专栏：007：xpath使用及其实战

E – 爬虫技术 爬虫技术

> 系列爬虫专栏
>
> 崇尚的学习思维是：输入，输出平衡，且平衡点不断攀升。
>
> 曾经有大神告诫说：没事别瞎写文章；所以，很认真的写的是能力范围内的，看客要是看不懂，不是你的问题，问题在我，得持续输入，再输出。
>
> 今天的主题是：xpath的使用及其当当心理学图书抓取

1：框架

序号	内容	说明
01	概念	–
02	xpath语法	–
03	语法实例	–
04	实战当当心理学图书抓取	–
05	参考及总结	–

2：概念

- Xpath
  XPath一门在 XML 文档中查找信息的语言。XPath即为XML路径语言（XML Path Language），它是一种用来确定XML文档中某部分位置的语言。
  XPath基于XML的树状结构，提供在数据结构树中找寻节点的能力。起初XPath的提出的初衷是将其作为一个通用的、介于XPointer与XSL间的语法模型。但是XPath很快的被开发者采用来当作小型查询语言。

复述：相当于定位地址，比如，我要找清华大学在哪：30 Shuangqing Rd, Haidian, Beijing, China 。先定位在中国，再定位在北京，再定位在海淀区，继续定位具体的街道地址。
那比如你定位到北京：那北京很多区。
那再比如你定位到海淀区：那海淀区也分很多地方。

在 XPath 中，有七种类型的节点：元素、属性、文本、命名空间、处理指令、注释以及文档节点（或称为根节点）。(解析就是对这些节点进行定位提取需要的信息)

- lxml
  lxml 是一种使用Python 编写的库，可以迅速、灵活地处理XML。 它支持XML Path Language (XPath) 和Extensible Stylesheet Language Transformation (XSLT)，并且实现了常见的 ElementTree API。

lxml是python第三方库，需要自己安装。安装会遇到很些问题，还是那句话：生命不息，折腾不止。

3：xpath语法

表格法：

序号	表达式	描述
01	nodename	选取此节点的所有子节点
02	/	从根节点选取
03	//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置
04	.	选取当前节点
05	..	选取当前节点的父节点
06	@	选取属性

> 看不懂？

推荐：chrome 插件：XPath Helper

chrome 浏览器也可以copy xpath.

> 打不开？ 那搜索引擎玩着吧.

效果显示：
w3school:xpath 教程
想要搜索到：路径表达式：如下表：中的nodename单词

下面列出了最有用的路径表达式：

表达式	描述
nodename	选取此节点的所有子节点。
/	从根节点选取。
//	从匹配选择的当前节点选择文档中的节点，而不考虑它们的位置。
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。

层层查询下来：

4：语法实例

xpath语法实例：chrome 插件：Xpath helper
目标：当当心理学图书链接

- 图书名称：Bookname `xpath = //li/div/a/@title`

- 作者： Writer
  `xpath=//div[@class="publisher_info"]/p[@class="author"]/a[1]//@title`

- 出版时间：Time `xpath = //div[@class="publisher_info"]/p[@class="publishing_time"]`
- 评价数： Star `xpath = //p[@class="star"]/a`
- 简介： Detail `xpath = //p[@class="detail"]`
- 图书售价：Price_n, `xpath = //div[@class="inner"]/p[@class="price"]/span[@class="price_n"]`
- 图书定价：Price_r `xpath = //div[@class="inner"]/p[@class="price"]/span[@class="price_r"]`
- 网址链接：Url `xpath = //div[@class="inner"]/a//@href`

5：实战心理学图书

在lxml下如何使用：

`selector = etree.HTML(response)` response为网页源代码

抓取：图书标题，评价人数实例：

```
# title属性是书名
self.Bookname_pattern_3 = r"//li/div/a/@title"
selector = etree.HTML(response)
booknames = selector.xpath(self.Bookname_pattern_3) # 返回一个list
# 评价人数
self.Star_pattern_3 = r'//p[@class="star"]/a/text()'
selector = etree.HTML(response)
stars = selector.xpath(self.Star_pattern_3)         # 返回一个list
```

```
# 核心代码：
    def contents_xpath(self, one_url):
        html = requests.get(one_url, headers=self.headers)
        if html.status_code != 200:
            return -1
        else:
            response = html.text
        selector = etree.HTML(response)
        booknames = selector.xpath(self.Bookname_pattern_3)
        writers = selector.xpath(self.Writer_pattern_3)
        time = selector.xpath(self.Time_pattern_3)
        stars = selector.xpath(self.Star_pattern_3)
        details = selector.xpath(self.Detail_pattern_3)
        price_n = selector.xpath(self.Price_n_pattern_3)
        price_r = selector.xpath(self.Price_r_pattern_3)
        urls = selector.xpath(self.Url_pattern_3)
        All_data = []
        for booknames, writers, time, stars, details, price_n, price_r, url
s in zip(booknames, writers, time, stars, details, price_n, price_r, urls):
            data = {
                "bookname": booknames,
                "writers": writers,
                "stars": stars,
                "details": details,
                "price_n": price_n,
                "price_r": price_r,
                "urls": urls
            }
            All_data.append(data)
```

完整版代码：待重构

最后的数据存放在一个文本中：
如图：

当然：代码还可以继续重构,比如，图书介绍好些空白行如何处理；比如：先抓大再进行xpath等等之类的代码优化...

心理学这个图书栏有100页。也可以尝试100页如何抓取，存取。会不会出现问题。
核心代码，try …except都没写…（差评！）；不写注释（差评！）

6：参考及总结

01: w3school：xpath教程
02: lxml文档
03: 练习版代码

爬取思路还是和之前的系列专栏一致，解析方法变了而已。

Github:github

> 关于本人：
>
> > 国内小硕，半路出家的IT学习者。
> > 兴趣领域：爬虫 ， 数据科学
> > 本人正在构建一个共同成长爬虫小型社群。有兴趣私信。
> > 文档及代码托管在Github上。
