**Tesseract以及验证码破解**

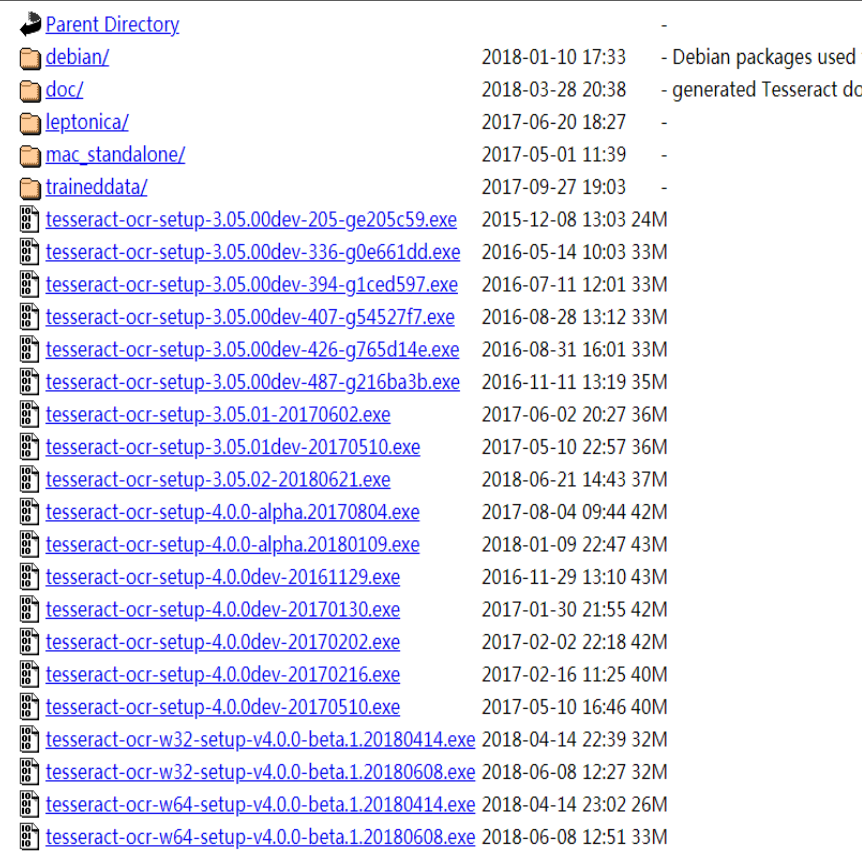
今天我们学习验证码破解，我们先来学习tesseract。

Tesseract就是OCR，即Optical Character Recognition，光学字符识别，是指通过扫描字符，然后通过其形状将其翻译成电子文本的过程。对于图形验证码来说，它们都是一些不规则的字符，这些字符确实是由字符稍加扭曲变换得到的内容。

言而言之，就是机器学习识别验证码，因为有些网站需要验证码登录，为了防止爬虫，魔高一尺道高一丈，你有你的验证码，我就有我的破解

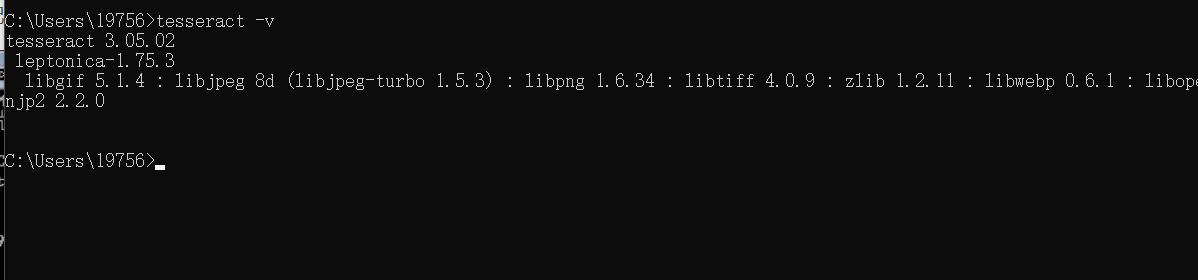
那么好了，我们先来安装treseract，<https://digi.bib.uni-mannheim.de/tesseract/>

进入下载界面有好多啊，我们要选择和个人版本对应的版本，

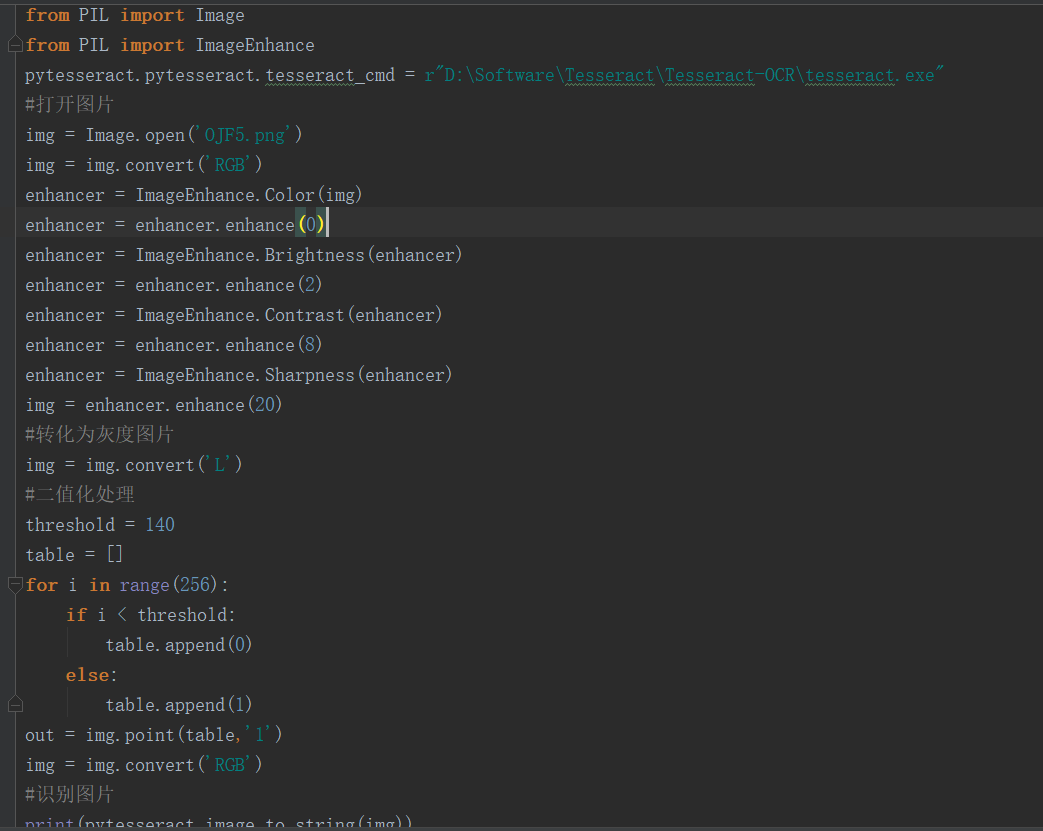


然后安装过程不做叙述，为了在python代码中只用teeseract功能，我们需要使用pip Install pytresseract

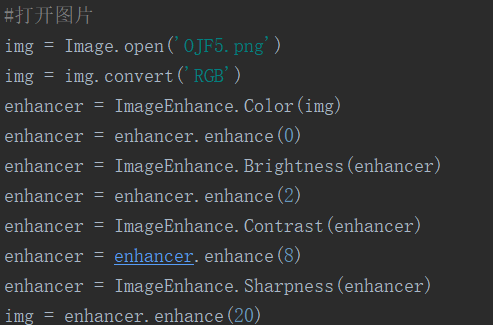
然后这个和别的不一样我们还需要在环境变量的path中配置一下环境变量在命令行输入tresseract -v

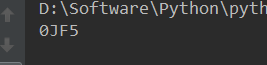


出现以上界面，就安装好了，那么开干吧

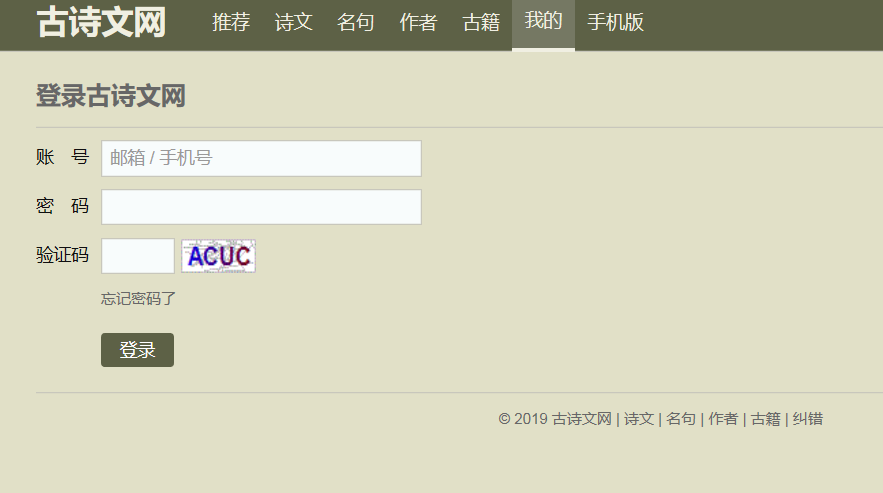


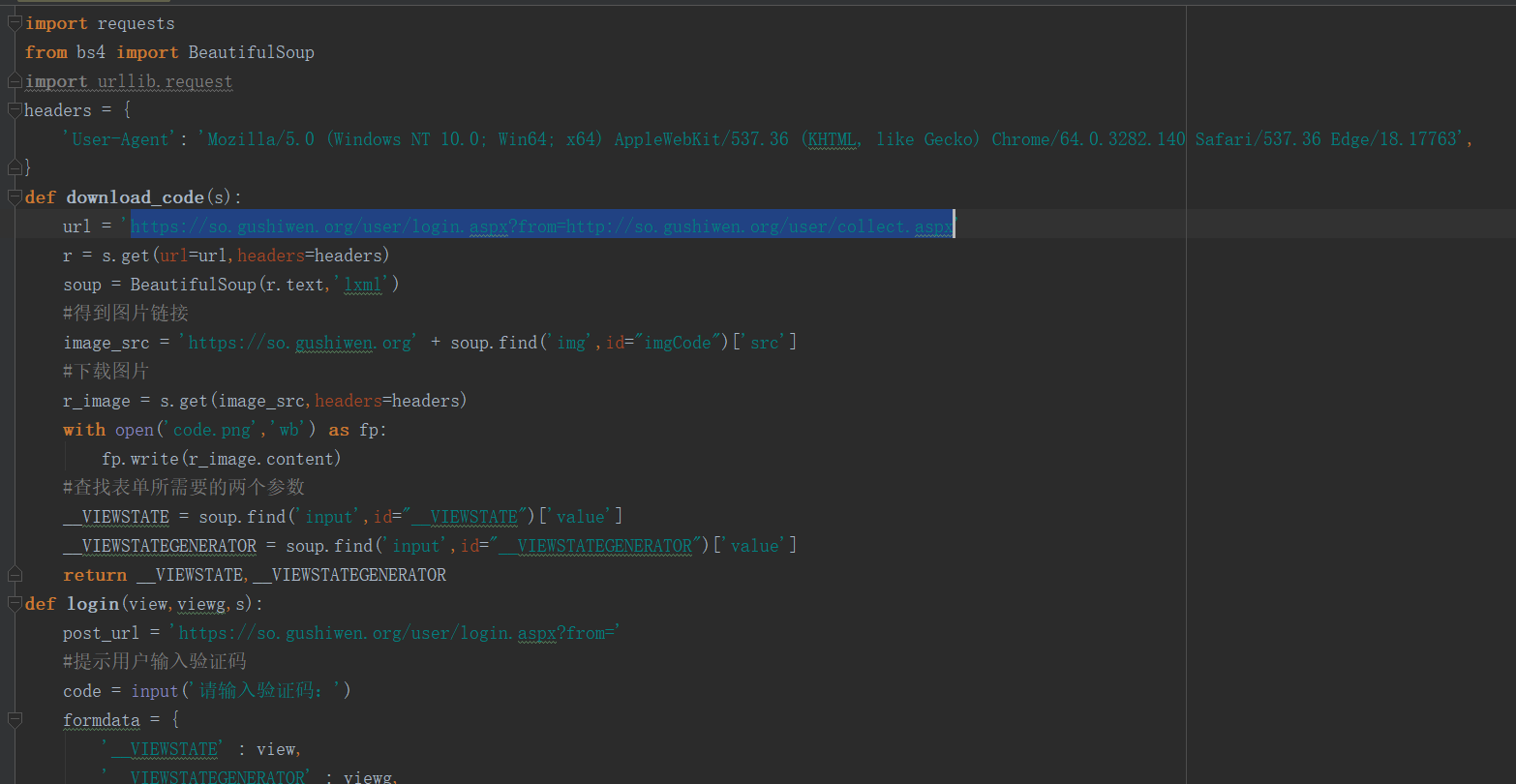
代码分析：

* 首先导入库，from PIL import Image、from PIL import ImageEnhance这俩是python自带的，然后导入pytressact
* 然后把tresseract软件路径输入
* 我已经准备好了图片
* 然后就是导入图片然后进行机器学习
* 在这里我们要转化为灰度图片，因为我会ps，转化为灰度图片的话，里面的文字会更好的显示出来，然后就是二值化处理，其实这里有很多需要学习的，随便先试几个
* 最后我们输出结果，识别图片，我们来看效果

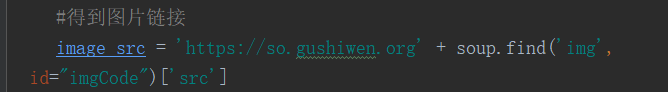
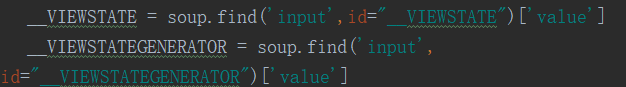
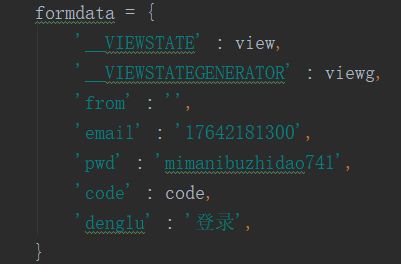


我靠，这么准确，真好，那么我们直接爬取带验证码的网站，但是有的验证码还是很难得，因为我又试了几个比较难的，他识别效果不是特别好，应该是机器学习还不行，那我就有了另外的思路，我们可以把验证码图片通过xpath获取到然后下载通过人工输入，那样的话我们也可以实现，废话不多说，开干。

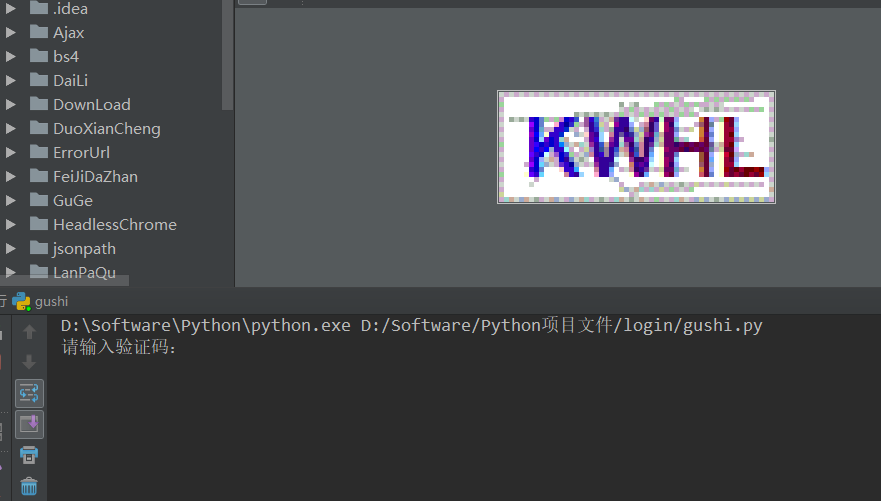
我们爬取的是

中国古诗文网站，其实我们可以用机器学习来做，但是思路都摆在那里了，我们就干吧，上代码：

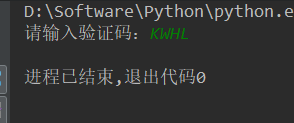
分析代码：

* 首先导入库
* 然后定义主函数，这是我们爬虫的主程序，我们这里还是用的面向对象，首先要创建对话，s = requests.Session()
* 然后就是下载验证码到本地，然后再发送请求，那么我们先写下载验证码的类，download\_code（s）,把刚才创建的对话当作参数传过去
* 在这个类里面，我们肯定要先写url，要让机器知道我们下载那里的图片，然后就用刚才创建的请求来创建请求，发送请求，另外我们在这里使用的是bs4，活学活用嘛，这都是老规矩了
* 然后就是得到图片链接，我没用xpath，用的是bs4
* 然后就是下载图片，下载完之后，重点来了，图片我们不是全部需要，我们需要的是里面的文字，那么我们就查找表单所需要的连个参数
* 然后return返回这两个参数内容，到现在我们已经获取到验证码
* 回到主函数，接收这两个参数，然后我们需要手动输入，这个在爬取时实现，接下来我们就要想post地址发送请求，我们需要另一个类，login，我们还需要把刚才接受的两个参数还有创建对话这三个当作参数传进去
* 然后我们来到login类里面，因为涉及到登录所以我早就注册了一个账号，需要获取cookies内容，经过登录抓包，我们获取到表单数据，formdata，定义一个字典，来存放
* 然后写入url地址，然后提示用户输入验证码，在这个过程里面，验证码已经被刚才的类下载到本地了，我们直接输入就行
* 然后表单数据我们已经写入，接下来就是发送响应，然后数据就应该到手了，我们直接在本地创建一个文本文档，用来存放数据

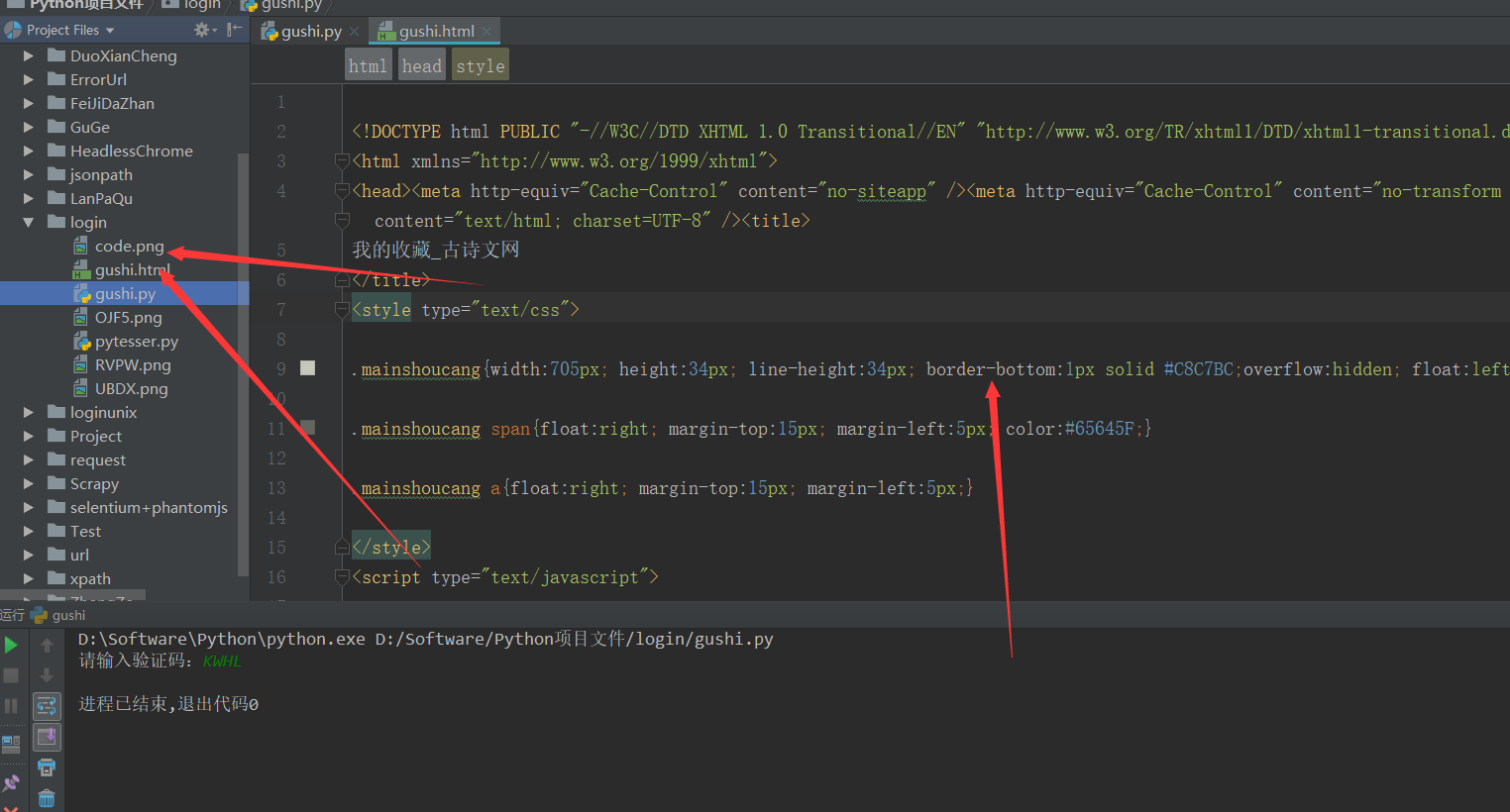
下面是运行过程：



看，验证码成功被下载，那么我们输入验证码

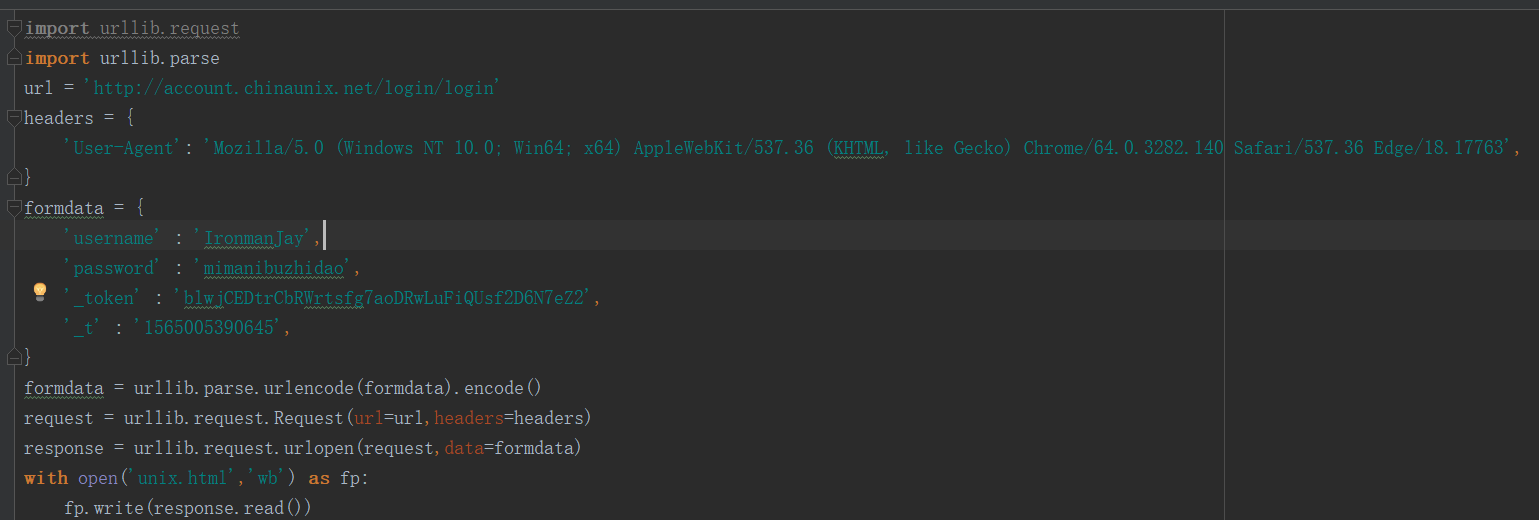


下面是代码运行结果，获取到了数据：

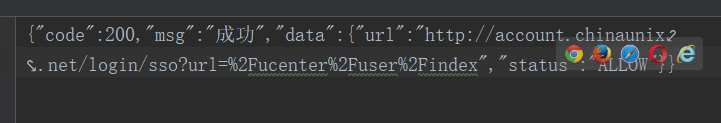


大功告成！

PS:在这里另外做了一个小爬虫

附上代码：  


依旧成功运行：



总结，这个东西真的让我很惊讶，没想到还能这么玩，但是也遇到了许多问题，大都是tresseract的，变量值中的路径为“D:/Program Files (x86)/Tesseract-OCR”，使用正斜杠“/”。windows中复制过来的路径默认是反斜杠“\”，这是第一个，然后配置完成后，正常了，然后在使用pytesseract的时候出现错误：

Traceback (most recent call last):

File "D:\Python36\lib\site-packages\pytesseract\pytesseract.py", line 170, in run\_tesseract

proc = subprocess.Popen(cmd\_args, \*\*subprocess\_args())

File "D:\Python36\lib\subprocess.py", line 709, in \_\_init\_\_

restore\_signals, start\_new\_session)

File "D:\Python36\lib\subprocess.py", line 997, in \_execute\_child

startupinfo)

FileNotFoundError: [WinError 2] 系统找不到指定的文件。

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

File "D:/python/20180911.py", line 4, in <module>

text = pytesseract.image\_to\_string(Image.open(r'D:\chromeDownload\image.png'))

File "D:\Python36\lib\site-packages\pytesseract\pytesseract.py", line 294, in image\_to\_string

return run\_and\_get\_output(\*args)

File "D:\Python36\lib\site-packages\pytesseract\pytesseract.py", line 202, in run\_and\_get\_output

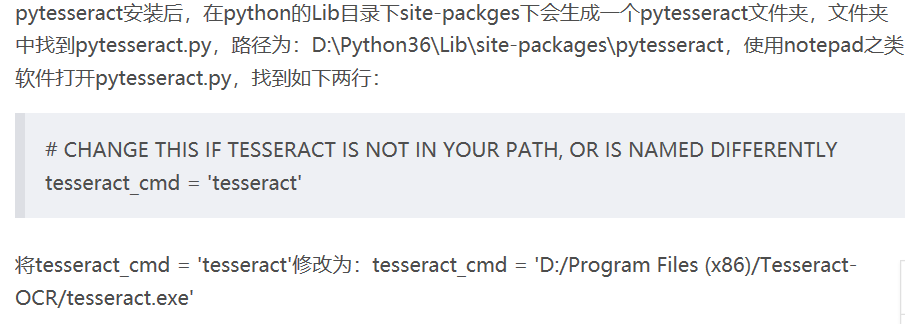
run\_tesseract(\*\*kwargs)

File "D:\Python36\lib\site-packages\pytesseract\pytesseract.py", line 172, in run\_tesseract

raise TesseractNotFoundError()

pytesseract.pytesseract.TesseractNotFoundError: tesseract is not installed or it's not in your path

这就很坑，添加了全局变量，还是提示tesseract没有安装或者不在path中，我百度了一下，解决方案如下：



原来表示tesseract\_cmd配置的是你安装tesseract的绝对路径，这样就能找到tesseract了。修改后保存，再去运行python代码，就可以成功了。

看来以后验证码的网站也不怕了哈哈哈，通过这个也学到了不少，机器学习现阶段还是不靠谱，那么我们就自动手动输入，这个还是很累的，没达到预期，接下来就要搞机器学习，让代码识别验证码更准确，这样就能解放双手了！