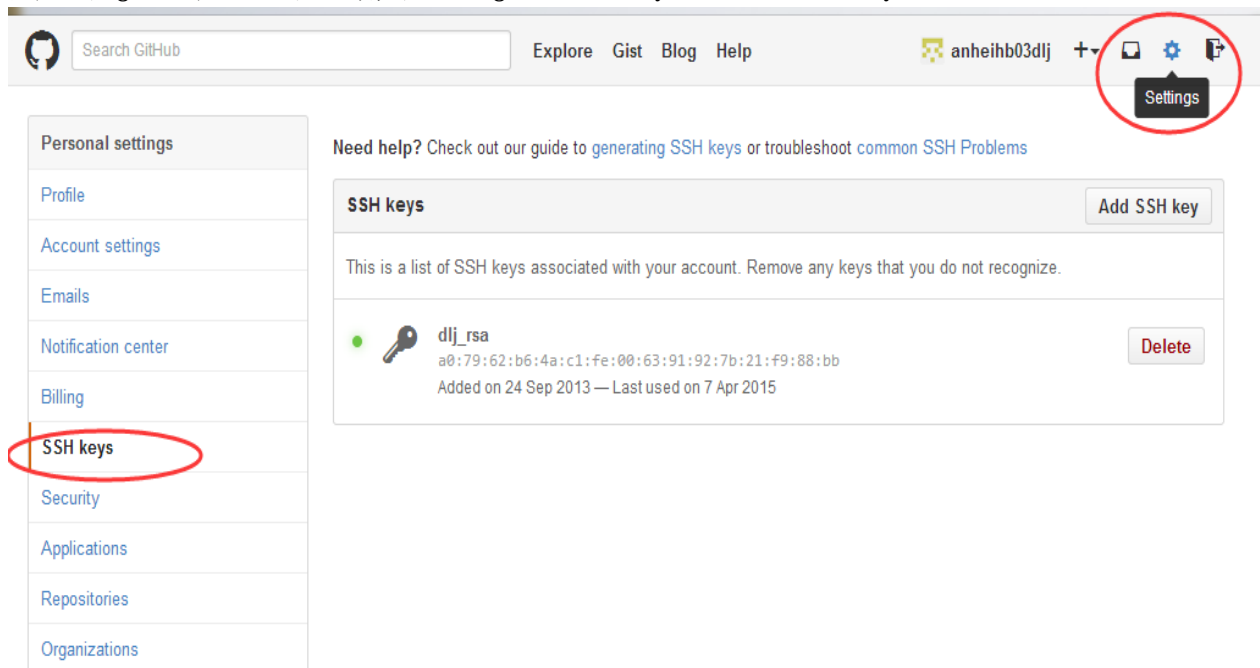
`id_rsa`                      密钥(private key)

`id_rsa.pub`                公钥(public key)

用记事本打开该文件夹里的`id_rsa.pub`，复制里面的所有内容



然后登陆github后, 点击右上角的Settings--> SSH keys --> Add SSH key



Title随便填, 粘贴id\_rsa.pub的内容到Key文本框中, 点击 add key 就ok了

Add an SSH key

Title

Key

Add key

测试连接是否成功，在Git Bash中输入以下命令：

```
ssh -T git@github.com
```

提示如下信息说明连接成功：

```
d1j@DLJ-PC ~  
$ ssh -T git@github.com  
Hi anheihb03dlj! You've successfully authenticated, but GitHub does not provide  
shell access.
```

说明：我个人不太喜欢 用Git Bash，用起来不太方便，更倾向于用windows自带的命令行cmd，所有后续的内容除非特别需要，我都会用命令行来代替Git Bash！！！！

## 4、fork本课程的仓库

打开本课程的github页面

<https://github.com/anheihb03dlj/learn-python>

点击上面的网址进入项目，然后点右上方的Fork按钮，这就拷贝一份我的项目的副本作为你自己的项目





fork成功之后，在你的github页面中应该能看到**learn-python**这个仓库（Repositories），并且显示你这个仓库是从anheihb03dlj@learn-python仓库fork过来的

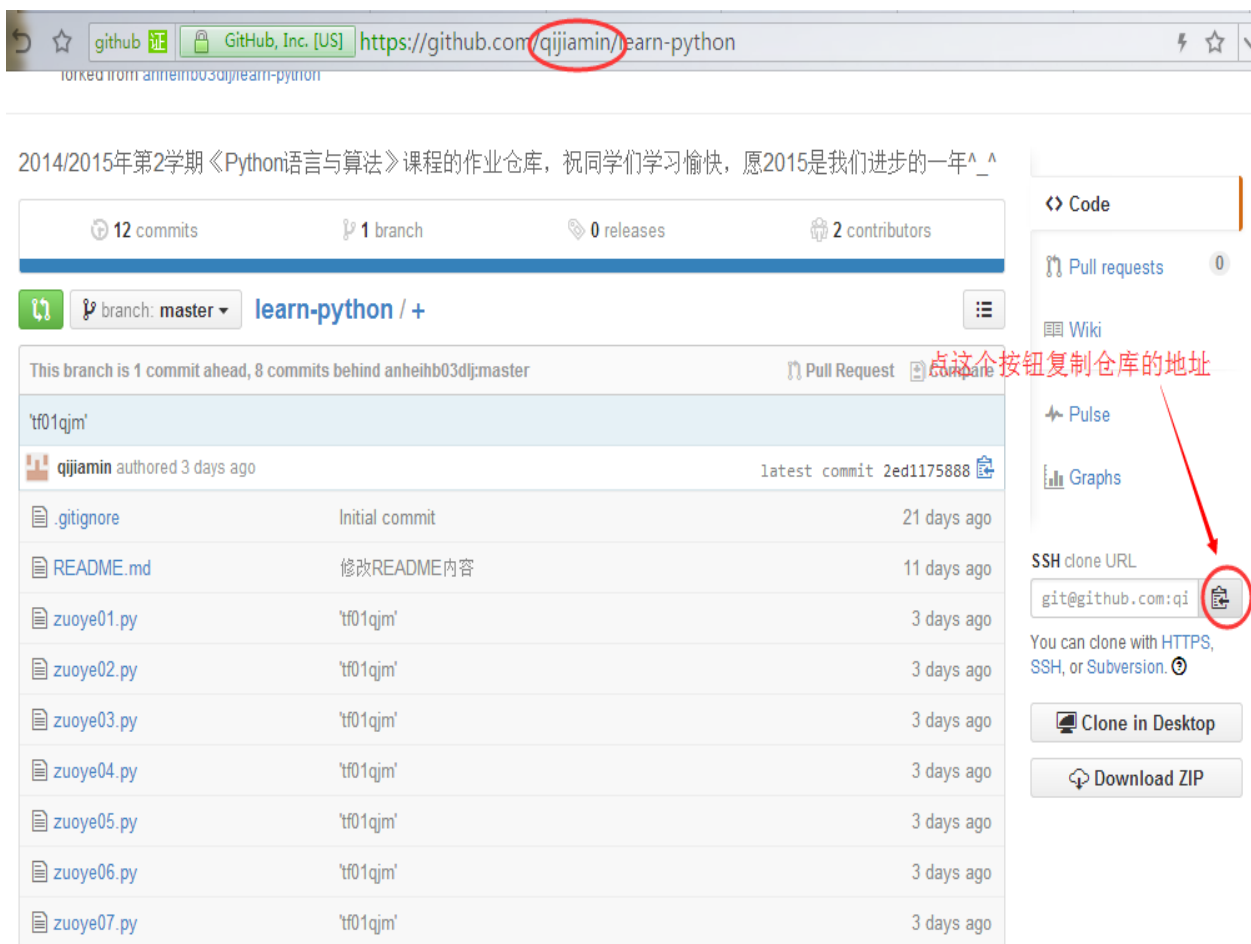
注：这里我用的是齐佳民同学的github仓库截图，希望没有侵权，嘿嘿



## 5、clone你自己的仓库到本地电脑上

**git操作的基本思路：用git客户端将仓库的代码复制到本地电脑上，修改后，再用git客户端将代码上传到github的仓库中**

5.1、点击下图所示的按钮复制仓库地址



细心的同学可能会发现，点击蓝色的超链接“[HTTPS](#)”或“[SSH](#)”后，SSH Clone URL的内容会发生变化

#### HTTPS clone URL

<https://github.com/qijiamin/learn-python.git>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

其实，github的仓库地址有2种：

<https://github.com/qijiamin/learn-python.git>

(https协议)

[git@github.com:qijiamin/learn-python.git](ssh://git@github.com:qijiamin/learn-python.git)

(ssh协议)

#### 它们的区别：

如果使用https协议的地址，每次上传代码到github仓库的时候，会要求输入github注册的用户名和密码；

如果你选择ssh协议的地址，并且已完成前面[第3节“生成SSH Key”](#)的操作过程，那么上传的时候，就不再要求输入用户名和密码

#### 给同学们的建议：

a、如果不想每次都输入用户名和密码，可以选择ssh协议的地址（前提是已完成[第3节“生成SSH Key”](#)的步骤）

b、如果和其他人共用一台电脑，可以选择https协议的地址（每次上传的时候，输入你自己的github用户名和密码）

c、如果不想折腾[第3节“生成SSH Key”](#)，建议同学们选择https协议的地址

## 5.2、clone仓库到本地

注意，这里clone的是你的仓库，而不是我的仓库（<https://github.com/anheihb03dlj/learn-python.git>）  
还是用齐佳民同学的仓库做例子，使用https协议的地址

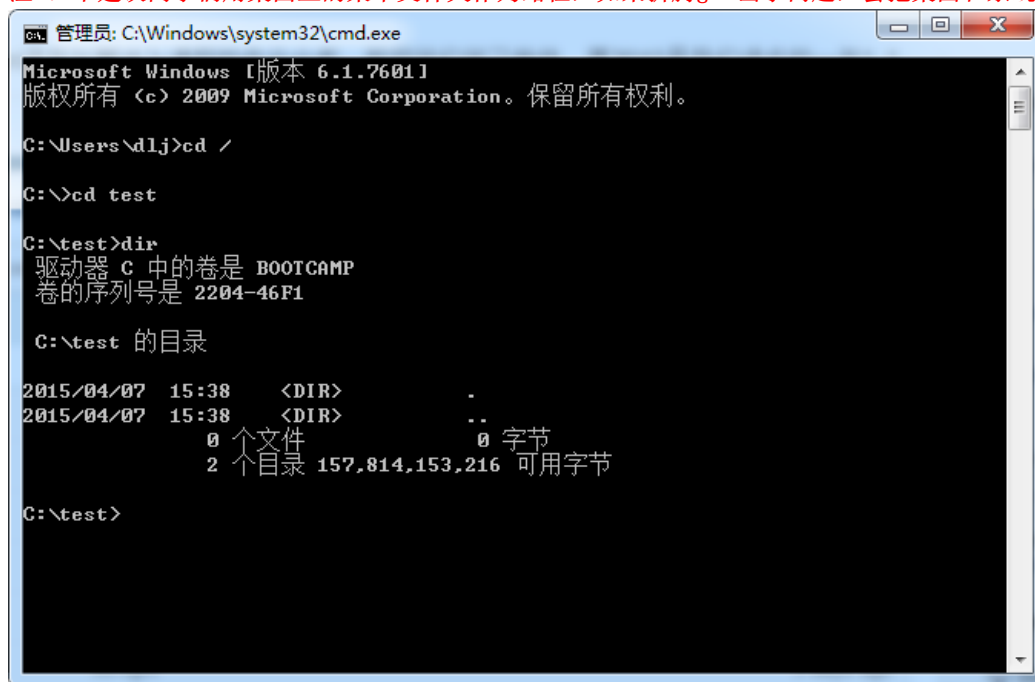
(1) 打开命令行，切换到某个路径下，比如C:\test

注1：如果C盘没有test文件夹，新建一个就可以

注2：如果不想放在C盘，放在别的盘某个文件夹也可以，比如D:\study\python，不用非要和我的一样

注3：路径尽量不要有中文和空格

注4：不建议同学们用桌面上的某个文件夹作为路径，如果折腾git出了问题，会把桌面和系统弄残废的！！



```
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\dlj>cd /

C:\>cd test

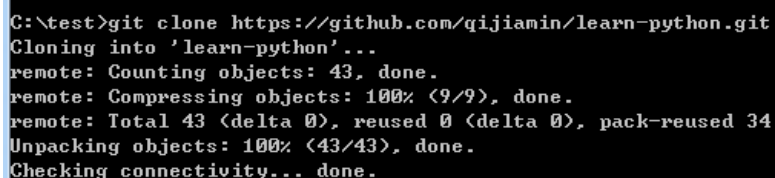
C:\test>dir
驱动器 C 中的卷是 BOOTCAMP
卷的序列号是 2204-46F1

C:\test 的目录
2015/04/07 15:38 <DIR>      .
2015/04/07 15:38 <DIR>      ..
                0 个文件          0 字节
                2 个目录 157,814,153,216 可用字节

C:\test>
```

(2) 输入以下命令，将仓库克隆到本地

```
git clone https://github.com/qijiamin/learn-python.git
```



```
C:\test>git clone https://github.com/qijiamin/learn-python.git
Cloning into 'learn-python'...
remote: Counting objects: 43, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 43 (delta 0), reused 0 (delta 0), pack-reused 34
Unpacking objects: 100% (43/43), done.
Checking connectivity... done.
```

小技巧：在输入仓库地址的时候，输入完git clone后，按空格，再点鼠标右键，弹出菜单，选择“粘贴”即可（前提是已经复制了仓库的地址）

## 6、修改和添加代码

clone成功之后，在[当前路径（第5步设置的路径）](#)下能看到一个文件夹，文件夹的名字就是仓库的名字learn-python

6.1、用如下的命令切换learn-python文件夹，并查看该文件夹的内容

```
C:\test>cd learn-python

C:\test\learn-python>dir
驱动器 C 中的卷是 BOOTCAMP
卷的序列号是 2204-46F1

C:\test\learn-python 的目录

2015/04/07  15:40    <DIR>          .
2015/04/07  15:40    <DIR>          ..
2015/04/07  15:40                729 .gitignore
2015/04/07  15:40            7,581 README.md
2015/04/07  15:40                206 zuoye01.py
2015/04/07  15:40                804 zuoye02.py
2015/04/07  15:40                383 zuoye03.py
2015/04/07  15:40                800 zuoye04.py
2015/04/07  15:40                397 zuoye05.py
2015/04/07  15:40                249 zuoye06.py
2015/04/07  15:40                774 zuoye07.py
                9 个文件             11,923 字节
                2 个目录  157,782,155,264 可用字节
```

## 6.2、添加py代码到learn-python文件夹

首先，和同学们约定作业代码的命名规则：**test{x}-{y}.py**

其中{x}表示第几次作业，{y}表示该次作业的第几个题目

例如：test1-1.py表示第1次作业的第1个题目

test3-6.py表示第3次作业的第6个题目

其次，py文件的模板参考：**test0-0.py**

以后做作业把test0-0.py文件复制一份改名后再写你们自己的代码

将你们的作业代码按照如上的要求写好后，放在learn-python文件夹里

## 6.3、通过git命令将代码缓存并提交到仓库中

考虑到很多同学对命令行不太习惯，这里推荐用**git gui**来进行操作，它实质就是git命令的有界面版本，方便初学者使用git

在命令行中输入：**git gui**

```
C:\test\learn-python>
C:\test\learn-python>git gui
```

弹出如下的窗口，通过“**未缓存的改动**”列表，我们可以看出文件的状态，比如

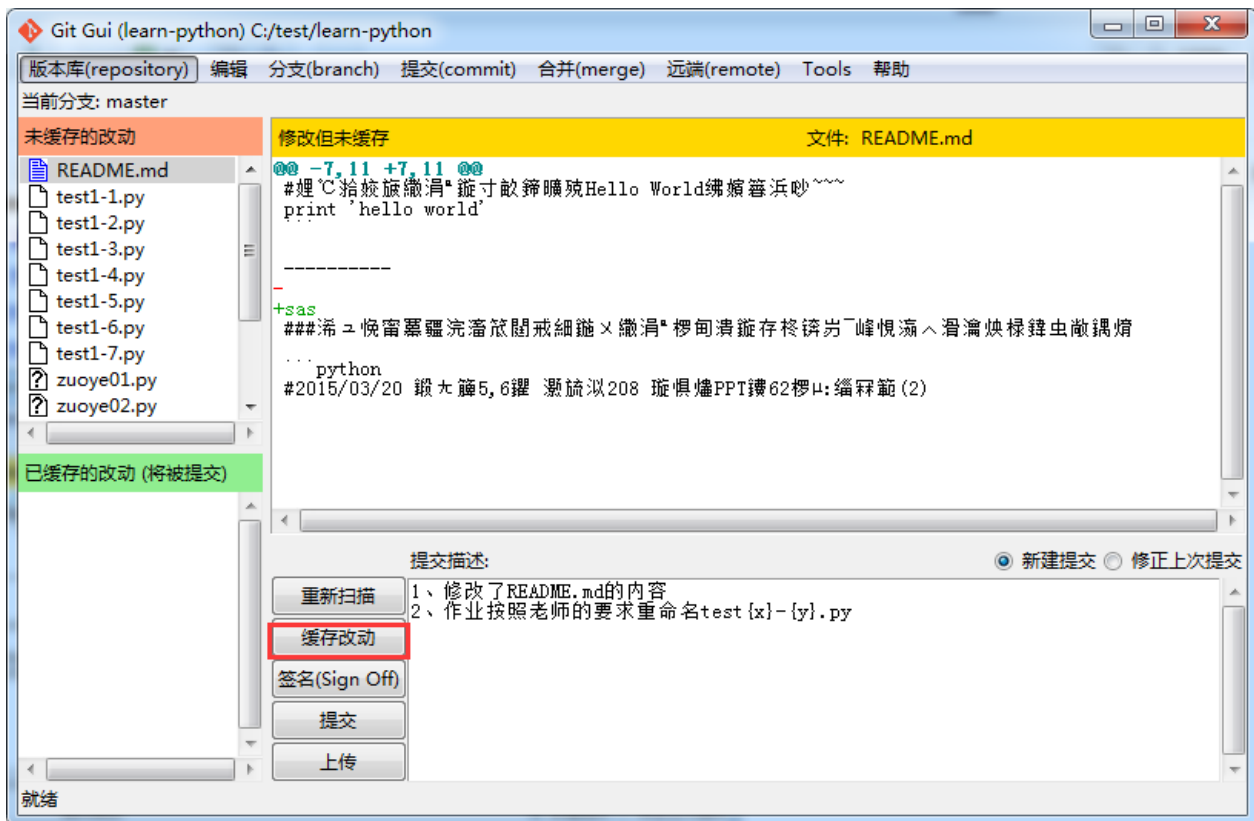
空白图标的文件      表示该文件是新增加的

带问号的文件      表示该文件上一次提交的时候还是存在的，但现在已经被删除了

带蓝色线条的文件      表示该文件以前存在，但内容被修改了（**+号**表示这一行是新增的，**-号**表示

这一行被删除了）





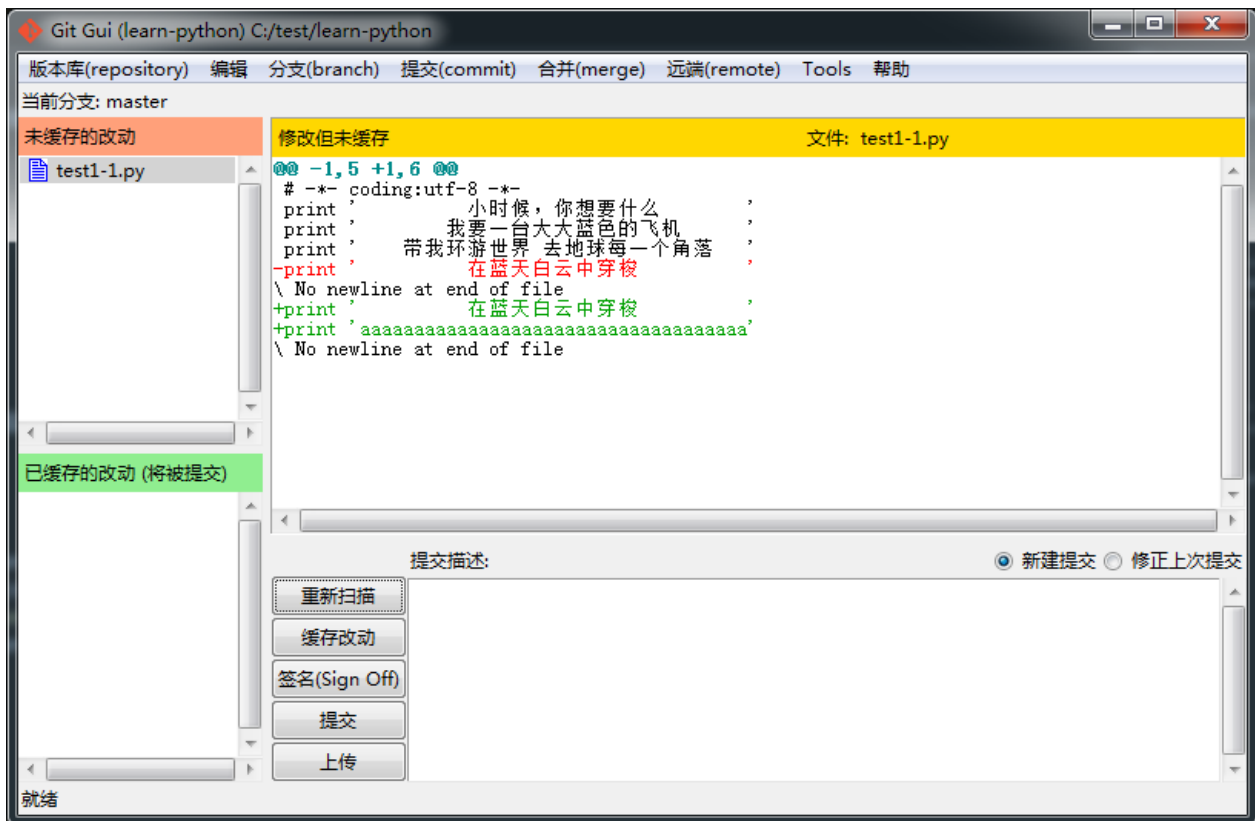
同学们可以逐个查看文件，确认修改没有啥问题或者遗漏，点击”**缓存改动**“按钮，然后在”**提交描述**“写上一段话，来描述你本次提交的时候做了哪些修改（这个提交描述并没有具体规定如何写，同学们按照自己的意愿来写就行，把你们的代码主要做了哪些修改说明白就行），然后点击”**提交**“按钮提交到本地仓库中

## 7、再次修改代码

提交之后，可能有的同学比较好学，又把自己写的代码看了一遍，发现有的地方写错了，或者代码运行不正确，他又修改了一下，比如test1-1.py里又加了一行。用notepad++**将代码改完后，保存一下，我们再次执行命令：git gui**

注1：写代码和使用git的时候，命令行不要关了，要不然还得启动命令，还得执行cd等命令，麻烦~

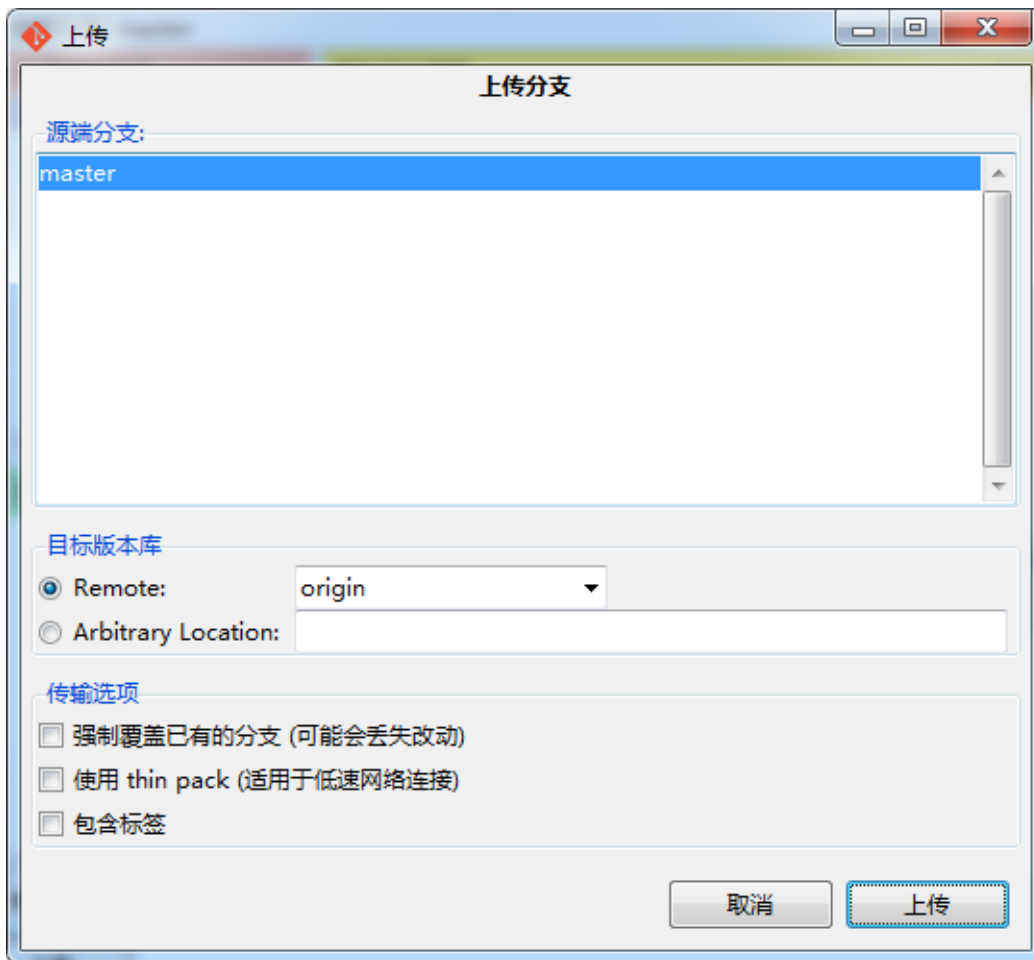
注2：如果没有要修改的，跳过这一步即可



很显然，我们可以清晰看到所做的修改，然后和上一步的操作一样：“缓存改动”-->在“提交描述”写一段话-->提交“

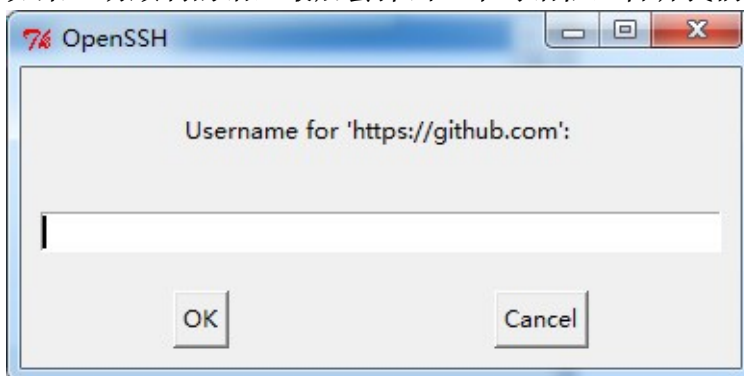
## 8、上传代码到你的github仓库

命令行输入命令：git gui，然后点击”上传“按钮，弹出如下对话框，点击右下角的”上传“按钮



如果“[5.1、点击下图所示的按钮复制仓库地址](#)”小节中选择的仓库地址是https协议的，接下来会弹出2次对话框，分别要求输入你的github用户名和密码

如果一切顺利的话，最后会弹出一个对话框，告诉我们上传成功了



小技巧：如果想知道仓库的详细地址，可以通过如下命令来查看

输入命令：`git remote -v`

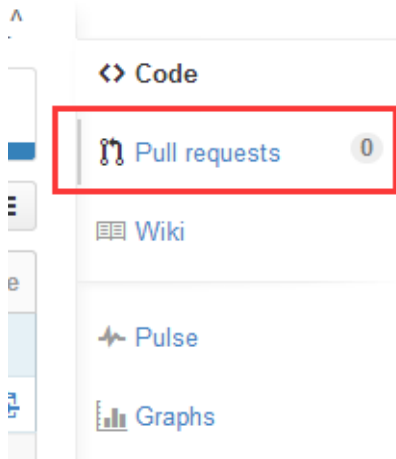
```
C:\test\learn-python>git remote -v
origin  https://github.com/qijiamin/learn-python.git (fetch)
origin  https://github.com/qijiamin/learn-python.git (push)
```

## 9、登录Github，并刷新你的仓库

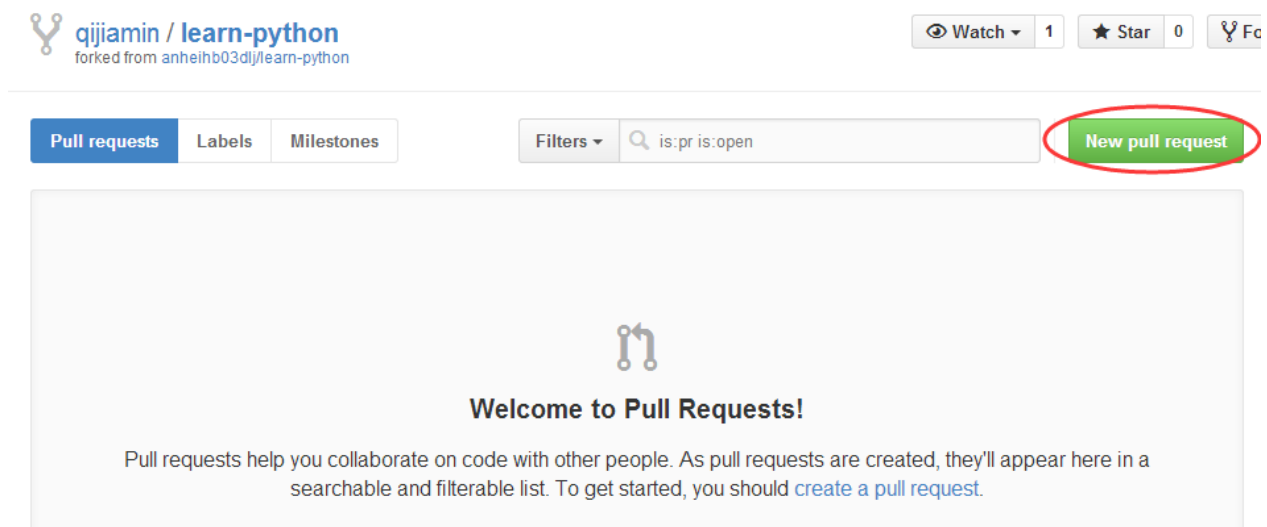
这时，同学们应该看到自己的learn-python仓库页面已经更新了吧。。。

## 10、通过Github的Pull-Request将同学们的作业发送我的仓库

打开你的learn-python仓库，点击右侧的“Pull requests”



点击 “New Pull Request”



转到如下的页面，要选择2个内容：**base**和**compare**

**base**选择的是你自己的仓库**master**分支，比如qijiamin/learn-python

**compare**选择的是我的仓库**master**分支，比如anheihb03dlj/learn-python

## Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also [compare across forks](#).

然后填写标题（用来简要说明你的pull-request是干什么的）、内容（可以适当描述一下做了哪些修改，或者你想解决什么问题）

比如同学们交作业，就可以在标题写上“通风14-1班xxx的作业”


然后，点击“[Create pull request](#)”按钮即可成功发送你们的作业代码到我的仓库了。

通过我的仓库（<https://github.com/anheihb03dlj/learn-python>）右侧的“Pull request”，大家都可以看到这个同学的Pull-request，也就是他的所有作业以及代码修改情况，并参与讨论（下面有一个评论框，可以写评论，比如点赞啊，请教做题方法和思路啊等等，类似于贴吧和论坛）

注1：点击下图的“Files changed”可以看到该同学的所有作业代码


注2：点击下图的“Commits”可以看出该同学是否写作业了，勤快爱学习的同学能看出他的代码经常修改、提交和上传，而懒惰抄袭的同学基本上不会有任何更新

## 'tf01qjm' #1

 **Open** qijiamin wants to merge 1 commit into `anheihb03dlj:master` from `qijiamin:master` Conversation 1 Commits 1 Files changed 7

qijiamin commented 3 days ago

通风14-1班齐佳民第一次作业

 'tf01qjm'[2ed1175](#)

anheihb03dlj commented 3 days ago


Owner

挺不错，给你100个赞，嘿嘿！

**This pull request can be automatically merged.**You can also merge branches on the [command line](#).**Merge pull request**

Write

Preview

 Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Close pull request

**Comment**

11、以后有新的作业，或者同学对原来的代码进行了修改，都可以重复6.3-->7-->8这3个步骤（不需要再创建新的pull-request了），这样你们github上的learn-python仓库会更新，同时你们的更新也会自动发送到[第10步](#)创建的pull-request

12、学到这一步，同学们基本可以对付着交作业了，但如果有的同学想更进一步的话，可以系统的学习一下git的教程（在我的github页面上有）