**Request**

这里是一个大的学习，基本就是爬虫的入门，包括cookies、代理、get请求、post请求

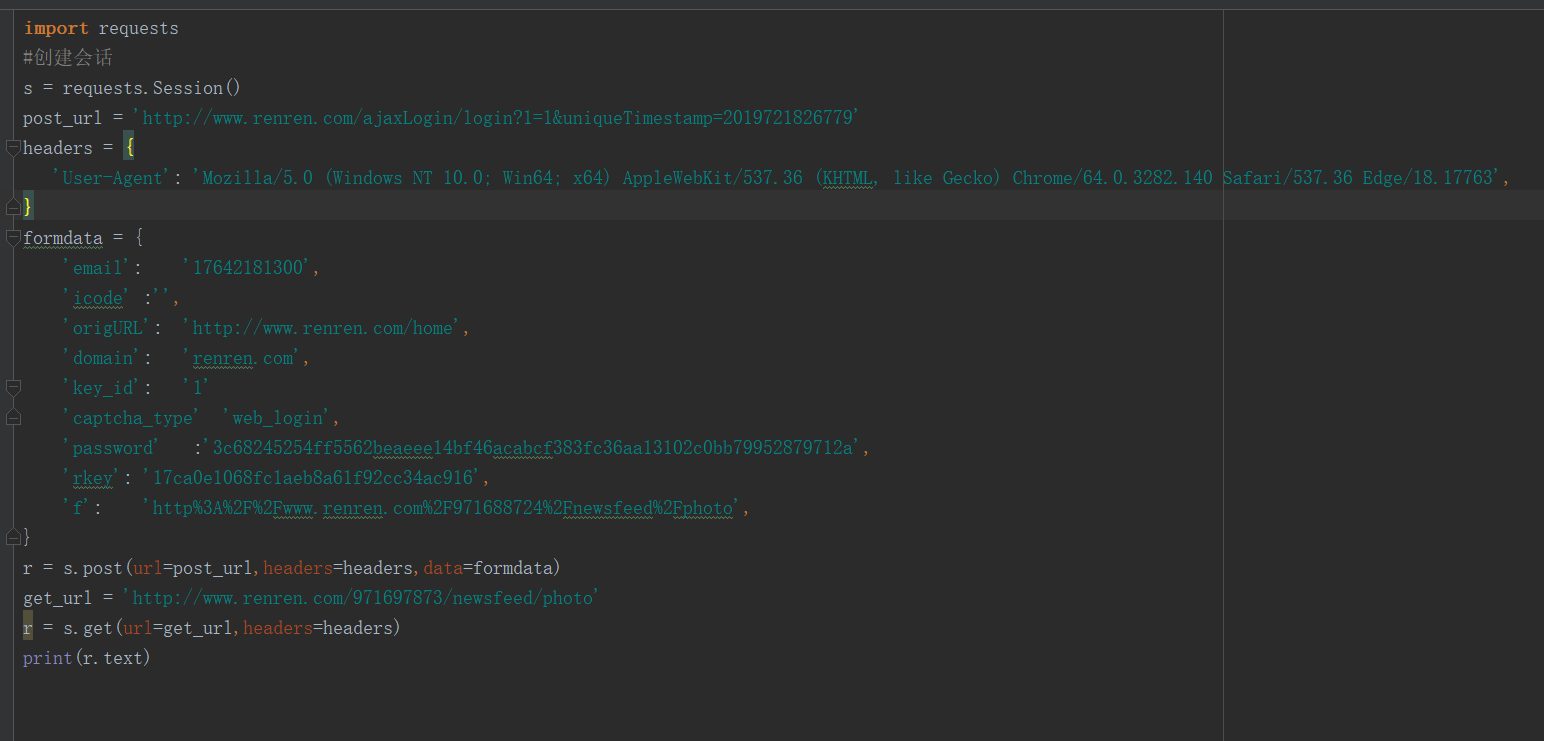
以及两个小案例

一个一个来。

1. cookies：

cookies就是网站为了辨认用户身份的东西，也是为了防止爬虫吧，当你请理浏览器缓存的时候就会发现有这个选项，如果你清除了，下次登录就会重新验证，我们直接来看实例。

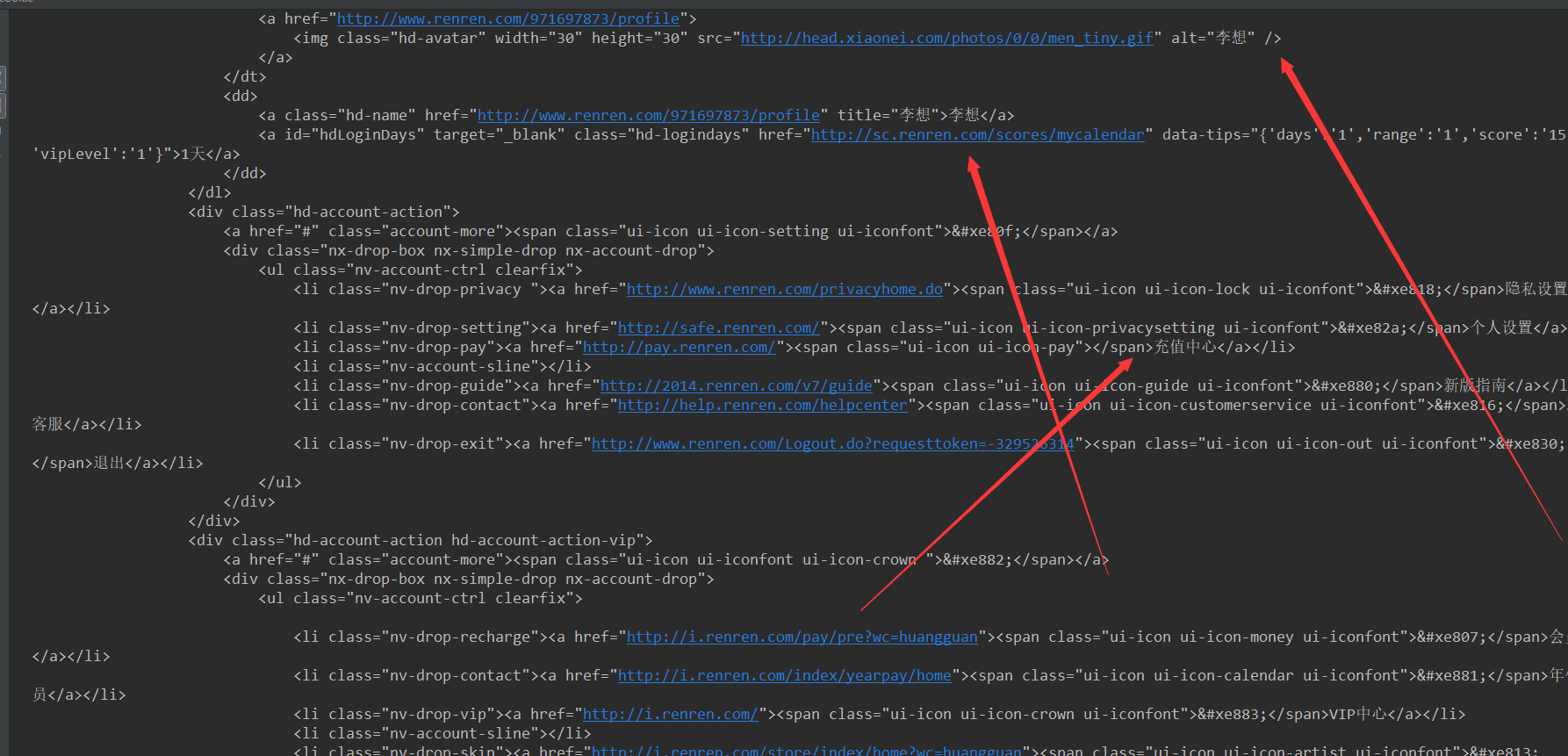
下面是代码：



分析代码：

* 第一步还是老规矩，uel和请求头，这里的formdata就是cookies里面的数据
* 
* 我们可以看到，里面有email，id，我密码以md5形式展示，所有我的登录信息都在这里，这就是这个网站的cookies，我们通过抓包获取到这些信息
* 然后就是构建请求，发送请求，最后打印，这个项目没有什么，就是更多的了解一下cookies，我们登陆的信息都在cookies里面，在写程序爬虫的时候直接爬肯定是不行的，这个时候我们就要获取用户登录的cookies，就能登陆上去也就能爬虫了。

下面是运行结果：



1. 代理

代理就是proxy，我么爬虫的时候肯定会频繁访问某一个网站，这个时候有些服务器会识别我们是非正常访问，就会把我们的IP禁掉，这个时候需要代理了，也就是把我们的ip换掉，这里我找到一个免费的代理网站：<http://www.goubanjia.com/>，在这里还需要用一个软件，因为免费的都不太靠谱，我使用的是一个检测ip是否可用的软件：



非常方便，可以判断是否可用，然后一键设置代理

下面是代码：  

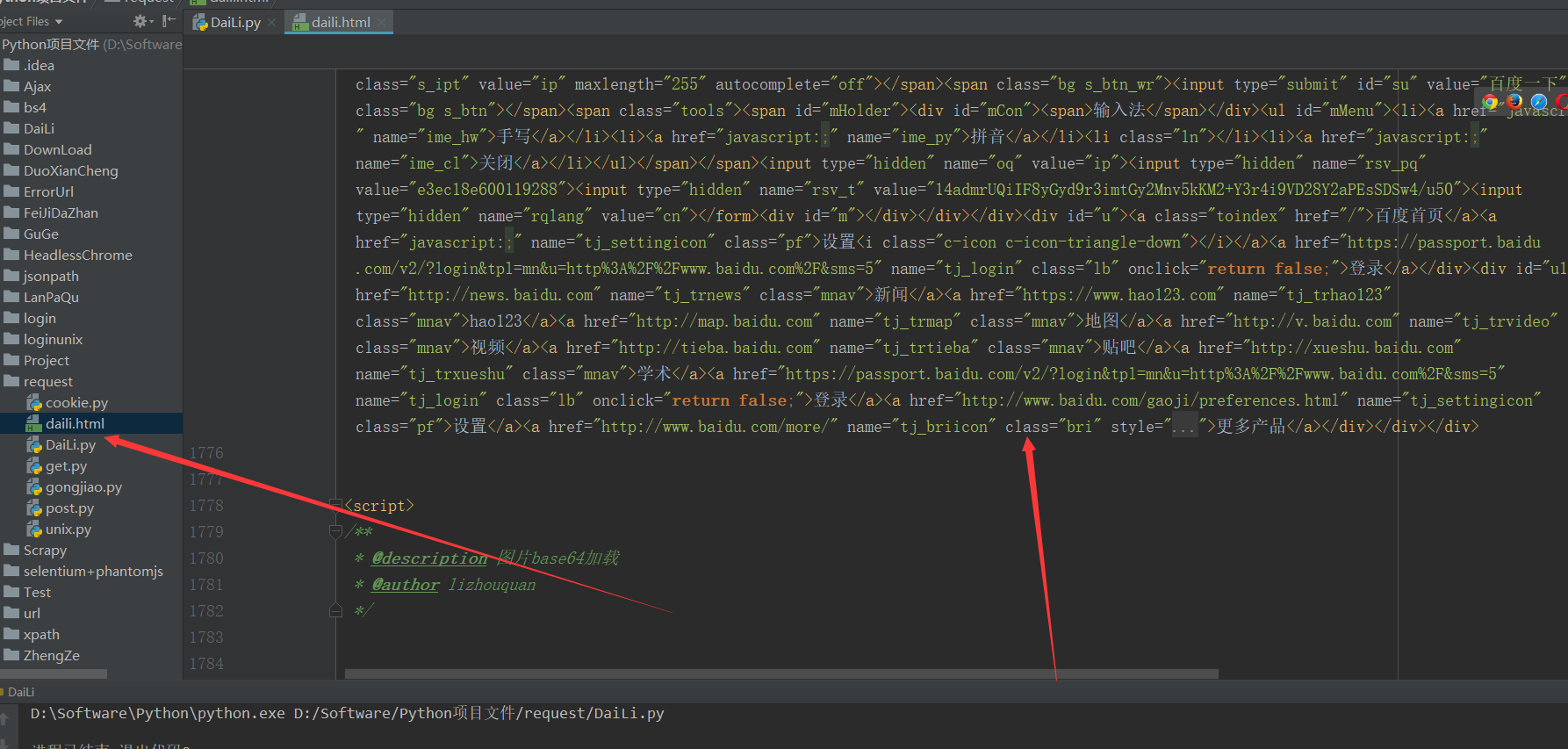

分析代码：

* 老规矩导入库和url，还有请求头
* 不一样的是，我们要写一个字典，把我们验证之后的ip传进去



* 然后就是发送请求，这个时候我们不出意外已经获取到了数据，然后就保存到本地

下面是运行结果：

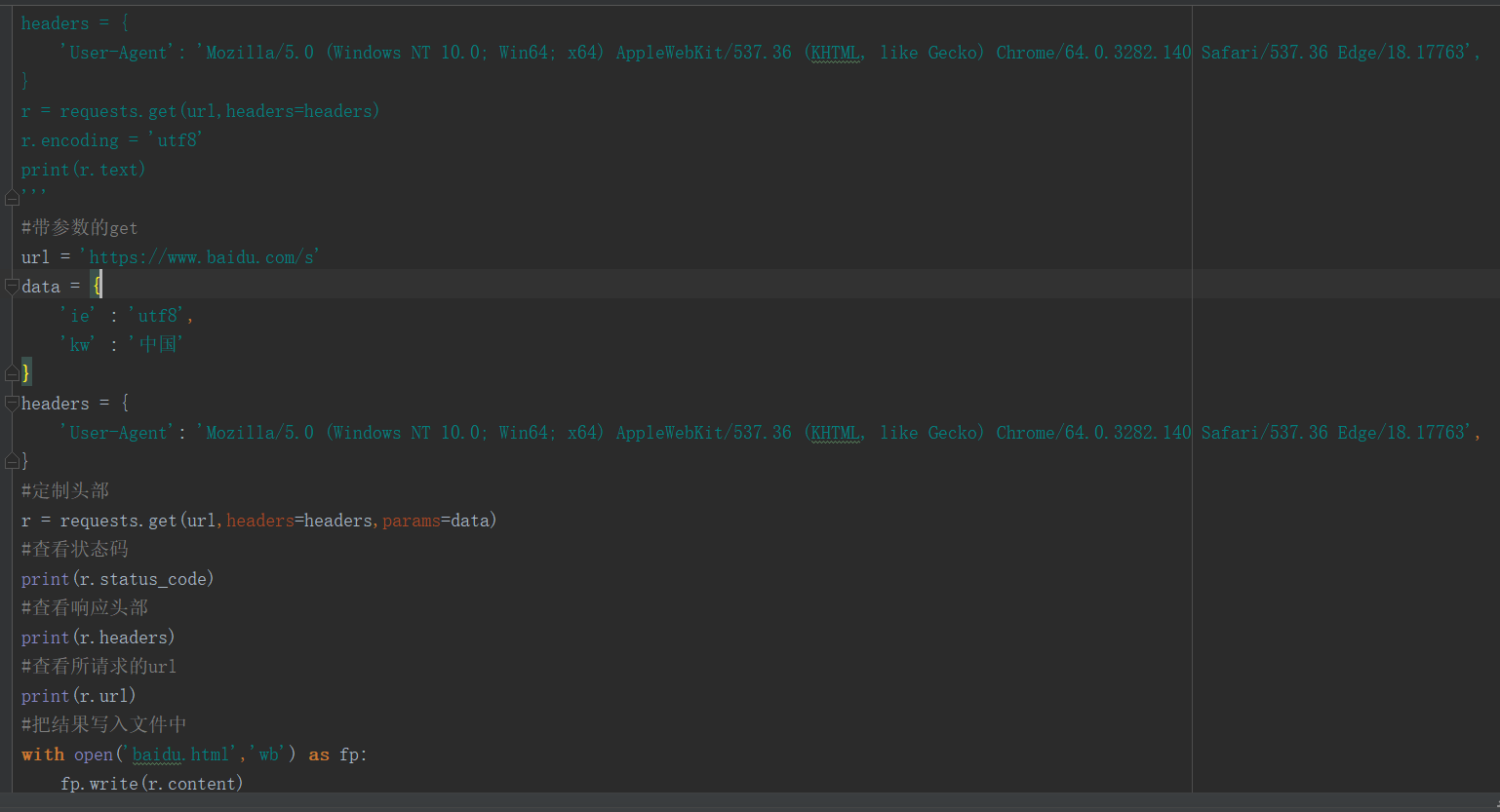


成功使用代理登录百度

1. get请求

请求有get和post，我先来get，get请求一般用于我们向服务器获取数据，我们用百度搜索一个东西，url就会跳转

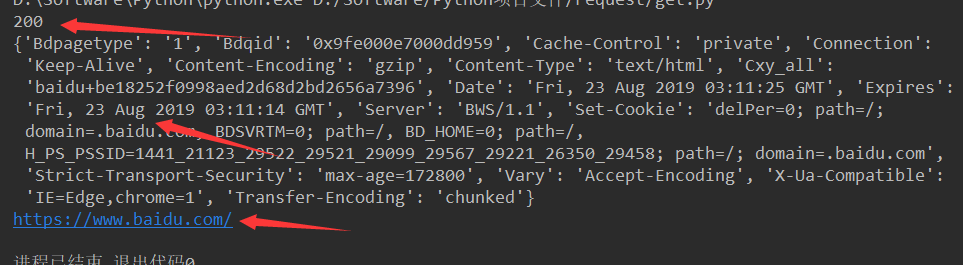
我们在这里向百度发送请求，分析后，url里面的ie就是编码格式，kw就是我们要搜索的东西，上代码分析：

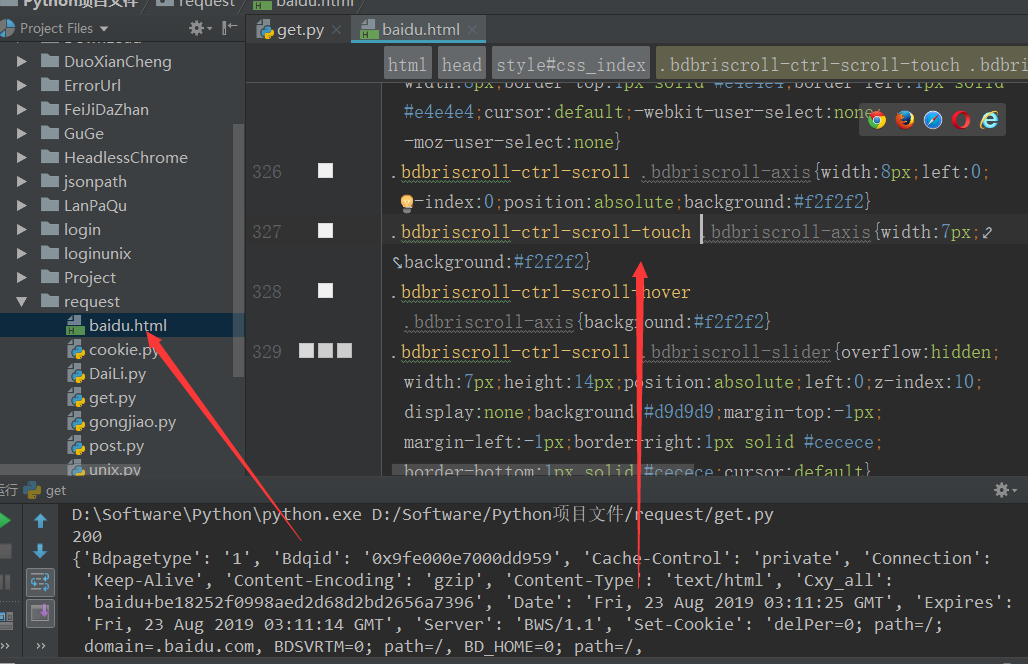


分析代码：

* 老规矩不多说
* 然后定义一个字典，里面的ie就是编码格式，我们写入utf8，kw就是我们要搜索的东西，我们暂且写中国
* 然后就是请求头，在这里我们要定制头部，以前只需要传入url和headers这次我们还要传入params参数，就是我们的data就是刚才我们的编码格式和要搜索的东西，我们在这里可以查看状态码，查看响应头部，查看所请求的url
* 然后把结果写入文件

下面是运行结果：

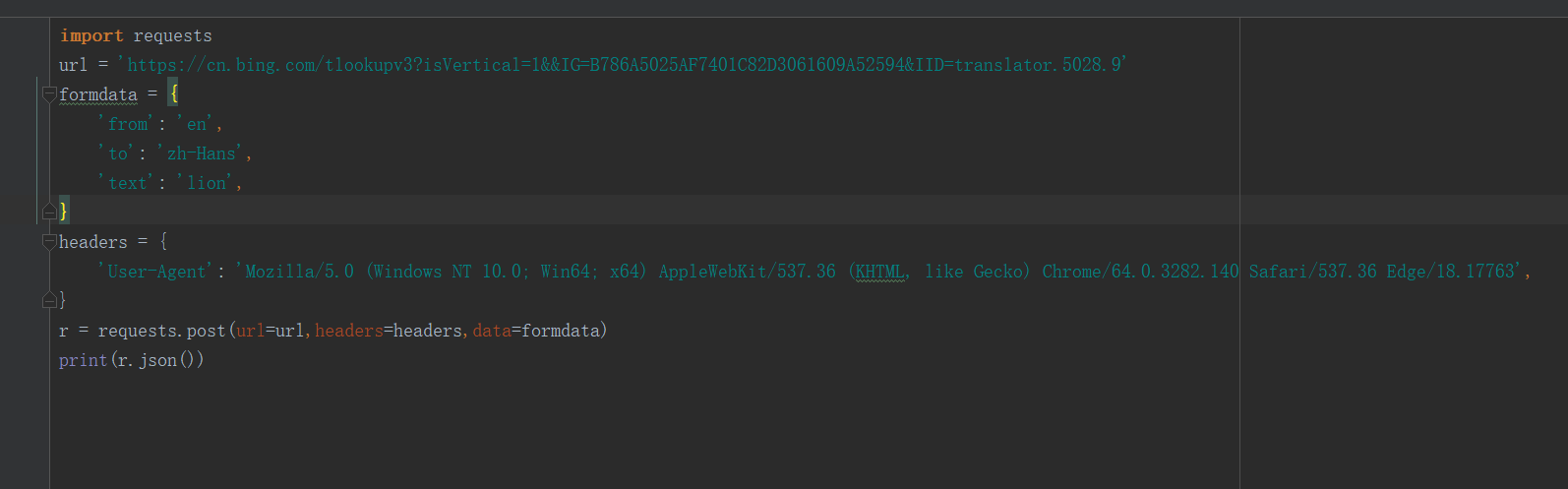




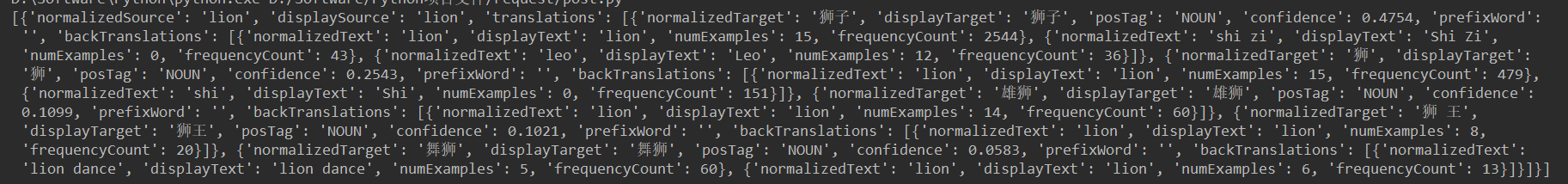
完成，其实大部分的网页都可以用get请求

1. post请求

其实post请求和get请求类型，刚才我们的get请求里面有data参数，他其实也就是可以用在post里面，我们要传送的数据就是这个参数data，data是一个字典，里面要匹配键值对，其实差不多，看代码：



这个是登陆必应翻译，我们还是老规矩请求url，然后这次是formdata和刚才的data一样，里面的text就是我们要查询的内容，不多解释，看结果，运行成功：



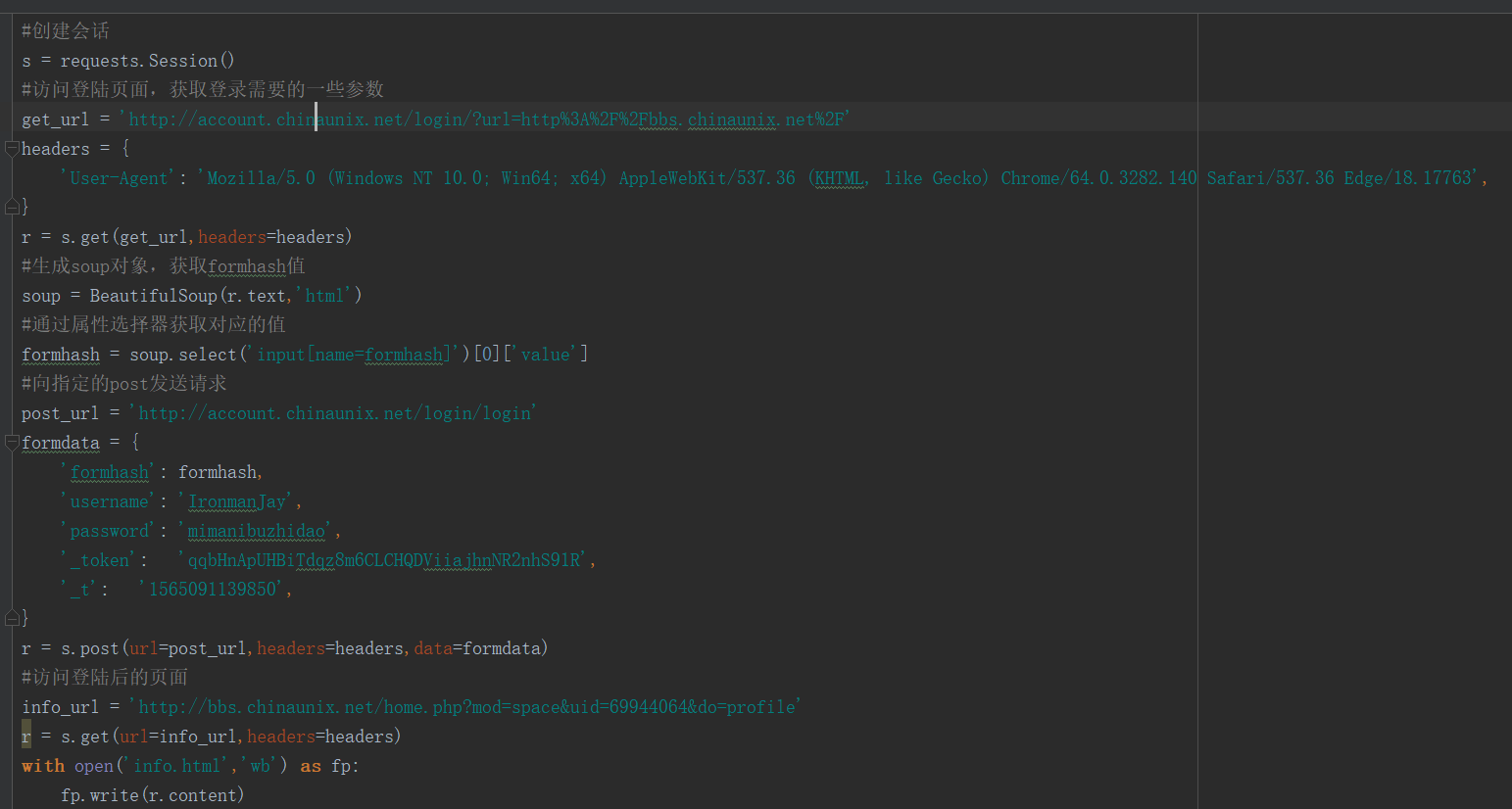
在学习这里我遇到几个问题，一个就是要特别注意headers的一些属性，我百度了一下，因为与到过

* Content-Length: 144： 是指发送的表单数据长度为144，也就是字符个数是144个。
* X-Requested-With: XMLHttpRequest ：表示Ajax异步请求。
* Content-Type: application/x-www-form-urlencoded ： 表示浏览器提交 Web 表单时使用，表单数据会按照 name1=value1&name2=value2 键值对形式进行编码。

1. 爬取chinaunix

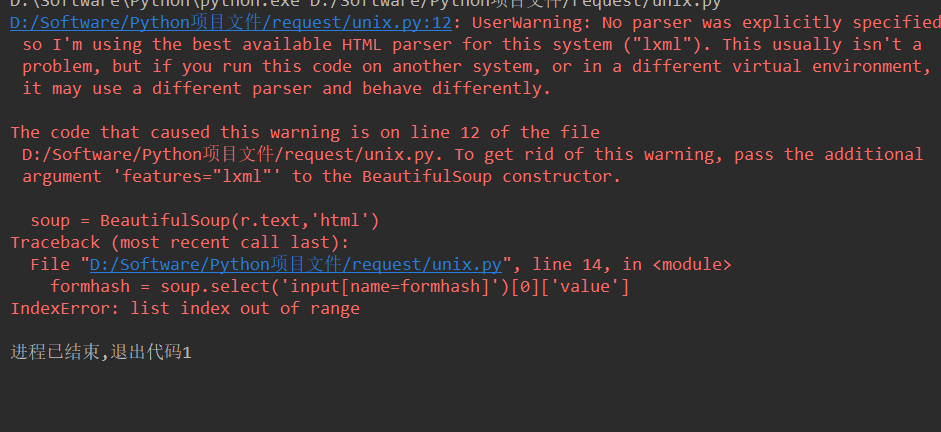
我们利用刚才学习的get和post请求来爬取这个网站，这里我们使用bs4

先看代码：



其实和以往没什么区别，爬虫的框架就那样，无非就是内容不一样，那让我们来分析代码：

* 首先，我遇到一个错误

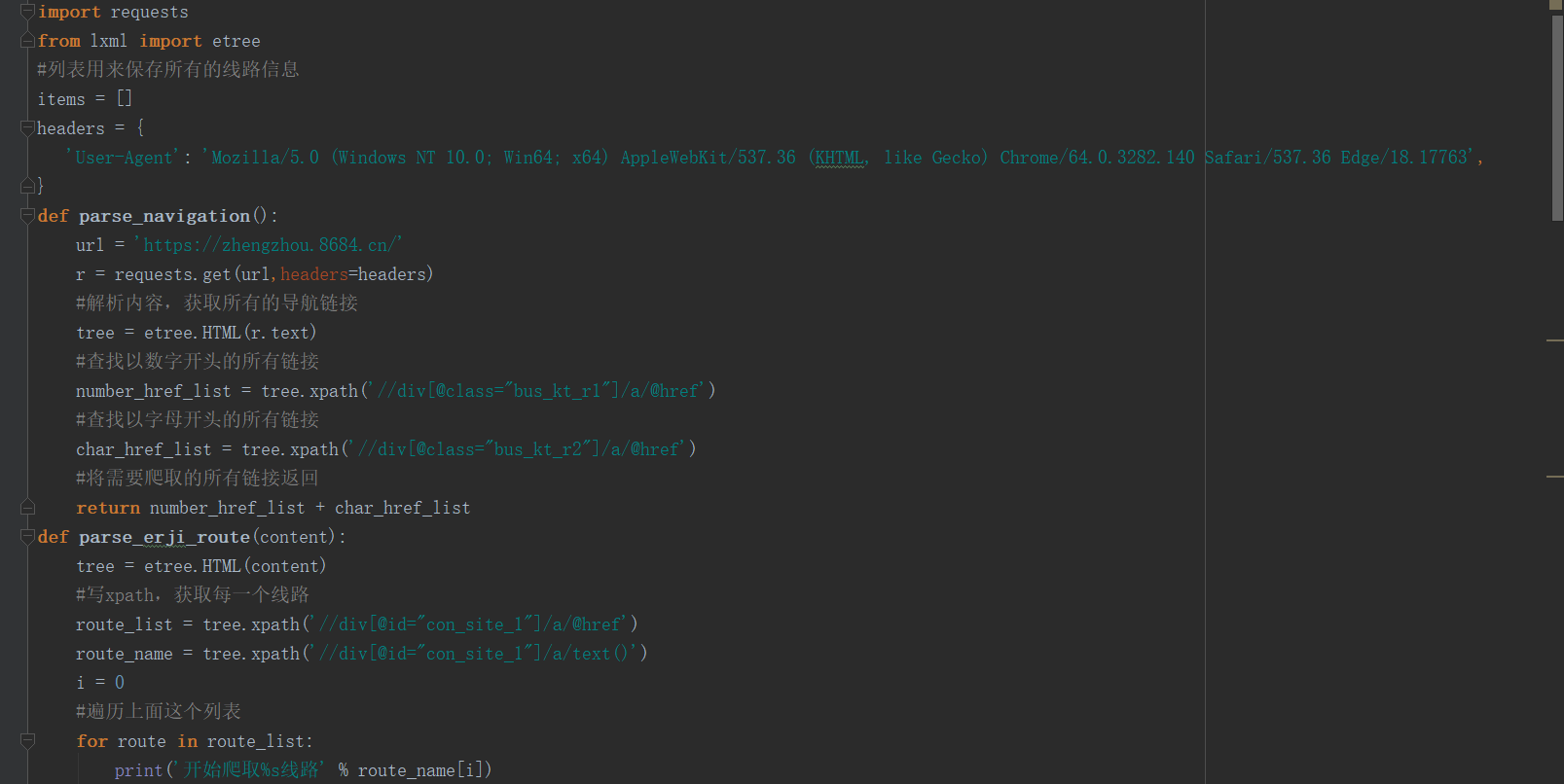


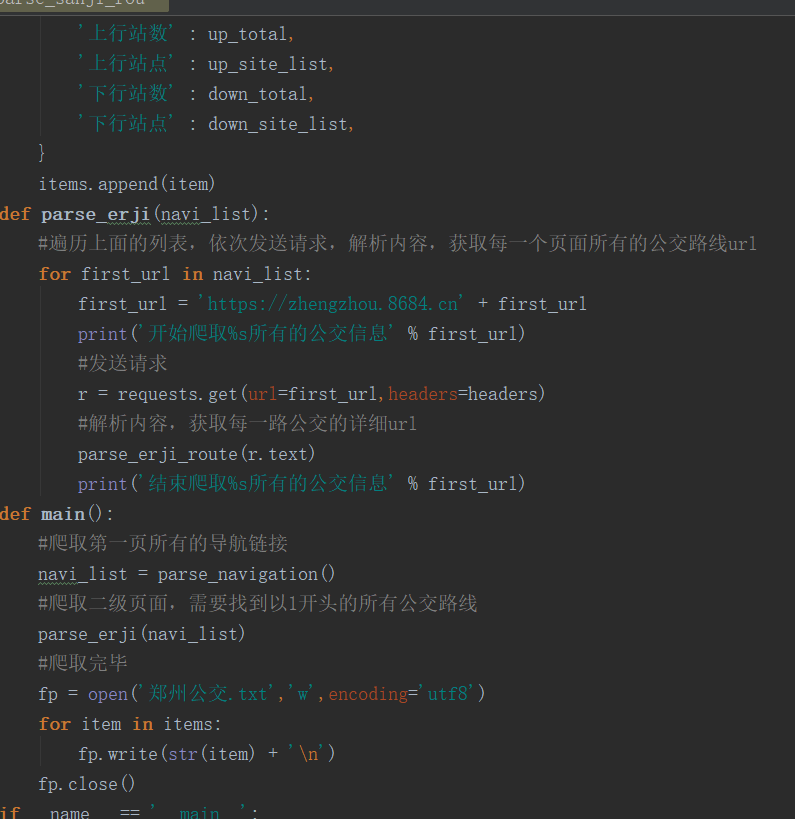
原来数组越界了，但是我取的0怎么能越界，后来我看，原来是这个列表本来就是空的，一个元素都没有，也就是我根本没获取到元素，好嘛，重新来，改了结构好了

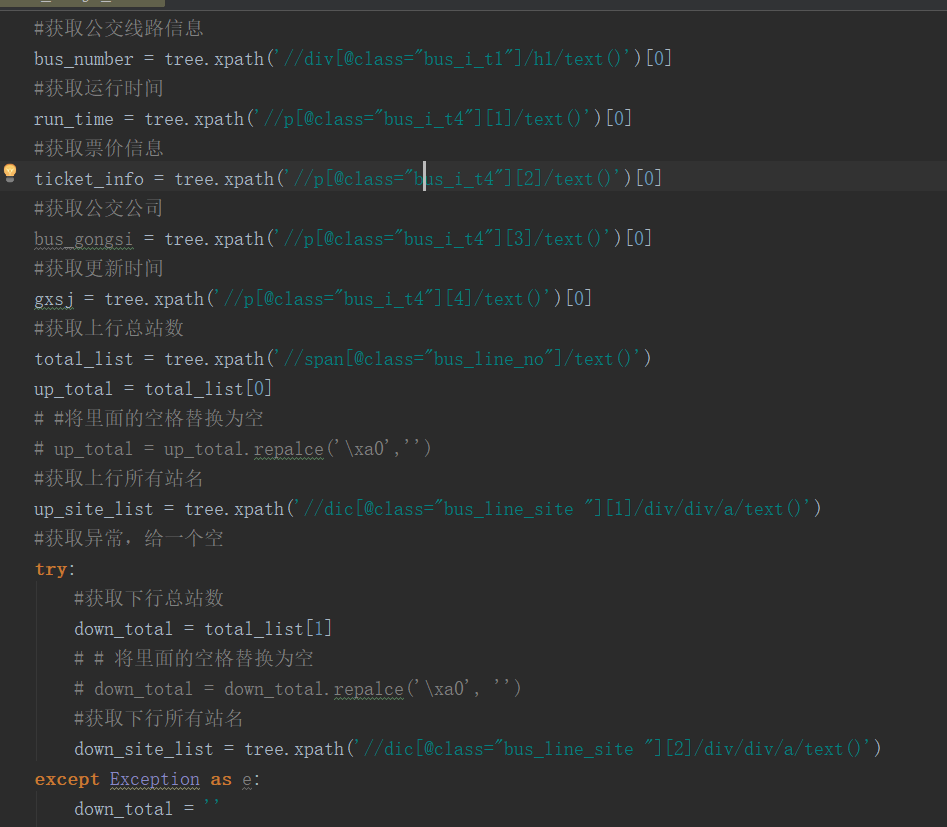
* 然后我们来看这个代码
* 先创建会话，然后访问登陆页面，获取登录需要的一些参数，我们发现这里有我们的用户名，密码等，一会说
* 然后生成soup对象，获取formhash值，这个网站不一样，用户的数据都在formhash里面
* 然后通过属性选择器获取对应的值
* 向指定的post发送请求
* 然后就是我们刚才写的字典了，用来存放用户名，密码等
* 然后构建请求发送请求
* 最后访问登陆后的页面，把获取到内容存到本地文件
* Ok

1. 最后一个案例整合了之前学的内容，爬取的是公交车网站，以前只是单单爬取一个网站页面，这次爬取的是二级页面，所以肯定要用面向对象，封装不同的类，对不同的页面发送请求，解析内容，获取内容

下面看代码然后分析：





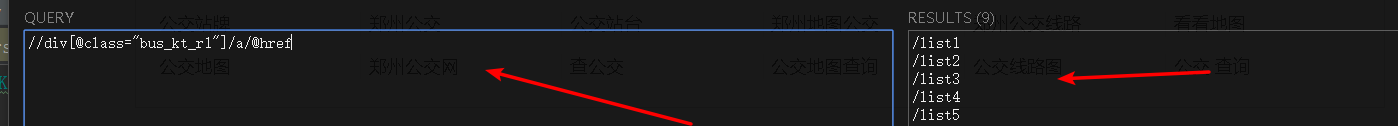


这里用的xpath，我个人更喜欢用xpath，在以后的爬虫中我也基本使用xpath。

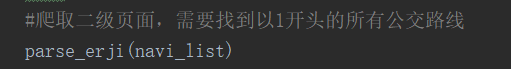
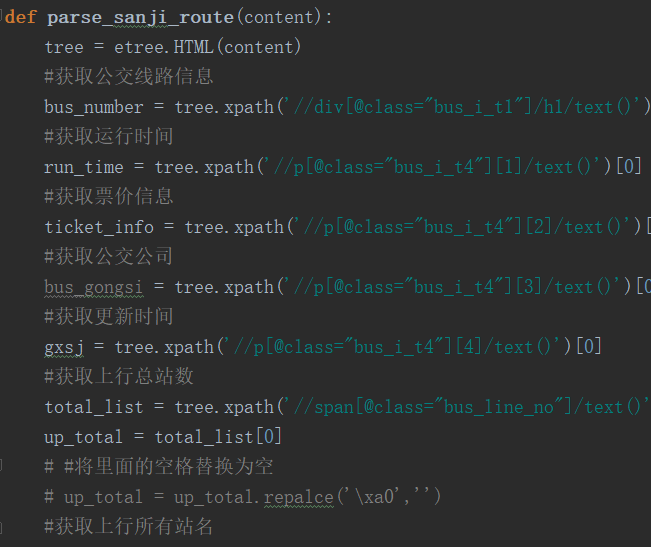
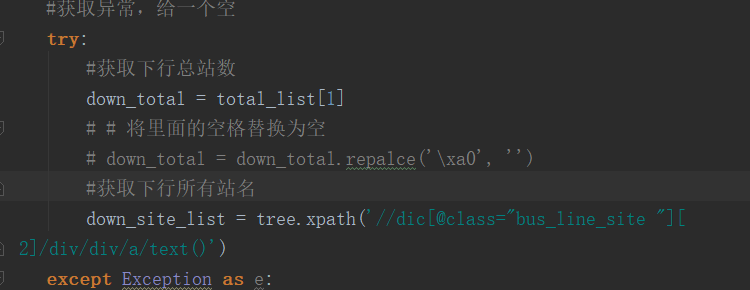
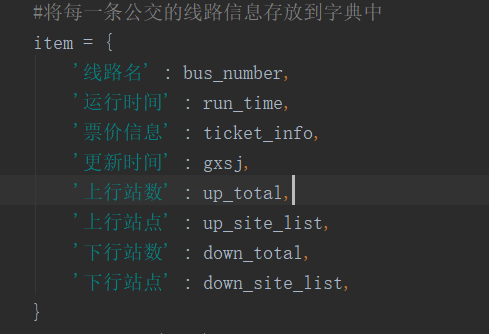
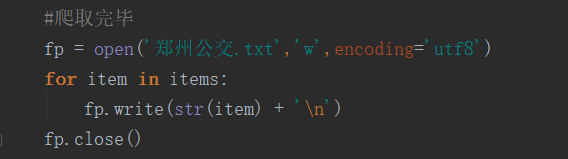
代码分析：

* 首先代码看起来很长哈，但是不要急，我们导入的库只有两个，一个是以前的requests

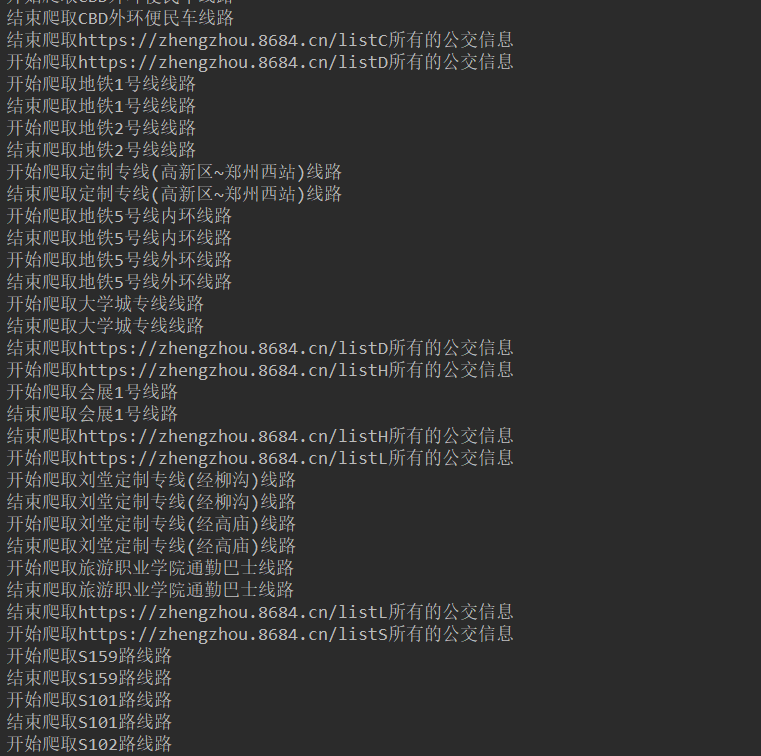
还有一个新的，from lxml import etree，这个是xpath的解析库，效率非常高

* 我们先定义主函数，里面存放各种类，第一个是爬取第一页的所有的导航链接，因为这个网站分析后发现点击之后才会到二级页面，新建类parse\_navigation()
* 好，我们来分析这个类，另外说一点，items字典和请求头都放在最前面了，因为他是全局变量，然后这个类先把网站的url写上，发送请求，好，我们已经获取到请求了，没什么好说的，最简单的爬虫，然后，解析内容，获取所有的导航链接，有好多路，有数字开头的，也有字母开头的，所以我们分别获取以数字开头的和以字母开头的，获取xpath在浏览器里就可以，对了，以前一直没说，我们需要在谷歌浏览器上安装xpath，不多解释，
* 这里就是我们需要获取的内容，

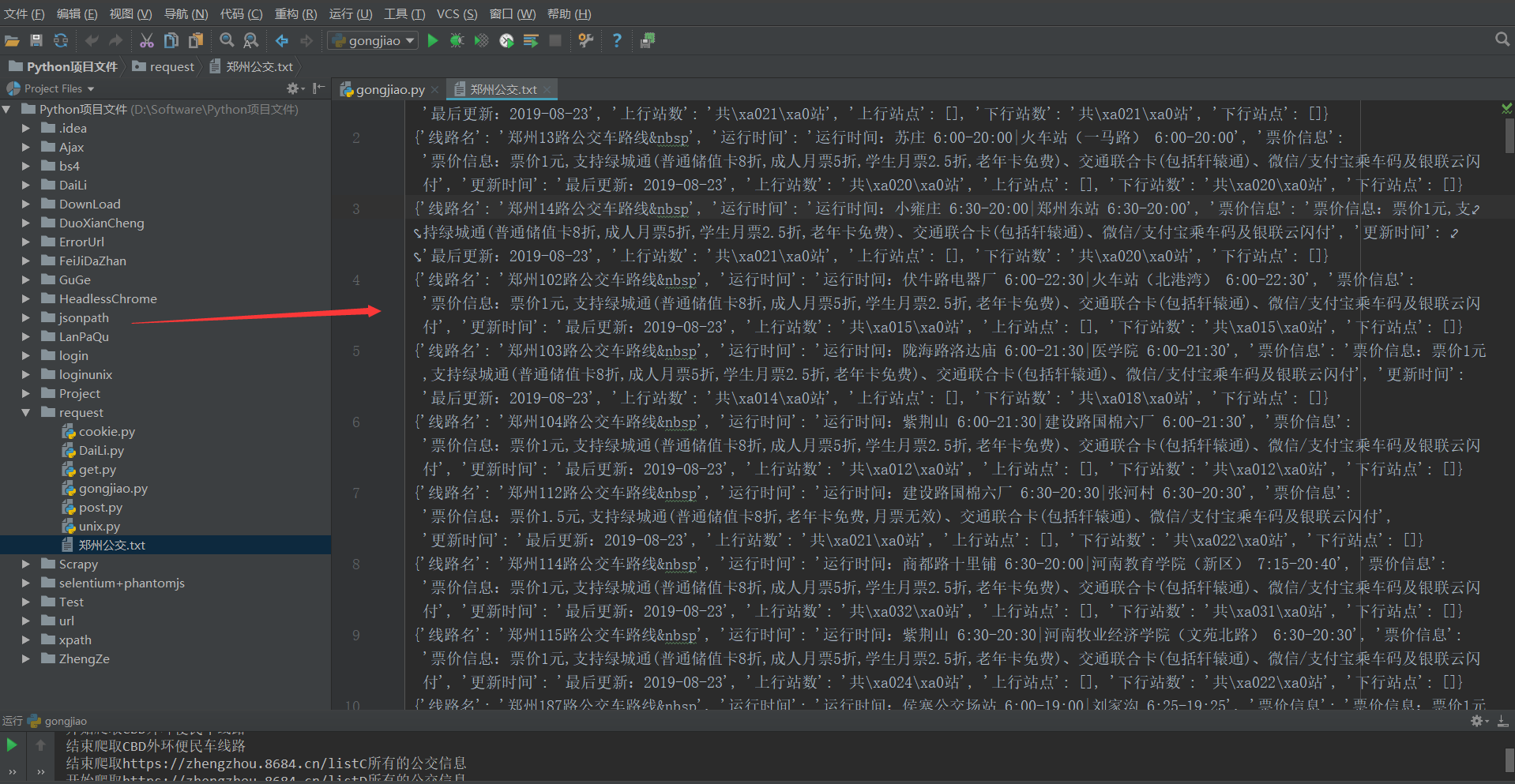
获取之后，我们需要把所有需要爬取的链接返回

* 返回到主函数用navi\_list接收，这样，我们已经获取到了一级内容，我们接下来要获取二级页面，需要新建一个类，然后把刚才的参数传进去
* 
* 来到这个类，我们首先要遍历上面的列表，依次对这些连接发送请求，解析内容，获取每一个页面的所有的公交路线url，所以需要写一个for循环，在这个for循环里面，首先要拼接url，因为缺少前面这个网站的url所以拼接之后才是我们想要的内容，然后我们再对这个url发送请求，发送完请求，我们就要解析内容，获取每一路公交的详细url，所以我们还要新建一个类来解析内容。
* 
* 来到这个类，我们已经获取到了二级页面的内容，其实这里面还有三级内容，就是每一条公交路线的内容，好，那我们就写xpath来获取每一个路线，他的名字和src，还是遍历，然后对这些内容发送请求，发送完请求还需要解析内容，这次解析的是三级页面的内容，我们要获取每一路公交的详细url，所以还要新建一个类
* 
* 来到这个类，才是我们真正要获取的内容，我们需要获取公交线路信息，运行时间，票叫信息，公交公司，更新时间，上行总站数和站名，在这里需要注意，在获取这个内容因为里面有空格，所以我们要把这些空格拿掉，然后就是下行路线，我们在这里还要注意，因为有的公交车没有下行，所以获取内容可能为空，会报错，所以我们在这里抛一个异常
* 
* 然后我们就获取到所有内容，这时我们需要新建一个字典，刚才不是建立了吗，刚才的是总字典，要把这个字典放进去，
* 然后我们存放到最开始建立的字典，因为他是一个全局变量嘛，然后在主函数里面，新建一个txt，用来存放这个字典，也就是我们爬取的内容
* 最后别忘了关闭
* Ps：这个网站信息太多了，爬了好久，反正爬出来了哈哈

下面是运行过程：



下面是程序运行结果：



大功告成！