

Xpath爬虫实战

B站：大地的收藏夹

本次课程我们完成以下三个爬虫任务，巩固Xpath语法和lxml库爬虫知识点

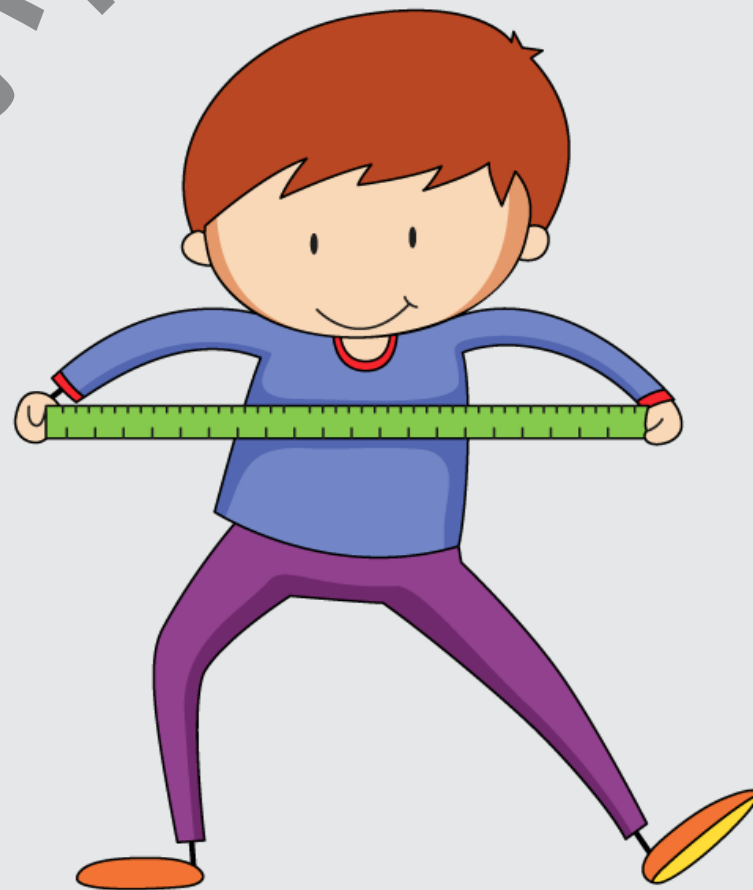
百度页面



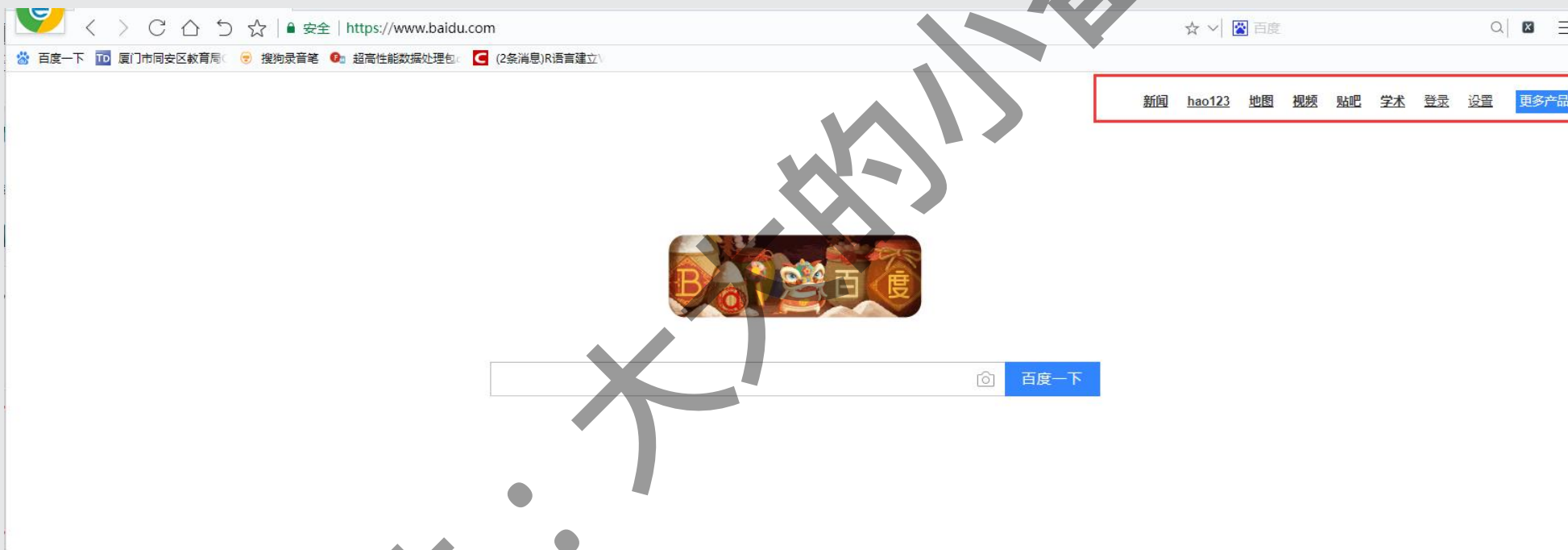
新浪热搜



豆瓣电影



百度页面



任务：抓取“新闻”等标签及它对应的链接。

提示点：

- 1、导入必要的库或模块
- 2、定义网页和请求头
- 3、获取html页面（注意编码和转码的问题）
- 4、etree解析
- 5、观察网页源码，查看标签特征
- 6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）
- 7、存储数据（zip函数双循环）



1、导入必要的库或模块

```
1 import requests
2 from lxml import etree
```

2、定义网页和请求头

```
4 # 爬取网页
5 url = "https://www.baidu.com/"
6
7 # 定义请求头
8 headers = {
9     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) Apple
10 }
```

3、获取html页面（注意编码和转码的问题）

```
11 # 获取html 字符串
12 response = requests.get(url, headers = headers)
13 content = response.content.decode('utf8')
```

4、etree解析

```
15 #将html 字符串解析
16 html = etree.HTML(content)
```

5、观察网页源码，查看标签特征

```
<div id="u" /.../div>
▼<div id="u1">
...
  <a href="http://news.baidu.com" name="tj_trnews" class="mnav">新闻</a> == $0
  <a href="https://www.hao123.com" name="tj_trhao123" class="mnav">hao123</a>
  <a href="http://map.baidu.com" name="tj_trmap" class="mnav">地图</a>
  <a href="http://v.baidu.com" name="tj_trvideo" class="mnav">视频</a>
  <a href="http://tieba.baidu.com" name="tj_trtieba" class="mnav">贴吧</a>
  <a href="http://xueshu.baidu.com" name="tj_trxueshu" class="mnav">学术</a>
  <a href="https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F" name="tj_login"
  class="lb" onclick="return false;">登录</a>
  <a href="http://www.baidu.com/gaoji/preferences.html" name="tj_settingicon" class="pf">设置</a>
  <a href="http://www.baidu.com/more/" name="tj_briicon" class="bri" style="display: block;">更多产品</a>
  <div class="bdnarrow bdbriarrow" style="display: none;"></div>
</div>
▶<div class="bdbri bdbriimg" style="opacity: 1; min-height: 873px; display: none;">...</div>
</div>
```

6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）

```
18 contents = html.xpath("//div[@id = 'u1']/a/text()")
19 print(contents)
20
21 urls = html.xpath("//div[@id = 'u1']/a/@href")
22 print(urls)
```

```
In [46]: contents = html.xpath("//div[@id = 'u1']/a/text()")
...: print(contents)
['新闻', 'hao123', '地图', '视频', '贴吧', '学术', '登录', '设置', '更多产品']
```

```
In [47]: urls = html.xpath("//div[@id = 'u1']/a/@href")
...: print(urls)
['http://news.baidu.com', 'https://www.hao123.com', 'http://map.baidu.com',
'http://v.baidu.com', 'http://tieba.baidu.com', 'http://xueshu.baidu.com',
'https://passport.baidu.com/v2/?login&tpl=mn&u=http%3A%2F%2Fwww.baidu.com%2F',
'http://www.baidu.com/gaoji/preferences.html', 'http://www.baidu.com/more/']
```

7、存储数据（zip函数双循环）

```
24 #存储数据
25 egs=[]
26 for content,url in zip(contents,urls):
27     eg = {}
28     eg = {
29         "content":content,
30         "url":url
31     }
32     egs.append(eg)
```

The screenshot shows a Python IDE with a list of 9 elements and a detailed view of the first element (index 0).

egs - List (9 elements)

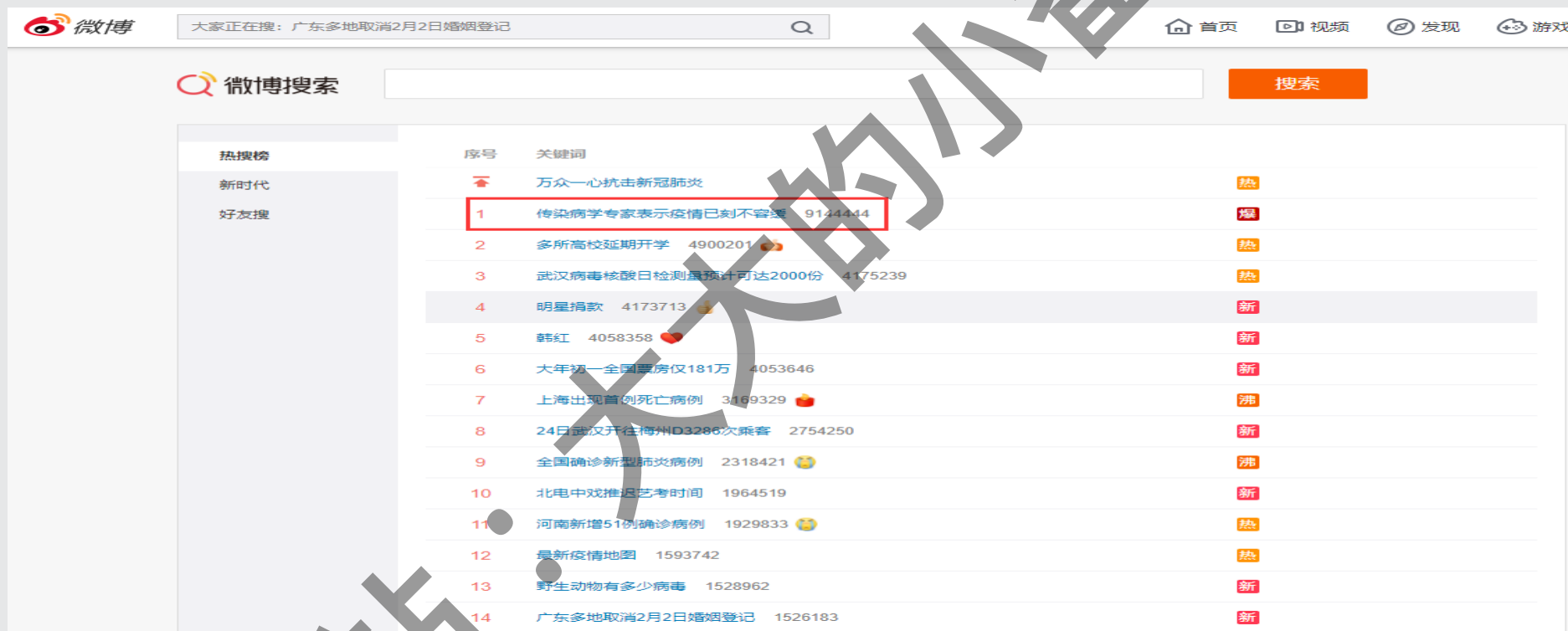
Index	Type	Size	Value
0	dict	2	{'content':_ElementUnicodeResult object of lxml....
1	dict	2	{'content':_ElementUnicodeResult object of lxml....
2	dict	2	{'content':_ElementUnicodeResult object of lxml....
3	dict	2	{'content':_ElementUnicodeResult object of lxml....
4	dict	2	{'content':_ElementUnicodeResult object of lxml....
5	dict	2	{'content':_ElementUnicodeResult object of lxml....
6	dict	2	{'content':_ElementUnicodeResult object of lxml....
7	dict	2	{'content':_ElementUnicodeResult object of lxml....
8	dict	2	{'content':_ElementUnicodeResult object of lxml....

0 - Dictionary (2 elements)

Key	Type	Size	Value
content	str	1	新闻
url	etree._Elemen...	1	http://news.baidu.com

Save and Close Close

新浪热搜



序号	关键词	热度
1	万众一心抗击新冠肺炎	热
2	传染病学专家表示疫情已刻不容缓 9144444	爆
3	多所高校延期开学 4900201	热
4	武汉病毒核酸日检测量预计可达2000份 4175239	热
5	明星捐款 4173713	新
6	韩红 4058358	新
7	大年初一全国票房仅181万 4053646	新
8	上海出现首例死亡病例 3469329	沸
9	24日武汉开往梅州D3286次乘客 2754250	新
10	全国确诊新型肺炎病例 2318421	沸
11	北电中戏推迟艺考时间 1964519	新
12	河南新增51例确诊病例 1929833	热
13	最新疫情地图 1593742	热
14	野生动物有多少病毒 1528962	新
15	广东多地取消2月2日婚姻登记 1526183	新

任务：爬取热度榜的内容和热度值

提示点：

- 1、导入必要的库或模块
- 2、定义网页和请求头
- 3、获取html页面（注意编码和转码的问题）
- 4、etree解析
- 5、观察网页源码，查看标签特征
- 6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）
- 7、存储数据



1—4常规操作

```
1 import requests
2 from lxml import etree
3
4 #爬取网页
5 url = "https://s.weibo.com/top/summary?cate=realtimehot&sudaref=www.b
6
7 #定义请求头
8 headers = {
9     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
10     "Referer": "https://login.sina.com.cn/sso/login.php?url=https%
11 }
12 #获取html字符串
13 response = requests.get(url, headers = headers)
14 content = response.content.decode('utf8')
15
16 #将html字符串解析
17 html = etree.HTML(content)
18
```

注：请求头主要完成user-agent, referrer, cookie, 这里我们多添加一些信息。

5、观察网页源码，查看标签特征

```
▼ <tbody>
  ▼ <tr class>
    ▼ <td class="td-01">
      <i class="icon-top"></i>
    </td>
    ▼ <td class="td-02">
      <a href="/weibo?q=%23%E4%B8%87%E4%BC%97%E4%B8%80%E5%BE%83%E6%8A%97%E5%87%BB%E6%96%B0%E5%86%A0%E8%82%BA%E7%82%8E%23&Refer=new_time" target="_blank">万众一心抗击新冠肺炎</a>
    </td>
    ▼ <td class="td-03">
      <i class="icon-txt icon-txt-hot">热</i>
    </td>
  </tr>
  ▼ <tr class>
    <td class="td-01 ranktop">1</td>
    ▼ <td class="td-02">
      ...
      <a href="/weibo?q=%23%E4%BC%A0%E6%9F%93%E7%97%85%E5%AD%A6%E4%B8%93%E5%AE%B6%E8%A1%A.../96%AB%E6%83%85%E5%B7%B2%E5%88%BB%E4%B8%8D%E5%AE%B9%E7%BC%93%23&Refer=top" target="_blank">传染病学专家表示疫情已刻不容缓</a> == $0
      <span>9144444</span>
    </td>
    ▶ <td class="td-03">...</td>
  </tr>
```

tr标签下的td标签下的一些a标签内容和span标签下的内容
下面我们具体的来分析一下，并写出对应的xpath语法

```
<tbody>
  <tr class="td-01">
    <td class="td-01">
      <i class="icon-top"></i>
    </td>
    <td class="td-02">
      <a href="/weibo?q=%23%E4%B8%87%E4%BC%97%E4%B8%80%E5%BF%83%E6%8A%97%E5%87%BB%E6%96%B0%E5%86%A0%E8%82%BA%E7%82%8E%23&Refer=new_time" target="_blank">万众一心抗击新冠肺炎</a>
    </td>
    <td class="td-03">
      <i class="icon-txt icon-txt-hot">热</i>
    </td>
  </tr>
  <tr class="td-01 ranktop">
    <td class="td-01 ranktop">1</td>
    <td class="td-02">
      <a href="/weibo?q=%23%E4%BC%A0%E6%9F%93%E7%97%85%E5%AD%A6%E4%B8%93%E5%AE%B6%E8%A1%A7%96%AB%E6%83%85%E5%B7%B2%E5%88%BB%E4%B8%8D%E5%AE%B9%E7%BC%93%23&Refer=top" target="_blank" class="class">传染病专家表示疫情已刻不容缓</a>
      <span class="9144444"></span>
    </td>
    <td class="td-03">...</td>
  </tr>
  <tr class="td-01 ranktop">
    <td class="td-01 ranktop">2</td>
    <td class="td-02">
      <a href="/weibo?q=%E5%A4%9A%E6%89%80%E9%AB%98%E6%A0%A%E5%B8%B6%E6%9C%9F%E5%8C%80%E5%AD%A6&Refer=top" target="_blank" class="class">多所高校延期开学</a>
      <span class="4900201"></span>
      
    </td>
  </tr>
```

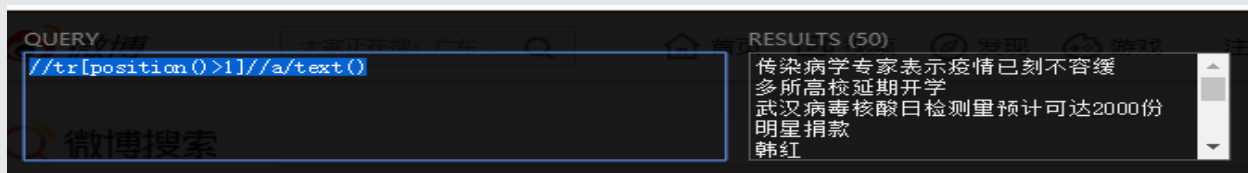
tr标签作为爷爷辈的标签
父标签有一些td标签。

热搜关键词的内容都在tr标签下的a标签中，或者td标签下的a标签中。

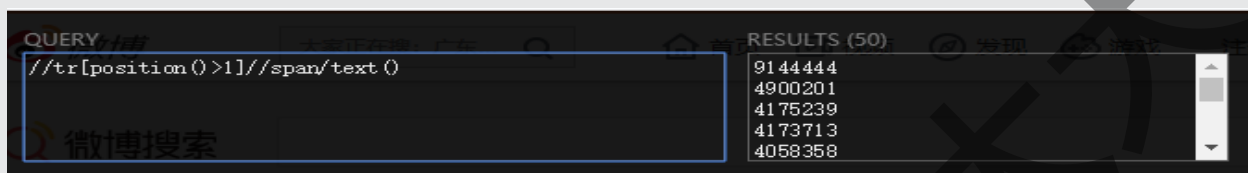
热搜的热度在tr下的i或span标签中，td标签下的i或span标签中。

QUERY: //tr[position()>1]/a/text()
RESULTS (50):
传染病专家表示疫情已刻不容缓
多所高校延期开学
武汉病毒核酸日检测量预计可达2000份
明星捐款
韩红

热搜榜	序号	关键词
新时代	↓	万众一心抗击新冠肺炎
好友搜	1	传染病专家表示疫情已刻不容缓 9144444
	2	多所高校延期开学 4900201 🇨🇳
	3	武汉病毒核酸日检测量预计可达2000份 4175239



热搜榜	序号	关键词
新时代	🔥	万众一心抗击新冠肺炎
好友搜	1	传染病专家表示疫情已刻不容缓 9144444
	2	多所高校延期开学 4900201 🧡
	3	武汉病毒核酸日检测量预计可达2000份 4175239



热搜榜	序号	关键词
新时代	🔥	万众一心抗击新冠肺炎
好友搜	1	传染病专家表示疫情已刻不容缓 9144444
	2	多所高校延期开学 4900201 🧡

需要解释一下：

1、为何是`//tr`，而不是`//td`

2、为何后面是`//a`和`//span`，而不是单斜杠`/`

3、强调`/text()`的作用

6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）

7、存储数据

方法一：

```
19 #Xpath提取数据
```

```
20 trs = html.xpath('//tr[position()>1]')
```

```
23 hots= []
```

```
24 for tr in trs:
```

```
25     eg = {}
```

```
26     content = tr.xpath('.//a/text()')[0]
```

```
27     hot = tr.xpath('.//span/text()')[0]
```

```
28     eg = {
```

```
29         "content":content,
```

```
30         "hot":hot
```

```
31     }
```

```
32     hots.append(eg)
```

“.”的作用：表示在当前路径下，不然会在全局下进行匹配

[0]的意义：在for循环中，单个处理，
逐个提取，不能批量处理。

结果完成

hots - List (50 elements)

Index	Type	Size	Value
0	dict	2	{'content': _ElementUnicodeResult...
1	dict	2	{'content': _ElementUnicodeResult...
2	dict	2	{'content': _ElementUnicodeResult...
3	dict	2	{'content': _ElementUnicodeResult...
4	dict	2	{'content': _ElementUnicodeResult...
5	dict	2	{'content': _ElementUnicodeResult...
6	dict	2	{'content': _ElementUnicodeResult...
7	dict	2	{'content': _ElementUnicodeResult...
8	dict	2	{'content': _ElementUnicodeResult...
9	dict	2	{'content': _ElementUnicodeResult...
10	dict	2	{'content': _ElementUnicodeResult...

Save and Close Close

0 - Dictionary (2 elements)

Key	Type	Size	Value
content	str	1	大年初一全国票房仅18...
hot	str	1	4864594

Save and Close Close

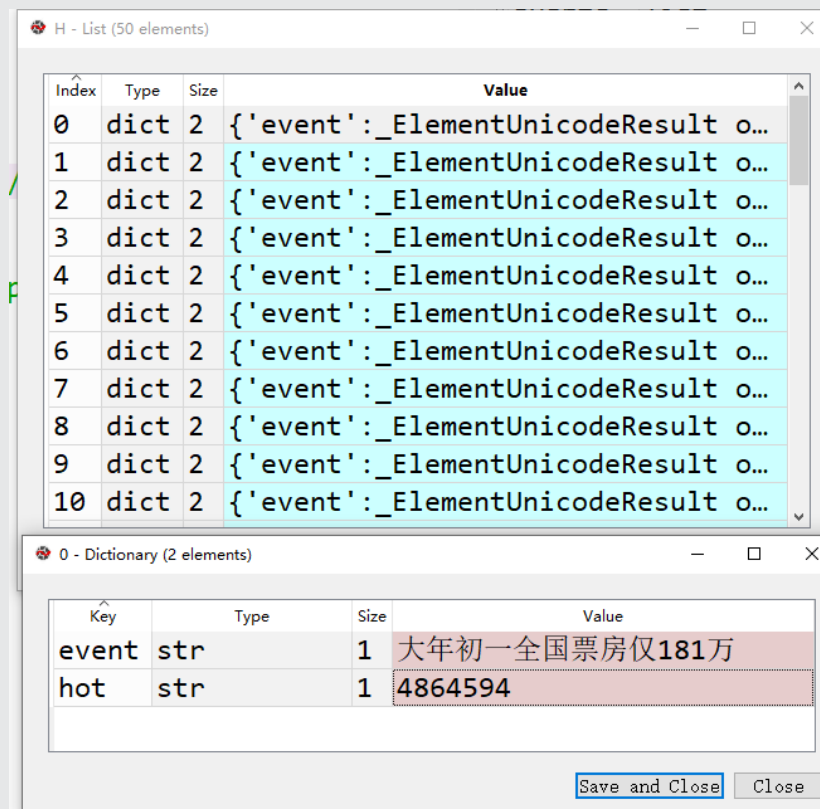
注：这里我刷新和重跑了程序，所以数据和之前页面中的不一致了。

方法二:

```
34 ##
35 events = html.xpath('//tr[position()>1]//a/text()')
36 #print(event)
37
38 hots = html.xpath('//tr[position()>1]//span/text()')
39 #print(hot)
40
41 #存储数据
42 H=[]
43 for event,hot in zip(events,hots):
44     eg = {}
45     eg = {
46         "event":event,
47         "hot":hot
48     }
49     H.append(eg)
```

分别获取数据

zip()多变量处理存储



The screenshot shows a web browser window with a table of movie events. The table has two columns: 'event' and 'hot'. The 'event' column contains movie titles, and the 'hot' column contains box office numbers. The table is titled 'H - List (50 elements)'.

Index	Type	Size	Value
0	dict	2	{'event': '_ElementUnicodeResult o...
1	dict	2	{'event': '_ElementUnicodeResult o...
2	dict	2	{'event': '_ElementUnicodeResult o...
3	dict	2	{'event': '_ElementUnicodeResult o...
4	dict	2	{'event': '_ElementUnicodeResult o...
5	dict	2	{'event': '_ElementUnicodeResult o...
6	dict	2	{'event': '_ElementUnicodeResult o...
7	dict	2	{'event': '_ElementUnicodeResult o...
8	dict	2	{'event': '_ElementUnicodeResult o...
9	dict	2	{'event': '_ElementUnicodeResult o...
10	dict	2	{'event': '_ElementUnicodeResult o...

Below the list, a dictionary window titled '0 - Dictionary (2 elements)' is shown, displaying the extracted data for the first row:

Key	Type	Size	Value
event	str	1	大年初一全国票房仅181万
hot	str	1	4864594

Buttons for 'Save and Close' and 'Close' are visible at the bottom right of the dictionary window.

豆瓣电影——基础



任务：
爬取电影名，评论者和评分，影评

提示点：

- 1、导入必要的库或模块
- 2、定义网页和请求头
- 3、获取html页面（注意编码和转码的问题）
- 4、etree解析
- 5、观察网页源码，查看标签特征
- 6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）
- 7、存储数据



1—4常规操作

```
1 import requests
2 from lxml import etree
3
4 #爬取网页
5 url = "https://movie.douban.com/review/12185134/"
6
7 #定义请求头
8 headers = {
9     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
10 }
11 #获取html 字符串
12 response = requests.get(url, headers = headers)
13 content = response.content.decode('utf8')
14
15 #将html 字符串解析
16 html = etree.HTML(content)
17
```

5、观察网页源码，查看标签特征

```
    </div>
    <div class="subject-title">
    <a href="https://movie.douban.com/subject/30306570/" class="">&nbsp;囡妈</a> == $0
    </div>
    <div class="subject-img">...</div>
    <div class="subject-info movie-info">...</div>
```

6、编写xpath语法，获取标签内容（文本信息末尾添加/text()）

```
17
18 #Xpath提取数据
19 #抓电影名
20 title = html.xpath('//div[@class="subject-title"]/a/text()')
21 print(title)
22
```

```
In [70]: title = html.xpath('//div[@class="subject-title"]/a/
text()')
: print(title)
['>\xa0囡妈']
```

结果：
是列表形式
有>等异常字符

对代码进行修改（列表的
提取，字符串的操作）

```
18 #Xpath提取数据
```

```
19 #抓电影名
```

```
20 title = html.xpath('//div[@class="subject-title"]/a/text())[0][2:]
```

```
21 print(title)
```

```
22
```

[0] 列表元素提取

[2:] 字符串按照要求从第3位开始

```
In [73]: title = html.xpath('//div[@class="subject-title"]/a/text())[0][2:]
        ....: print(title)
```

囧妈

```

▼ <div class=
  ▼ <div class="main" id="12185134">
    ▶ <a class="avatar author-avatar left" href="https://www.douban.com/people/renjiananhuo/">...</a>
    ▼ <header class="main-hd">
      ▼ <a href="https://www.douban.com/people/renjiananhuo/" class=
        <span class>邓安庆</span> == $0
      </a>
      "
      评论
      "
      <a href="https://movie.douban.com/subject/30306570/" class>囧妈</a>
      <span class="allstar40 main-title-rating" title="推荐"></span>
      <span class="main-title-hide">4</span>
      <span content="2020-01-25" class="main-meta">2020-01-25 11:59:09</span>
      ▶ <script type="application/ld+json">...</script>
    </header>
    ▶ <div class="main-bd taboola-hide-container" id="review-content" data-ad-ext="有用627 · 没用31">...
  </div>
  <div class="main-ft"> </div>

```

header下的a标签下的span标签，其它span标签是header的子标签

//header/a/span/text()

```

▼ <header class="main-hd">
  ▼ <a href="https://www.douban.com/people/renjiananhuo/" class>
    <span class>邓安庆</span>
  </a>
  "
  评论
  "
  <a href="https://movie.douban.com/subject/30306570/" class>囧妈</a>
  <span class="allstar40 main-title-rating" title="推荐">/span> == $0
  <span class="main-title-hide">4</span>
  <span content="2020-01-25" class="main-meta">2020-01-25 11:59:09</span>
  ▶ <script type="application/ld+json">...</script>

```

header下的众多span标签下仅含有title属性的span标签中就有评分

//header//span/@title

```
23 #抓评论者和评分
```

```
24 commenter = html.xpath('//header/a/span/text()')[0]
```

```
25 rank = html.xpath('//header//span/@title')[0]
```

```
26
```

```
In [80]: commenter = html.xpath('//header/a/span/text()')[0]
```

```
....: rank = html.xpath('//header//span/@title')[0]
```

```
....: print(commenter,rank)
```

邓安庆 推荐



```
</p>
▼<div id="link-report" class="" == $0
▼<div class="review-content clearfix" data-author="邓安庆" data-url="https://movie.douban.com/
review/12185134/" data-original="1">
▼<p class="xh-highlight">
"
看电影时，我在想徐峥饰演的徐伊万有多大？之所以好奇这个问题，是忽然想起村上春树一个采访，有人问他为何他小说中的主人公多是三十五六岁，村上说：“（他们）还停留在人生的中间地带。我想我的主角需要的，大概是扮演故事的‘引水人’。如果到了五、六十岁，会有人生的种种关系纠葛，所以动作必然会变得迟缓……（他们）虽已不年轻，却又尚未达到中年。虽有某种程度的自我，却又尚未巩固，也还有迷惘。要朝哪一个方向前进都很自由。”
</p>
▼<p class="xh-highlight">
"
我就是村上春树所说的这个年龄段，虽还有迷惘，但来去自由。但回到徐伊万的身上，他吗？他有个妈妈（爸爸已经去世），有个前妻，没有孩子，事业有成，成天忙碌……这样的人，在我的身边有很多。那他可能比我大一点，四十岁左右。村上说三十五六岁的人“虽有某种程度的自我，却又尚未巩固，也还有迷惘”，那四十多岁的人往往就是“我执”，周作人《说中年》里有一句非常精炼的概括：“执著人生，私欲深。”年轻时的弹性柔软都已经变得固执僵硬，不再去尝试了解别人的想法，因为他已经对这个世界有一套成见，而且他也会认为自己现在的成功就是按照那套成见而达到的。所以他在自己的世界里，听不到周遭人的声音，不论是妻子的，还是母亲的。”
</p>
▼<p class="xh-highlight">
"
电影里袁泉饰演的妻子张璐让徐伊万明白他想要的是一个“理想的妻子”，然后他按照这个标准想去改造张璐，他意识不到妻子是一个有自己想法的活生生的人。这点意在提醒“世界不是围绕你一个人转的”，你躲在“我执”之中，得到是别人的疏远和逃避，而这反过来会让徐伊万这样的人加深了自己的控制欲望。这是一个恶性循环。而张璐说的这一套台词，徐伊万也对徐妈妈说了，只不过是“理想的妻子”换成“理想的儿子”。控制与被控制，改造与被改造，越是亲密关系，越容易如此。”
</p>
▼<p class="xh-highlight">
"
看这部电影，我最深的感受是到人到中年后的“疲惫感”：一切都是紧紧揪着的，一切都是深陷其中，于是“纾解”成了戏剧的发展动力，母与子，夫与妻，在整部电影里不断地冲突、计较、试探、和解……单纯想放松的观众，可以看到一个好玩的电影；想看更深的观众，可以体会其中的人生百味。有些喜剧电影往深处想，核心是苦涩的。如果没有这个心，那就是闹剧。而《囧妈》是非常合格的喜剧电影，电影艺术我不懂也不论，单就我作为一个普通观众看，我是能感受到电影呈现后面的那个“心”的——那是一颗中年人的心。”
</p>
▼<p class="xh-highlight">
"
一直以来都很想乘坐k3/4，从北京出发，经蒙古，横穿西伯利亚，走它个7692公里，最后达到莫斯科。在这样一个封闭的空间，与陌生人相处，一直在路上，会让我有很多想象。但如果是跟自己的老妈待在这列火车上呢？我无法想象。徐峥是一个非常聪明的导演，他很敏锐地从其中抓到戏剧冲突点：足够的时间（时长131小时31分），不变的空间，互不理解的母子……是可以结构出一部好看的戏来。看完后的感受，我觉得《囧妈》是做到好看这一点。徐峥与黄梅莹真是好演员，我觉得他们把母子之间解不断理还乱的复杂感情状态表演了出来。前一部分母子冲突产生的笑点是由着表演的节奏感带出来的，很自然，也的确惹人笑；后半部分随着母子间相互深入地接触了解，好几场戏十分动人。”
```

div标签下所有的p标签



//div[@id="link-report"]//p/text()


```
28 #抓影评
29 comment = html.xpath('//div[@id="link-report"]//p/text()')
30 print(len(comment))
31
```

查看结果的长度

```
In [81]: comment = html.xpath('//div[@id="link-report"]//p/
text()')
...: print(len(comment))
5
```

结果的长度为5，所有的p标签分别存在了一个列表里。

这里需要列表的拼接！

```
28 #抓影评
29 comment = html.xpath('//div[@id="link-report"]//p/text()')
30 #print(len(comment))
31 #影评拼接
32 comment = ''.join(comment)
33 print(comment)
34
```

7、存储数据

存入movie的字典

```
36 movie = {  
37     "title":title,  
38     "commenter":commenter,  
39     "rank":rank,  
40     "comment":comment,  
41 }  
42
```



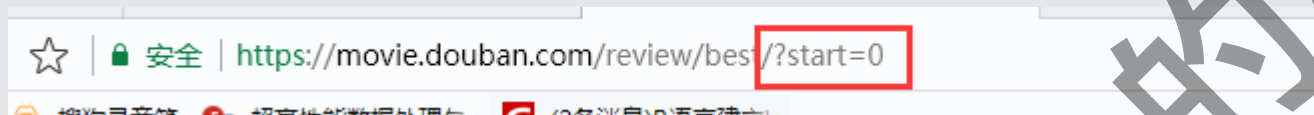
豆瓣电影——进阶



任务：获取这个页面以下所有的影评数据，共计5页100条。

该怎么做？

一页页点击，每页中的每个电影一次次点击进去，复制相应数据出来保存，再进行下一次工作，循环往复，直至全部复制完成。



这里的`start=0`意味着页面的跳转，修改后，即可跳转到其他影评页面

```
https://movie.douban.com/review/best/?start=0  
https://movie.douban.com/review/best/?start=20  
https://movie.douban.com/review/best/?start=40
```

url几乎完全一样，只要修改`start=`的值，并且该值是+20递增的。

我们要模拟这个换页的操作，





这里的标题都可以帮助我们跳转到具体的影评页面，意味着每个标题下面都有链接供跳转。

这里的a标签下的href属性，就是文本信息对应的链接。



意味着，我们要将当前页面下的所有影评的链接，即href做一个提取。

等我们进入下一页的时候，就是重复工作而已。

总结：

- 1、获取共计5页的所有的url；
- 2、遍历每一页的url，从每页中得到所有的a标签下的href属性；
- 3、将href中的链接按上节课的内容，将所有关于影评的内容爬取下来。



1、获取共计5页的所有的url

```
1 """
2 1、构造共计5页的所有的url
3 """
4 |
5 for i in range(0,5,1):
6     i= i*20
7     url = "https://movie.douban.com/review/best/?start={}".format(i)
8     print(url)
9
```

← 构造从0开始，20位步长的等差数列

← format()函数代入

```
.... print(url)
https://movie.douban.com/review/best/?start=0
https://movie.douban.com/review/best/?start=20
https://movie.douban.com/review/best/?start=40
https://movie.douban.com/review/best/?start=60
https://movie.douban.com/review/best/?start=80
```

修改代码，将构造的url保存到空列表中

2 1、构造共计5页的所有的url

3 """

4 urls = []

5 for i in range(0,5,1):

6 i= i*20

7 url = "https://movie.douban.com/review/best/?start={}".format(i)

8 urls.append(url)

urls list 5 ['https://movie.douban.com/review/best/?start=0', ...

urls - List (5 elements)

Index	Type	Size	Value
0	str	1	https://movie.douban.com/review/bes...
1	str	1	https://movie.douban.com/review/bes...
2	str	1	https://movie.douban.com/review/bes...
3	str	1	https://movie.douban.com/review/bes...
4	str	1	https://movie.douban.com/review/bes...

Save and Close Close

2、遍历每一页的url，从每页中得到所有的a标签下的href属性

在for循环内完成请求，解码，解析和提取的工作，即常规操作

```
14 import requests
15 from lxml import etree
16
17 headers={
18     "User-Agent":"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
19     }
20
21 for url in urls:
22     #发送请求
23     response = requests.get(url,headers = headers)
24     #解码
25     content = response.content.decode('utf8')
26     #解析html字符串
27     html = etree.HTML(content)
28     #xpath提取每个电影的url
29
```

Xpath语法进行提取的时候，观察下网页的情况，编辑xpath代码

```
<div class="main-bd">
  <h2 class="">
    <a href="https://movie.douban.com/review/12185134/" class="">一颗中年人的心</a>
  </h2>
  <div id="review_12185134_short" class="review-short" data-rid="12185134">...</div>
  <div id="review_12185134_full" class="hidden">...</div>
  <div class="action">...</div>
</div>
::after
</div>
</div>
<div data-cid="12184732">...</div>
<div data-cid="12184901">...</div>
<div data-cid="12182419">...</div>
<div data-cid="12182543">...</div>
<div data-cid="12184976">...</div>
<div data-cid="12184593">...</div>
<div data-cid="12181209">...</div>
<div data-cid="12183685">...</div>
<div data-cid="12175335">...</div>
<script type="text/javascript" src="https://img3.doubanio.com/misc/mixed_static/57017d0be3a16dcb.js"></script>
```

h2下的a标签的href属性即我们需要的电影的url

//h2/a/@href

QUERY

//h2/a/@href
(展开)

RESULTS (10)

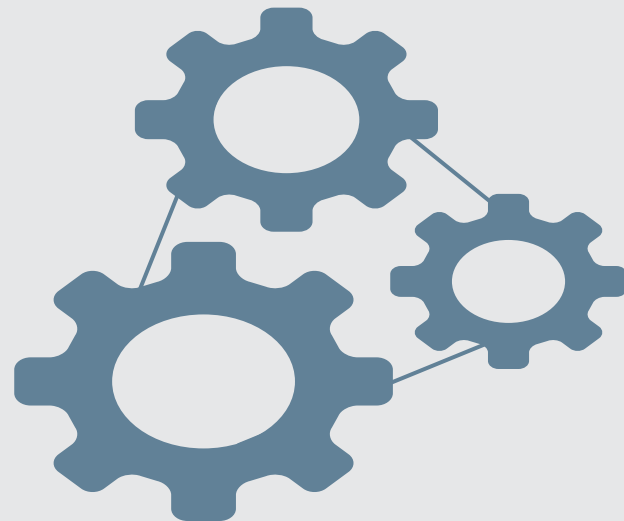
<https://movie.douban.com/review/12185134/>
<https://movie.douban.com/review/12184732/>
<https://movie.douban.com/review/12184901/>
<https://movie.douban.com/review/12182419/>
<https://movie.douban.com/review/12182543/>

```
11 """
12 2、获取每个影评的url
13 """
14 import requests
15 from lxml import etree
16
17 headers={
18     "User-Agent":"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
19     }
20
21 for url in urls:
22     #发送请求
23     response = requests.get(url,headers = headers)
24     #解码
25     content = response.content.decode('utf8')
26     #解析html字符串
27     html = etree.HTML(content)
28     #xpath提取每个电影的url
29     detail_url = html.xpath('//h2/a/@href')
30     print(detail_url)
31     break
32
```

→

```
[ 'https://movie.douban.com/review/12185134/', 'https://
movie.douban.com/review/12184732/', 'https://movie.douban.com/
review/12184901/', 'https://movie.douban.com/review/
12182419/', 'https://movie.douban.com/review/12182543/',
'https://movie.douban.com/review/12184976/', 'https://
movie.douban.com/review/12184593/', 'https://movie.douban.com/
review/12181209/', 'https://movie.douban.com/review/
12183685/', 'https://movie.douban.com/review/12175335/']
```

修改代码，遍历所有的url，获取全部电影的url，并且存储到列表中。



```
11 """
12 2、获取每个影评的url
13 """
14 import requests
15 from lxml import etree
16
17 headers={
18     "User-Agent":"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit
19 }
20 detail_urls = []
21 for url in urls:
22     #发送请求
23     response = requests.get(url,headers = headers)
24     #解码
25     content = response.content.decode('utf8')
26     #解析html字符串
27     html = etree.HTML(content)
28     #xpath提取每个电影的url
29     detail_url = html.xpath('//h2/a/@href')
30     detail_urls.append(detail_url)
31
32 print(detail_urls)
33
```

空列表

结果添加进空列表中

Xpath返回的结果是列表

3、将href中的链接按上节课的内容，将所有关于影评的内容爬取下来

由于我们全部的url存储在列表的列表中，因此这里需要一个嵌套的for循环。

```
35 """
36 3、获取每个电影的内容
37 """
38 for page in detail_urls:
39     for url in page:
40         #发送请求
41         response = requests.get(url,headers = headers)
42         content = response.content.decode('utf8')
43         #etree解析
44         html = etree.HTML(content)
45         #抓电影名
46         title = html.xpath('//div[@class="subject-title"]/a/text()')
47         #抓评论者和评分
48         commenter = html.xpath('//header/a/span/text()')[0]
49         rank = html.xpath('//header//span/@title')[0]
50         #抓影评
51         comment = html.xpath('//div[@id="link-report"]//p/text()')
52         #print(len(comment))
53         #影评拼接
54         comment = ''.join(comment)
55         movie = {
56             "title":title,
57             "commenter":commenter,
58             "rank":rank,
59             "comment":comment,
60         }
61         break
62     break
```

movie - Dictionary (4 elements)

Key	Type	Size	Value
comment	str	1	看电影时，我在想徐...
commenter	etree...	1	_ElementUnicodeResult o...
rank	etree...	1	_ElementUnicodeResult o...
title	str	1	囡妈

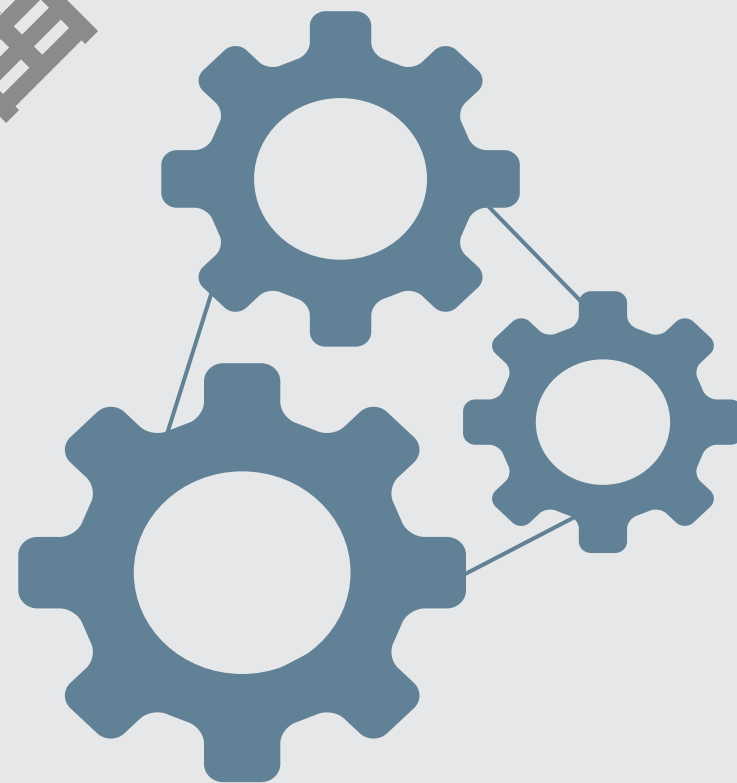
Save and Close Close

```
39 movies = []
40 i = 0
41 for page in detail_urls:
42     for url in page:
43         #发送请求
44         response = requests.get(url, headers = headers)
45         content = response.content.decode('utf8')
46         #etree解析
47         html = etree.HTML(content)
48         #抓电影名
49         title = html.xpath('//div[@class="subject-title"]/a/text()')[0][2:]
50         #抓评论者和评分
51         commenter = html.xpath('//header/a/span/text()')[0]
52         rank = html.xpath('//header//span/@title')[0]
53         #抓影评
54         comment = html.xpath('//div[@id="link-report"]//p/text()')
55         #print(len(comment))
56         #影评拼接
57         comment = ''.join(comment)
58         movie = {
59             "title": title,
60             "commenter": commenter,
61             "rank": rank,
62             "comment": comment,
63         }
64         movies.append(movie)
65         i += 1
66         print("第{}页已经爬取完了".format(i))
```

制作一个提示语，告知我们爬到第几页了

每次循环添加进空列表

制作一个提示语，告知我们爬到第几页了



```
....:         i += 1
....:         print("第{}页已经爬取完了".format(i))
第1页已经爬取完了
第2页已经爬取完了
Traceback (most recent call last):

  File "<ipython-input-105-37af0966de19>", line 14, in
<module>
    rank = html.xpath('//header//span/@title')[0]
IndexError: list index out of range
```

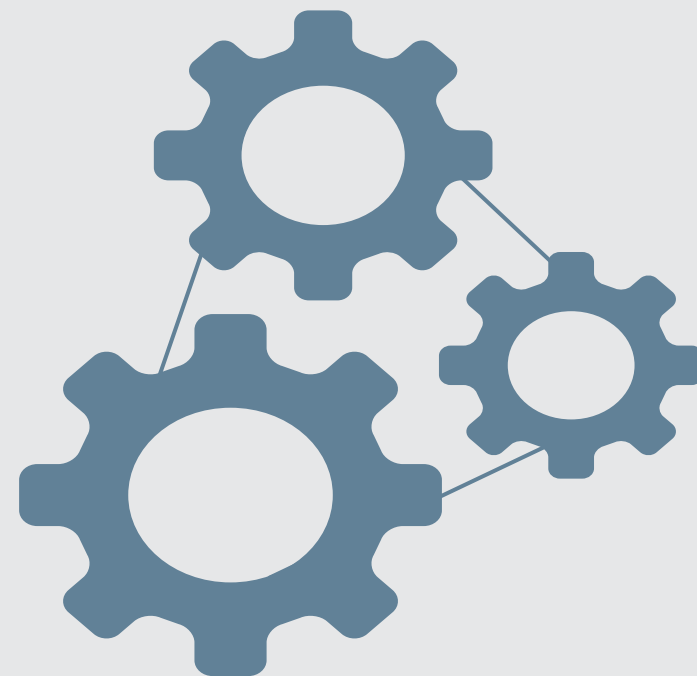
【报错原因】某个电影的html并不是统一的格式，因此终止了代码。

注意：万一爬取到某个电影的时候，它并不是统一的格式，代码就报错了，如何跳过错误继续执行？

这是所有爬虫代码都要做的事情！

```
38
39 movies = []
40 i = 0
41 for page in detail_urls:
42     for url in page:
43         try:
44             #发送请求
45             response = requests.get(url, headers = headers)
46             content = response.content.decode('utf8')
47             #etree解析
48             html = etree.HTML(content)
49             #抓电影名
50             title = html.xpath('//div[@class="subject-title"]/a/text()')[0][2:]
51             #抓评论者和评分
52             commenter = html.xpath('//header/a/span/text()')[0]
53             rank = html.xpath('//header//span/@title')[0]
54             #抓影评
55             comment = html.xpath('//div[@id="link-report"]//p/text()')
56             #print(len(comment))
57             #影评拼接
58             comment = ''.join(comment)
59             movie = {
60                 "title":title,
61                 "commenter":commenter,
62                 "rank":rank,
63                 "comment":comment,
64             }
65             movies.append(movie)
66         except:
67             continue
68     i += 1
69     print("第{}页已经爬取完了".format(i))
70
```

异常处理模块
try下面的代码，
如果出错了，
就继续下一个
循环。




```
...:         continue
...:         i += 1
...:         print("第{}页已经爬取完了".format(i))
```

第1页已经爬取完了
第2页已经爬取完了
第3页已经爬取完了
第4页已经爬取完了
第5页已经爬取完了

完结撒花~



movies - List (47 elements)

Index	Type	Size	Value
0	dict	4	{'title': '囡妈', 'commenter': _ElementUnicodeResult object of 1...
1	dict	4	{'title': '囡妈', 'commenter': _ElementUnicodeResult object of 1...
2	dict	4	{'title': '囡妈', 'commenter': _ElementUnicodeResult object of 1...
3	dict	4	{'title': '三生三世枕上书', 'commenter': _ElementUnicodeResult ob...
4	dict	4	{'title': '三生三世枕上书', 'commenter': _ElementUnicodeResult ob...
5	dict	4	{'title': '三生三世枕上书', 'commenter': _ElementUnicodeResult ob...
6	dict	4	{'title': '阳光普照', 'commenter': _ElementUnicodeResult object o...
7	dict	4	{'title': '非典十年祭', 'commenter': _ElementUnicodeResult object...
8	dict	4	{'title': '宸汐缘', 'commenter': _ElementUnicodeResult object of ...
9	dict	4	{'title': '性爱自修室 第二季', 'commenter': _ElementUnicodeResult ...
10	dict	4	{'title': '性爱自修室 第二季', 'commenter': _ElementUnicodeResult ...
11	dict	4	{'title': '甜蜜的永远', 'commenter': _ElementUnicodeResult object...
12	dict	4	{'title': '明星大侦探 第五季', 'commenter': _ElementUnicodeResult ...
13	dict	4	{'title': '爱情公寓5', 'commenter': _ElementUnicodeResult object ...
14	dict	4	{'title': '爱的迫降', 'commenter': _ElementUnicodeResult object o...
15	dict	4	{'title': '传染病', 'commenter': _ElementUnicodeResult object of ...
16	dict	4	{'title': '剧场版 吹响！上低音号～誓言的终章～', 'commenter': _Eleme...
17	dict	4	{'title': '乜代宗师', 'commenter': _ElementUnicodeResult object o...
18	dict	4	{'title': '绝代双骄', 'commenter': _ElementUnicodeResult object o...
19	dict	4	{'title': '非典十年祭', 'commenter': _ElementUnicodeResult object...

Save and Close Close

注：你可以在这里算出有几部电影没爬到
更高的要求：这里的模块化的代码是为了
大家学习掌握完整，

每块代码都可以定义为函数，在日常工作
中模块化的函数是被要求的，有兴趣的同
学试一试。把每个部分的代码改写成函数。

谢谢聆听