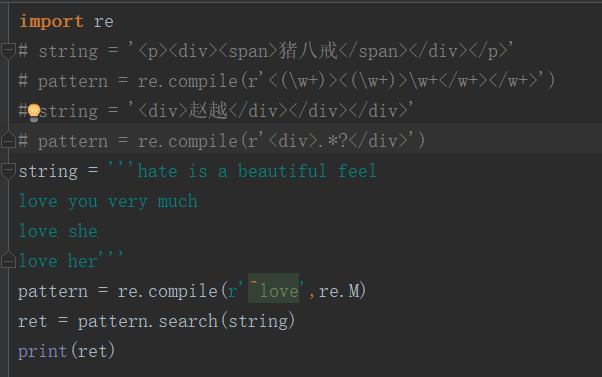
**正则**

今天我们来学习正则，其实在java基础阶段，和python基础阶段，还有linux里面，我们已经学习了正则，是可以获取内容的，同样我们在爬虫里也可以使用，之前我们学习的xpath，也是用来获取内容的，和我们的用法和功能大同小异，我们先来看看正则的用法，我在网上找了一个图片



这个就是正则的基本用法，废话不多说，我们先来练习一下

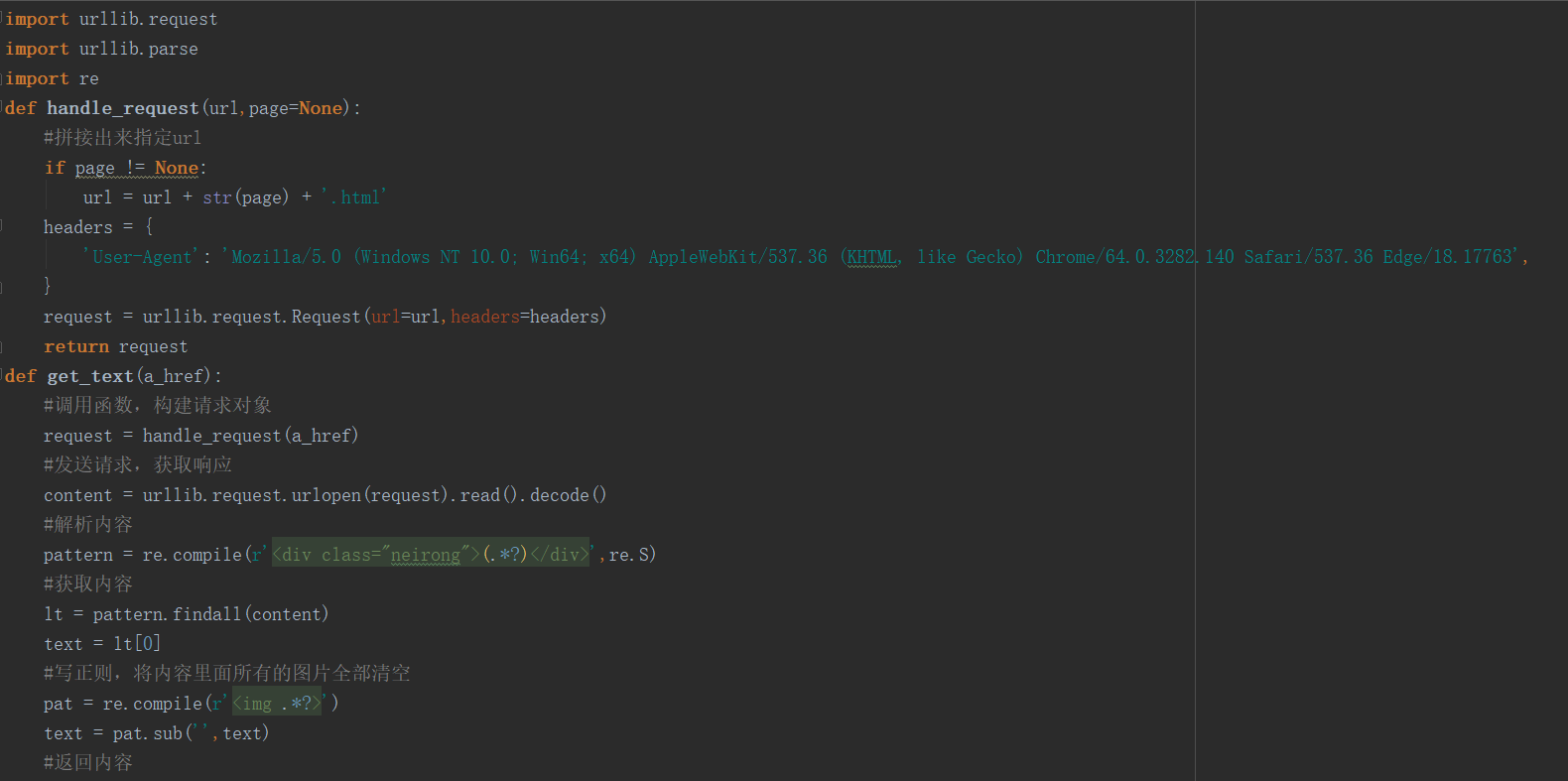


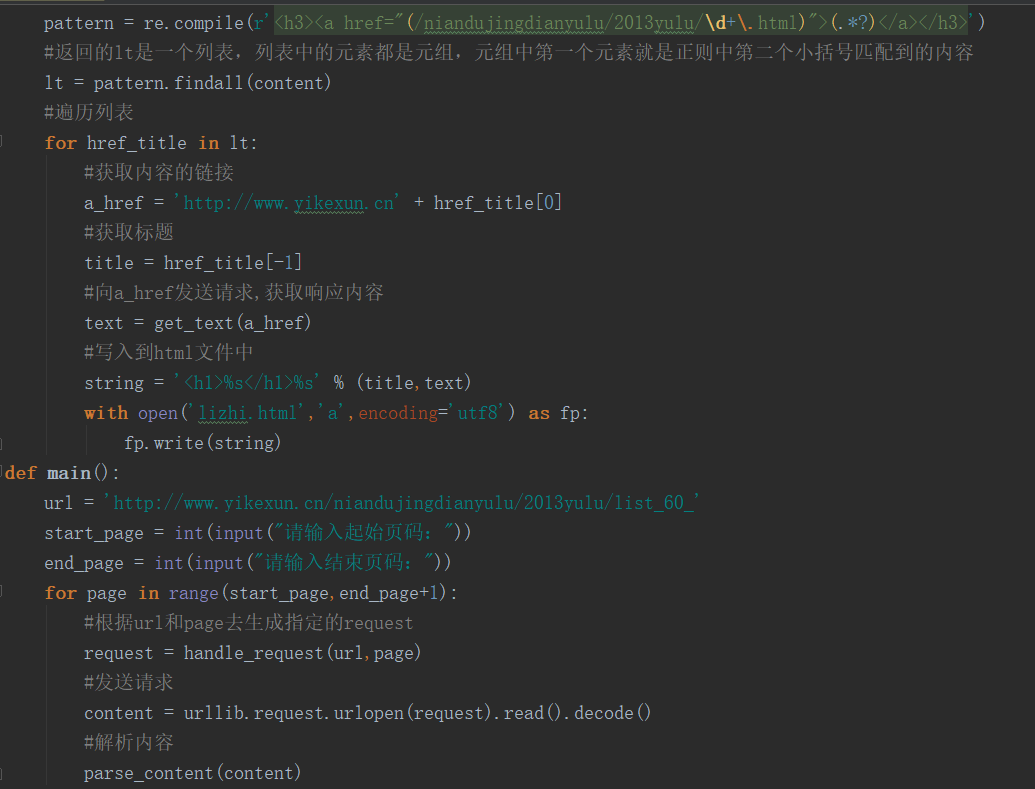
这是我们联系的代码，我们需要获取内容，首先使用正则我们需要导入re库，在这个字符串里面，我们需要获取带有love的句子，我们用re的用法，来看运行结果：



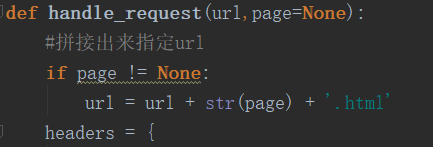
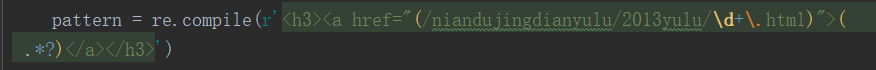
成功获取到

那么接下来我们做两个实例，第一个爬取的是励志的文字，其实和xpath一样，我们都要检查网页，然后获取到我们需要爬取内容的位置，写re表达式来获取内容，我们在这里同样需要实现用户手动输入爬取页面，其实我之前说过，所有的爬虫都大同小异，整体的框架摆在那里，我们只需要利用不同的方法获取到这些内容而已，我们先来看代码：

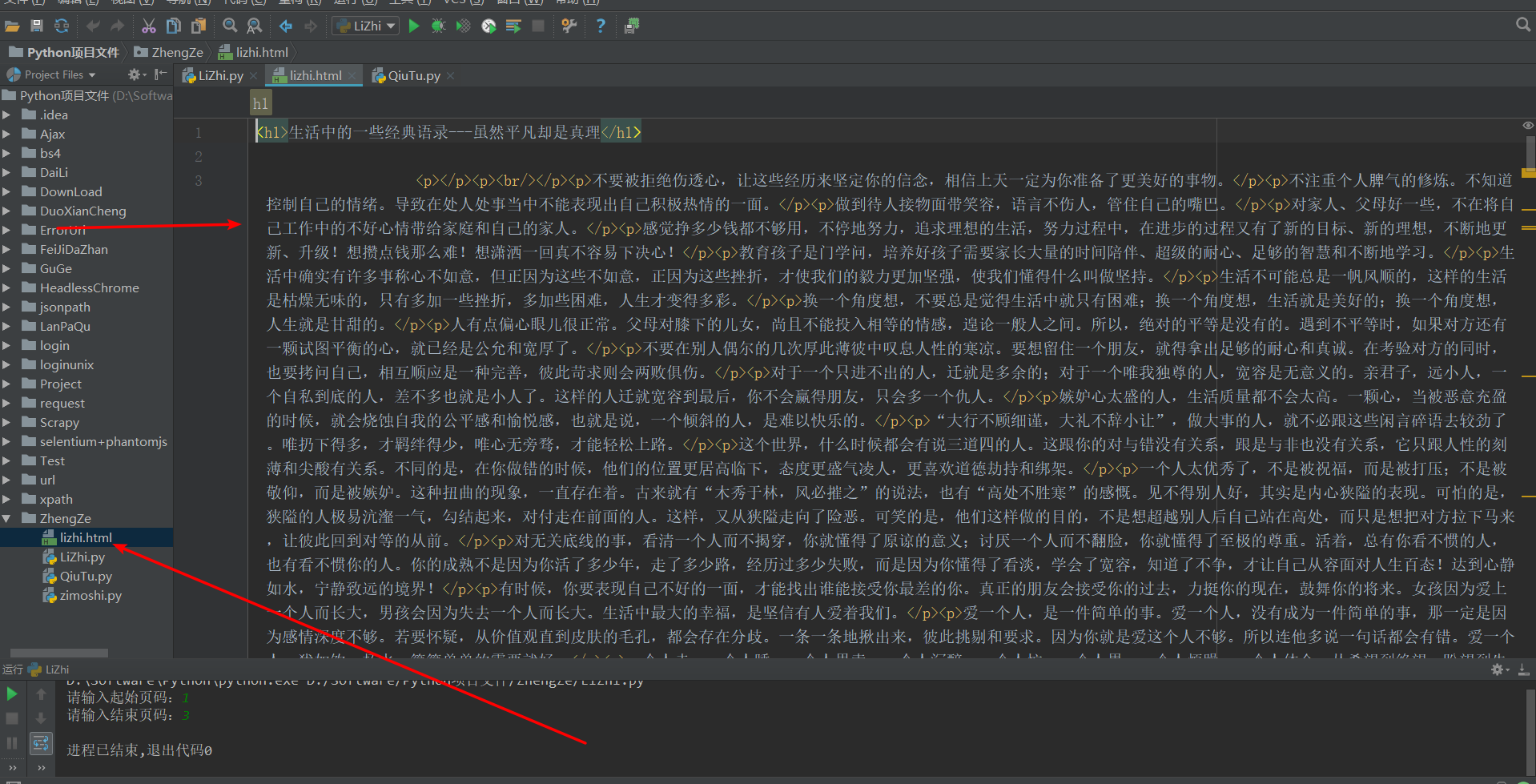




分析代码：

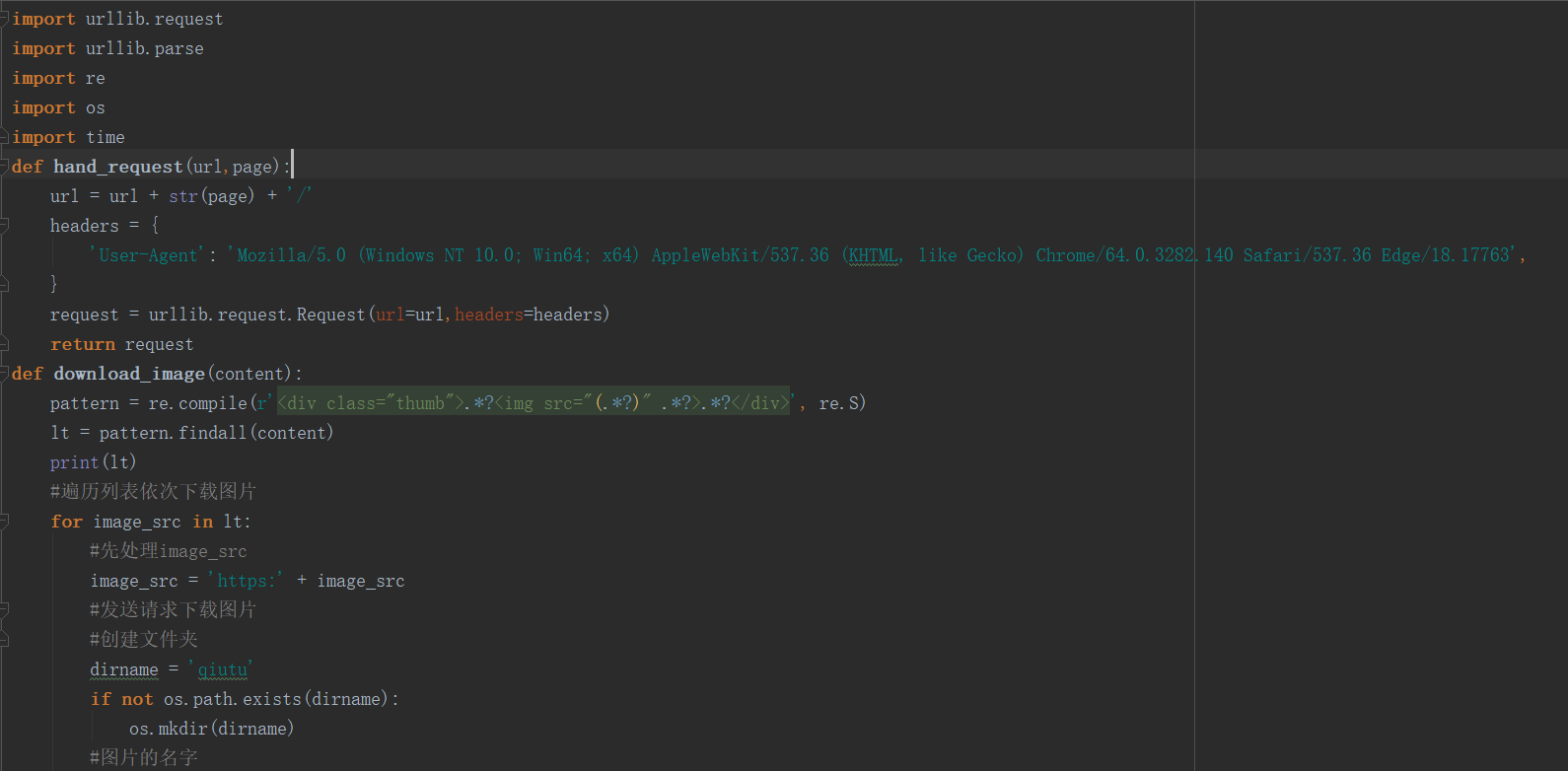
* 还是先定义主函数，在这里是我们爬取的主程序
* 导入所需要的库，写上url，还有提醒用户手动输入页数
* 然后我们遍历这些页数，在这个循环里面，我们根据url和page去省城指定的request
* 我们需要新建一个函数用来专门发送请求和拼接url，刚才的页数和url我们当作参数传进去
* 我们来到这个类，我们需要判断我们输入的page在这个网站有没有数据，所以我们需要判断page是否为空
* 
* 如果不为空我们拼接出来指定的url
* 然后就是请求头，发送请求，最后把请求返回到主函数
* 我们在这里已经向这个网站发送了请求，现在我们就要获取响应内容
* 
* 不出意外我们已经获取到了相应内容，所以我们要来解析内容，我们还是新建一个函数，在这里面解析内容
* 我们来到这个函数
* 既然是解析，所以我们要提取内容，在这里使用正则
* 
* 这时返回的lt是一个列表，列表中的元素都是元组，元组中的第一个元素就是正则中第二个小括号匹配到的内容
* 现在我们遍历这个列表，首先获取内容的链接，然后获取标题，因为这是一个二级请求内容，所以我们要获取href然后向这个href发送请求，获取相应内容，我们要获取href里面的内容，所以需要新建一个函数，在这里面获取，把刚才获取内容的链接，当作参数传递进去
* 我们来到这个函数，这个就好比我们之前做的爬取公交路线的东西，有好几级内容，所以我们需要调用之前的请求函数，构建请求对象，然后就是向这个href发送请求，获取相应，再解析内容，为了不让函数太多，直接解析
* 
* 这个时候我们已经获取到了内容，因为有的文字里面有图片，我们没写获取图片的代码，所以写一个正则，将内容里面所有的图片全部清空
* 
* 最后返回这些内容
* 回到解析函数，我们已经获取到了内容，最后保存到本地就ok了

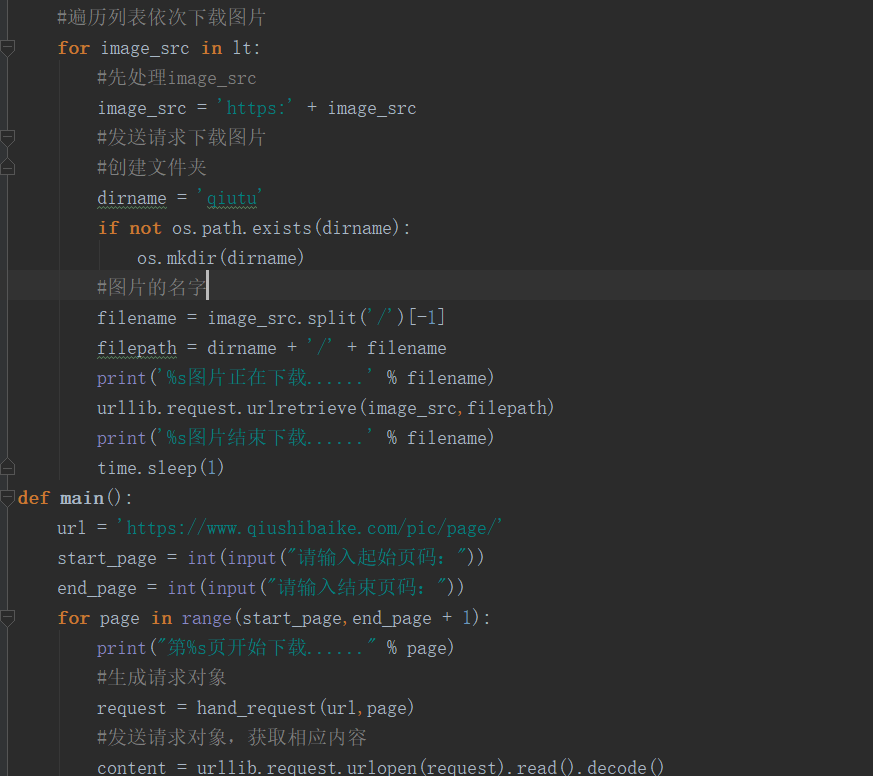
我们来看看运行结果：



完美！

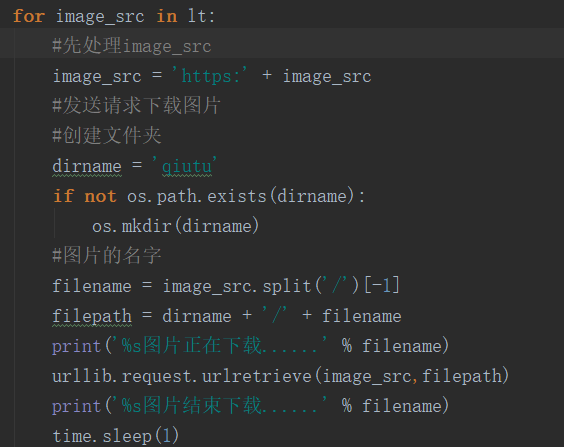
光爬取文字不够爽，我们第二个就来爬取图片，直接上代码



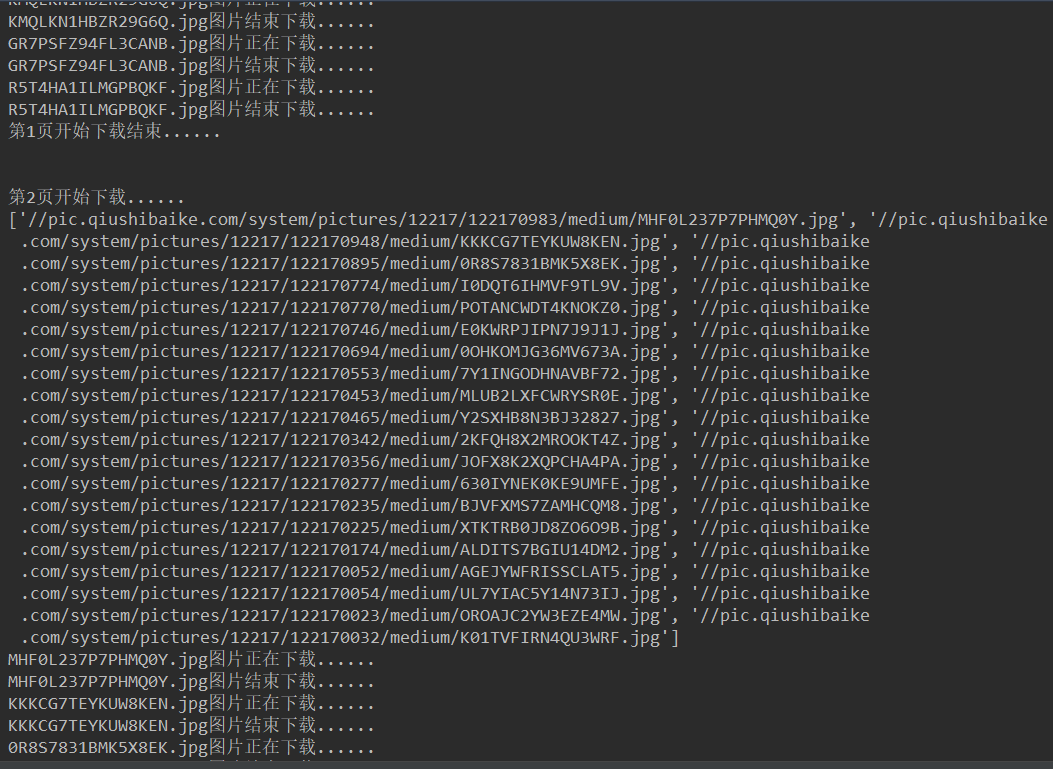




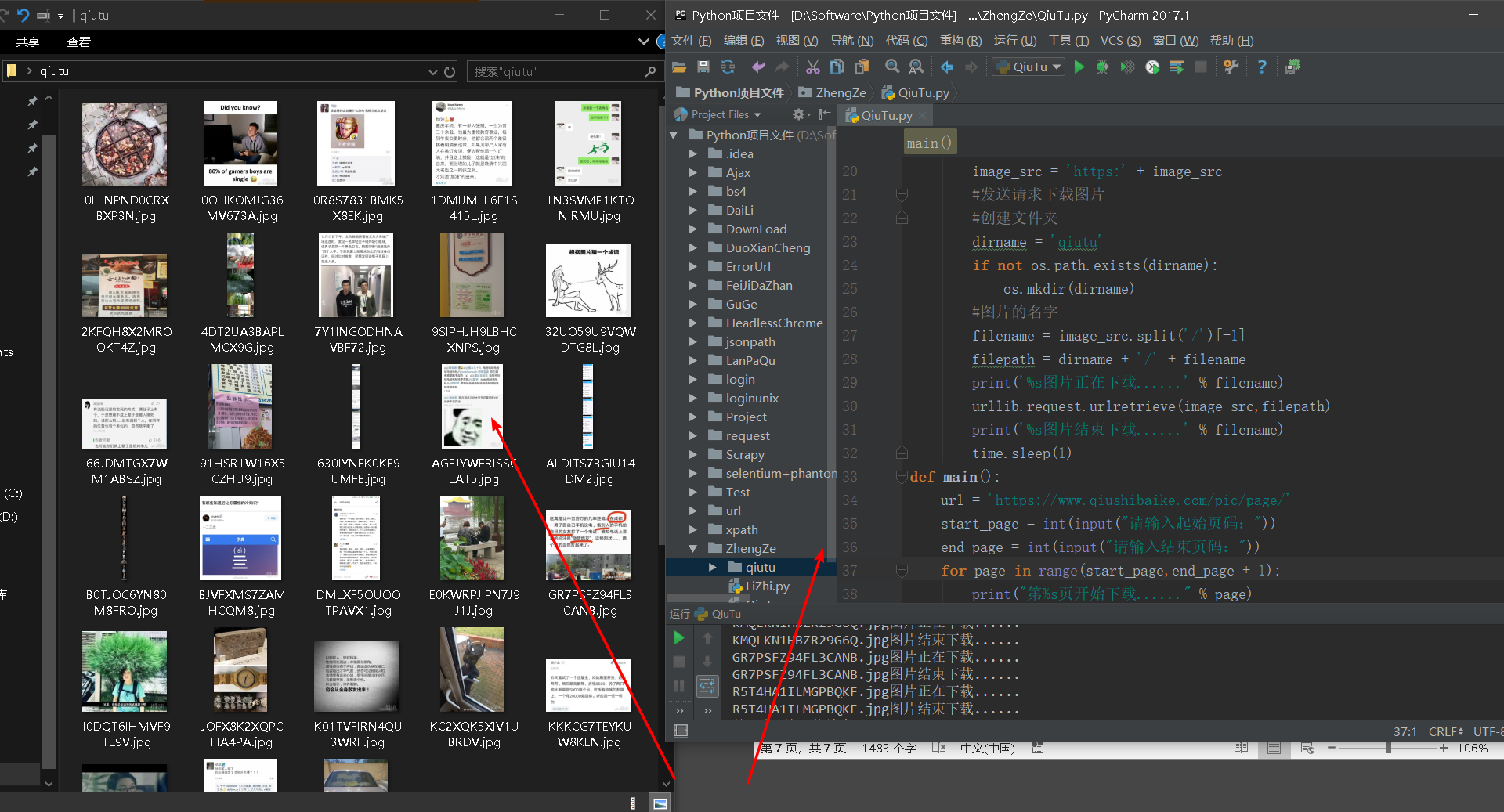
其实和之前的做法差不多，只是我们新增了，提示爬到哪页了，写正则的方法不太一样了，别的没什么不一样的，我们来分析代码：

* 首先导入库，然后是主函数，在这里是我们爬虫的主程序，输入url和页码
* 然后遍历这些页码，在这个循环里面我们print一下从第几页开始下载，和结束下载
* 还是老规矩，我们生成请求对象，还是新建一个函数，专门用来发送请求，把页码和url当作参数传递进去
* 我们来到这个函数
* 还是拼接url和指定的页码，写上请求头，向指定之后的url发送请求，最后把请求返回到主函数
* 来到主函数，我们发送请求对象，获取响应内容
* 
* 然后我们解析内容，提取所有的图片链接，下载图片，我们还要新建一个函数，专门用来解析内容
* 来到这个类和刚才一样写正则，获取内容，现在我们想要的内容都在这个列表里面，所以我们遍历列表，依次下载图片
* 
* 最后保存到本地

我们来看看运行过程：



我们再来看看运行结果：



大功告成！