Python网络爬虫笔记（三）：下载博客园随笔到Word文档

（一）   说明

在上一篇的基础上修改了下，使用lxml提取博客园随笔正文内容，并保存到Word文档中。

操作Word文档会用到下面的模块：

pip install python-docx

修改的代码（主要是在link\_crawler()的while循环中增加了下面这段）

（二）   完整代码（delayed.py的代码就不贴出来了，和上一篇一样）

限速最好设置大一些 ，下面这句，以秒为单位。

（三）结果

（四）存在的问题

 （1）代码部分是添加到正文内容后面的。（使用过博客园插入代码功能的随笔，排版会不一致）

（2）图片是直接插入到代码部分后面的。（随笔有插入图片的，排版会不一致）

1 tree = lxml.html.fromstring(html) #解析HTML为统一的格式  
 2 title = tree.xpath('//a[@id="cb\_post\_title\_url"]') #获取标题  
 3 the\_file = tree.xpath('//div[@id="cnblogs\_post\_body"]/p') #获取正文内容  
 4 pre = tree.xpath('//pre') #获取随笔代码部分（使用博客园自带插入代码功能插入的）  
 5 img = tree.xpath('//div[@id="cnblogs\_post\_body"]/p/img/@src') #获取图片  
 6 #修改工作目录  
 7 os.chdir('F:\Python\worm\博客园文件')  
 8 #创建一个空白新的Word文档  
 9 doc = docx.Document()  
10 #添加标题  
11 doc.add\_heading(title[0].text\_content(), 0)  
12 for i in the\_file:  
13 #将每一段的内容添加到Word文档（p标签的内容）  
14 doc.add\_paragraph(i.text\_content())  
15 # 将代码部分添加到文档中  
16 for p in pre:  
17 doc.add\_paragraph(p.text\_content())  
18 #将图片添加到Word文档中  
19 for i in img:  
20 ure.urlretrieve(i, '0.jpg')  
21 doc.add\_picture('0.jpg')  
22 #截取标题的前8位作为Word文件名  
23 filename = title[0].text\_content()[:8] + '.docx'  
24 #保存Word文档  
25 #如果文件名已经存在，将文件名设置为title[0].text\_content()[:8]+ str(x).docx，否则将文件名设置为filename  
26 if str(filename) in os.listdir('F:\Python\worm\博客园文件'):  
27 doc.save(title[0].text\_content()[:8] + str(x) + '.docx')  
28 x += 1  
29 else:  
30 doc.save(filename)

waitFor = WaitFor(2)

1 import urllib.request as ure  
 2 import re  
 3 import urllib.parse  
 4 from delayed import WaitFor  
 5 import lxml.html  
 6 import os  
 7 import docx  
 8 #下载网页并返回HTML(动态加载的部分下载不了)  
 9 def download(url,user\_agent='FireDrich',num=2):  
10 print('下载:'+url)  
11 #设置用户代理  
12 headers = {'user\_agent':user\_agent}  
13 request = ure.Request(url,headers=headers)  
14 try:  
15 #下载网页  
16 html = ure.urlopen(request).read()  
17 except ure.URLError as e:  
18 print('下载失败'+e.reason)  
19 html=None  
20 if num>0:  
21 #遇到5XX错误时，递归调用自身重试下载，最多重复2次  
22 if hasattr(e,'code') and 500<=e.code<600:  
23 return download(url,num-1)  
24 return html  
25 #seed\_url传入一个url，例如https://www.cnblogs.com/  
26 #link\_regex传入一个正则表达式  
27 #函数功能：提取和link\_regex匹配的所有网页链接并下载  
28 def link\_crawler(seed\_url, link\_regex):  
29 html = download(seed\_url)  
30 crawl\_queue = []  
31 #迭代get\_links（）返回的列表，将匹配正则表达式link\_regex的链接添加到列表中  
32 for link in get\_links(html):  
33 if re.match(link\_regex, link):  
34 #拼接https://www.cnblogs.com/ 和 /cate/...  
35 link = urllib.parse.urljoin(seed\_url, link)  
36 #不在列表中才添加  
37 if link not in crawl\_queue:  
38 crawl\_queue.append(link)  
39 x = 0  
40 #调用WaitFor的wait（）函数，下载限速，间隔小于2秒则等待,直到间隔等于2秒才继续下载（大于5秒则直接继续下载）  
41 waitFor = WaitFor(2)  
42 #下载crawl\_queue中的所有网页  
43 while crawl\_queue:  
44 #删除列表末尾的数据  
45 url = crawl\_queue.pop()  
46 waitFor.wait(url)  
47 html = download(url)  
48 tree = lxml.html.fromstring(html) #解析HTML为统一的格式  
49 title = tree.xpath('//a[@id="cb\_post\_title\_url"]') #获取标题  
50 the\_file = tree.xpath('//div[@id="cnblogs\_post\_body"]/p') #获取正文内容  
51 pre = tree.xpath('//pre') #获取随笔代码部分（使用博客园自带插入代码功能插入的）  
52 img = tree.xpath('//div[@id="cnblogs\_post\_body"]/p/img/@src') #获取图片  
53 #修改工作目录  
54 os.chdir('F:\Python\worm\博客园文件')  
55 #创建一个空白新的Word文档  
56 doc = docx.Document()  
57 #添加标题  
58 doc.add\_heading(title[0].text\_content(), 0)  
59 for i in the\_file:  
60 #将每一段的内容添加到Word文档（p标签的内容）  
61 doc.add\_paragraph(i.text\_content())  
62 # 将代码部分添加到文档中  
63 for p in pre:  
64 doc.add\_paragraph(p.text\_content())  
65 #将图片添加到Word文档中  
66 for i in img:  
67 ure.urlretrieve(i, '0.jpg')  
68 doc.add\_picture('0.jpg')  
69 #截取标题的前8位作为Word文件名  
70 filename = title[0].text\_content()[:8] + '.docx'  
71 #保存Word文档  
72 #如果文件名已经存在，将文件名设置为title[0].text\_content()[:8]+ str(x).docx，否则将文件名设置为filename  
73 if str(filename) in os.listdir('F:\Python\worm\博客园文件'):  
74 doc.save(title[0].text\_content()[:8] + str(x) + '.docx')  
75 x += 1  
76 else:  
77 doc.save(filename)  
78 #传入html对象，以列表形式返回所有链接  
79 def get\_links(html):  
80 #使用正则表达式提取html中所有网页链接  
81 webpage\_regex = re.compile('<a[^>]+href=["\'](.\*?)["\']',re.IGNORECASE)  
82 html = html.decode('utf-8')  
83 # 以列表形式返回所有网页链接  
84 return webpage\_regex.findall(html)  
85   
86 link\_crawler('https://www.cnblogs.com/cate/python/','.\*/www.cnblogs.com/.\*?\.html$')





