Python网络爬虫笔记（一）：网页抓取方式和LXML示例

（一）   三种网页抓取方法

1、    正则表达式：

模块使用C语言编写，速度快，但是很脆弱，可能网页更新后就不能用了。

2、    Beautiful Soup

模块使用Python编写，速度慢。

安装：

pip install beautifulsoup4

3、    Lxml

模块使用C语言编写，即快速又健壮，通常应该是最好的选择。

（二）   Lxml安装

pip install lxml

如果使用lxml的css选择器，还要安装下面的模块

pip install cssselect

（三）   使用lxml示例

1 import urllib.request as re  
 2 import lxml.html  
 3 #下载网页并返回HTML  
 4 def download(url,user\_agent='Socrates',num=2):  
 5 print('下载:'+url)  
 6 #设置用户代理  
 7 headers = {'user\_agent':user\_agent}  
 8 request = re.Request(url,headers=headers)  
 9 try:  
10 #下载网页  
11 html = re.urlopen(request).read()  
12 except re.URLError as e:  
13 print('下载失败'+e.reason)  
14 html=None  
15 if num>0:  
16 #遇到5XX错误时，递归调用自身重试下载，最多重复2次  
17 if hasattr(e,'code') and 500<=e.code<600:  
18 return download(url,num=num-1)  
19 return html  
20 html = download('https://tieba.baidu.com/p/5475267611')  
21 #将HTML解析为统一的格式  
22 tree = lxml.html.fromstring(html)  
23 # img = tree.cssselect('img.BDE\_Image')  
24 #通过lxml的xpath获取src属性的值，返回一个列表  
25 img = tree.xpath('//img[@class="BDE\_Image"]/@src')  
26 x= 0  
27 #迭代列表img,将图片保存在当前目录下  
28 for i in img:  
29 re.urlretrieve(i,'%s.jpg'%x)  
30 x += 1