第五关 有惊无险：存储数据

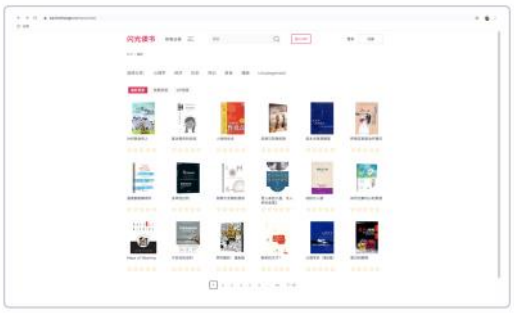
2022年12月5日　12:23

1. 项目代码

在展示代码之前，我们要知道，本节课要爬取网站的哪些信息。

1.1 明确需求

本节课要爬取的目标网站是《闪光科技》，依旧以谷歌 Chrome 浏览器为例，打开网站：https://wp.forchange.cn/resources/ 。



爬取的内容是该网站书籍列表页所有书籍的信息，包括：书名、ISBN、作者、用户评分、出版社、出版年、定价、页数。

上节课已经教会你写出一个爬取单本书籍信息的代码。虽说今天的项目可以手动复制网站各本书籍详情页的链接，作为参数依次粘贴到上节课代码中 requests.get() 的括号内，可以实现所有书籍信息的获取。

但一个一个网页去请求，再去爬取书籍的信息，是一个重复性很高的工作。所以我们需要结合 Python 的多个知识点来帮我们处理这个问题。

由于爬取整个网站的所有书籍信息需要点时间，这里我先提供实现批量爬取网站书籍列表页前 3 页书籍信息的代码。

在学习代码之前，需要知道数据是在网页的哪个位置爬取。

1.2 分析网页

项目目标是取出 3 个书籍列表页里每本书籍的信息，以书籍列表页的第 1 页为例：https://wp.forchange.cn/resources/ 。

进入网页后可以看到页面上有 18 本书籍，但页面上的书籍没有提供书名以外的信息，所以我们需要进入书籍的详情页来获取书籍的各项信息。

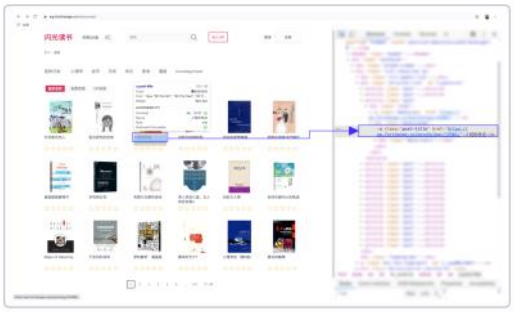
我们随机在网页上选择一本书籍，比如《人性的优点》这本书，我们点击下面的链接进入到书籍的详情页：

https://wp.forchange.cn/psychology/13501/ 。

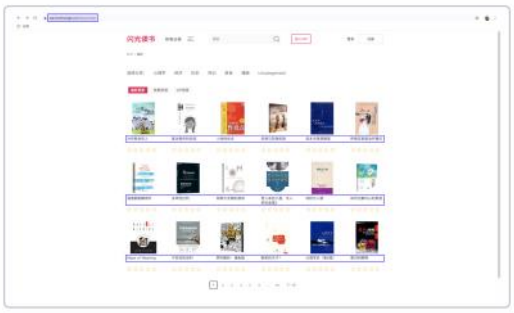
可以看到书籍的各项信息都在该详情页内。

通过键盘快捷键的方式打开浏览器开发者工具（Windows 用户可以在浏览器页面下按 Ctrl + Shift + I 键打开浏览器开发者工具，Mac 用户的快捷键为 command + option + I），并使用指针工具将鼠标光标放在书籍的各项信息，各点击一下。

可以看到，书名在属性为 class='title-detail' 的 h1 元素内。
书籍的详细信息在属性为 class='res-attrs' 的 div 元素内，每一项信息都在 dl 元素内。且信息的提示项（如"作者："、"出版社："等）都在 dt 元素中，信息的内容（如"戴尔·卡耐基"、"中国城市"等）都在 dd 元素中。
知道数据是在网页的哪个位置爬取以后，需要找到不同书籍详情页间的规律，实现批量爬取的功能。我们可以通过观察多个书籍详情页的链接来找找看规律。
回到网站书籍列表页的第 1 页，点击书籍《人性的优点》后面的三本书籍，进入它们的详情页，观察它们的书籍链接：
1）《自我与防御机制》：https://wp.forchange.cn/psychology/13499/ ；
2）《叔本华思想随笔》：https://wp.forchange.cn/psychology/13539/ ；
3）《萨提亚家庭治疗模式》：https://wp.forchange.cn/psychology/13497/。
可以发现，三本书籍的链接与书籍《人性的优点》的链接：https://wp.forchange.cn/psychology/13501/ ，在尾部的数字上没有很明显的规律。
但访问这四本书籍详细页的操作是一样的，都是在网站书籍列表页上点击书籍的名字后跳转至该书籍的详细页，所以在书籍列表页面里应该包含着书籍链接。我们可以在 HTML 源码上找找看。
再回到网站书籍列表页的第 1 页，打开浏览器开发者工具，并使用指针工具将鼠标光标放在书籍《人性的优点》的书名上，并点击一下。



在 HTML 源码上，书籍《人性的优点》的书名以及链接 https://wp.forchange.cn/psychology/13501/ 都在属性为 class='post-title' 的 a 元素内。
书籍列表页面里有书籍的链接，那就可以通过提取书籍链接并请求书籍详情页，提取对应书籍的详细信息。
由于书名与链接在同一个元素内，所以在提取书籍链接的同时顺便提取书名，不需要在书籍详情页上再去定位一个元素提取。
ok，网页分析完，我们再来明确一下我们项目的目标。
首先，我们会在网站前 3 页的书籍列表页获取书名和书籍链接：



其次，通过书籍链接访问书籍的详情页。
最后，在书籍的详细页面中爬取书籍的详细信息，书籍的详细信息包括：ISBN、作者、用户评分、出版社、出版年、定价、页数。

接下来体验一下我写好的代码吧。爬取数据需要一点时间，你要保持耐心。

1.3 体验代码

```python
# 体验代码

import requests
import csv
from bs4 import BeautifulSoup

# 设置列表，用以存储每本书籍的信息
data_list = []
# 设置页码 page_number
page_number = 1

# while 循环的条件设置为 page_number 的值是否小于 4
while page_number < 4:
    # 设置要请求的网页链接
    url = 'https://wp.forchange.cn/resources/page/' + str(page_number)

    # 请求网页
    books_list_res = requests.get(url)

    # 解析请求到的网页内容
    bs = BeautifulSoup(books_list_res.text, 'html.parser')
    # 搜索网页中所有包含书籍名和书籍链接的 Tag
    href_list = bs.find_all('a', class_='post-title')

    # 使用 for 循环遍历搜索结果
    for href in href_list:
        # 创建字典，用以存储书籍信息
        info_dict = {}
        # 提取书籍名
        info_dict['书名'] = href.text
        # 提取书籍链接
        book_url = href['href']
        # 通过书籍链接请求书籍详情页
        book_list_res = requests.get(book_url)

        # 解析书籍详情页的内容
        new_bs = BeautifulSoup(book_list_res.text, 'html.parser')
        # 搜索网页中所有包含书籍各项信息的 Tag
        info_list = new_bs.find('div', class_='res-attrs').find_all('dl')
```

```
        # 使用 for 循环遍历搜索结果
        for info in info_list:
            # 提取信息的提示项
            key = info.find('dt').text[:-2]
            # 提取信息的内容
            value = info.find('dd').text
            # 将信息添加到字典中
            info_dict[key] = value

        # 打印书籍的信息
        print(info_dict)
        # 存储每本书籍的信息
        data_list.append(info_dict)

    # 页码 page_number 自增
    page_number += 1
```

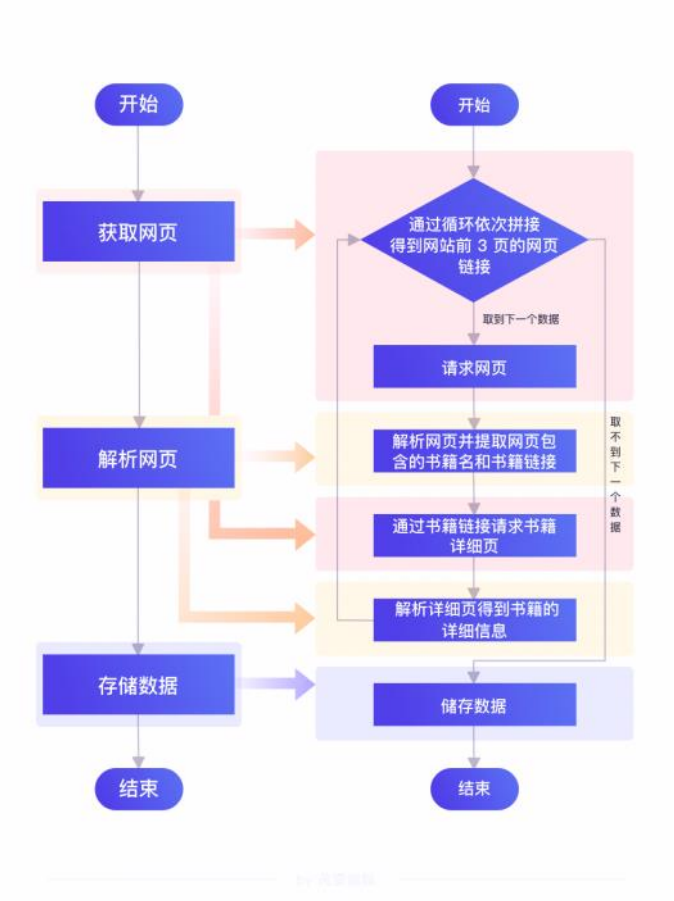
```
# 新建 csv 文件存储书籍信息
with open(r'D:\PythonTest\风变python学习资料\Python爬虫\第5关爬取数据.csv', 'w',encoding='utf-8-sig') as f:
    # 将文件对象转换成 DictWriter 对象
    writer = csv.DictWriter(f, fieldnames=['书名', '作者', '出版社', 'ISBN', '页数', '出版年', '定价'])
    # 写入表头与数据
    writer.writeheader()
    writer.writerows(data_list)
```

```
{'书名': '离经叛道', '作者': '亚当·格兰特/AdamGrant', '出版社': '浙江大学出版社', 'ISBN': '9787308157186', '页数': '280', '出版年': '2016-7-1', '定价': '49'}
{'书名': '孤獨的冷漠：逃避型依戀障礙的分析與修復', '作者': '岡田尊司', '出版社': '聯合文學', 'ISBN': '9789863232124', '出版年': '2017-5-15', '定价': '330NTD'}
{'书名': '心理统计学（第3版）', '作者': 'B.H.科恩', '出版社': '华东师范大学出版社', 'ISBN': '9787561782057', '页数': '935', '出版年': '2011-2-1', '定价': '98.00元'}
{'书名': '电影与新心理学', '作者': '[法]莫里斯·梅洛-庞蒂', '出版社': '商务印书馆', 'ISBN': '9787100166676', '页数': '163', '出版年': '2019-4-1', '定价': '45'}
```

```
#体验代码

importrequests
importcsv
frombs4importBeautifulSoup

#设置列表，用以存储每本书籍的信息
data_list=[]
#设置页码page_number
page_number=1

#while循环的条件设置为page_number的值是否小于4
whilepage_number<4:
#设置要请求的网页链接
url='https://wp.forchange.cn/resources/page/'+str(page_number)

#请求网页
books_list_res=requests.get(url)

#解析请求到的网页内容
bs=BeautifulSoup(books_list_res.text,'html.parser')
#搜索网页中所有包含书籍名和书籍链接的Tag
href_list=bs.find_all('a',class_='post-title')

#使用for循环遍历搜索结果
forhrefinhref_list:
#创建字典，用以存储书籍信息
```

```
info_dict={}
#提取书籍名
info_dict['书名']=href.text
#提取书籍链接
book_url=href['href']
#通过书籍链接请求书籍详情页
book_list_res=requests.get(book_url)

#解析书籍详情页的内容
new_bs=BeautifulSoup(book_list_res.text,'html.parser')
#搜索网页中所有包含书籍各项信息的Tag
info_list=new_bs.find('div',class_='res-attrs').find_all('dl')

#使用for循环遍历搜索结果
for info in info_list:
#提取信息的提示项
key=info.find('dt').text[:-2]
#提取信息的内容
value=info.find('dd').text
#将信息添加到字典中
info_dict[key]=value

#打印书籍的信息
print(info_dict)
#存储每本书籍的信息
data_list.append(info_dict)

#页码page_number自增
page_number+=1

#新建csv文件存储书籍信息
with open(r'D:\PythonTest\风变python学习资料\Python爬虫\第5关爬取数据.csv','w',encoding='utf-8-sig') as f:
#将文件对象转换成DictWriter对象
writer=csv.DictWriter(f,fieldnames=['书名','作者','出版社','ISBN','页数','出版年','定价'])
#写入表头与数据
writer.writeheader()
writer.writerows(data_list)
```

网络爬虫的流程有三个步骤：获取网页、解析网页以及存储数据。
这个项目也可以划分出这三个步骤，我用图片给你展示爬取流程：



该爬取流程中有以下四个难点：
1）如何批量获取网站书籍列表页前 3 页的网页链接；
2）如何提取网站书籍列表页前 3 页所有书籍的名字和链接；

3）如何提取网站书籍列表页前 3 页所有书籍的详细信息；

4）如何存储爬取下来的书籍信息。

为了让你更好的理解代码，我将每个步骤对应实现的代码标注了出来，你仔细观察观察。

1.4 功能拆解

2. 课前复习

【第一题】字典的使用

已知字典 person_1 = {'姓名': '陈知枫', '身高': 175.5, '部门': '运营部'}。

请根据以下要求实现程序：

一、使用字典添加键值对的方式，为空字典 person_2 添加数据：

①、姓名：南大川；②、身高：180.0；③、部门：技术部；

二、再使用列表的 append() 将两个字典添加至空列表 data_list 中，最后打印列表 data_list。

```
# 复习
'''一、使用字典添加键值对的方式，为空字典 person_2 添加数据：
①、姓名：南大川；②、身高：180.0；③、部门：技术部；
二、再使用列表的 append() 将两个字典添加至空列表 data_list 中，最后打印列表 data_list。'''
data_list = []
person_1 = {'姓名': '陈知枫', '身高': 175.5, '部门': '运营部'}
person_2 = {}
# 根据题目要求实现功能
```

答案：

```
# 复习
'''一、使用字典添加键值对的方式，为空字典 person_2 添加数据：
①、姓名：南大川；②、身高：180.0；③、部门：技术部；

二、再使用列表的 append() 将两个字典添加至空列表 data_list 中，最后打印列表 data_list。'''

data_list = []
person_1 = {'姓名': '陈知枫', '身高': 175.5, '部门': '运营部'}
person_2 = {}
# 根据题目要求实现功能
person_2['姓名'] = '南大川'
person_2['身高'] = 180.0
person_2['部门'] = '技术部'
data_list.append(person_1)
data_list.append(person_2)

data_list
```

```
[{'姓名': '陈知枫', '身高': 175.5, '部门': '运营部'},
 {'姓名': '南大川', '身高': 180.0, '部门': '技术部'}]
```

【第二题】while 循环相关的练习

请使用 while 循环以及增强赋值 “-=” 实现功能：

依次打印出倒计时 5 秒、倒计时 4 秒、 …、倒计时 1 秒。

【提示】number -= 1 等价于 number = number - 1。

```python
'''【第二题】while 循环相关的练习


请使用 while 循环以及增强赋值 “-=” 实现功能：
依次打印出倒计时 5 秒、倒计时 4 秒、 …、倒计时 1 秒。
【提示】number -= 1 等价于 number = number - 1。'''

number = 5
while number > 0:

    print(f'倒计时 {number} 秒')
    number -= 1

print('倒计时结束')
```

✓ 0.3s

```
倒计时 5 秒
倒计时 4 秒
倒计时 3 秒
倒计时 2 秒
倒计时 1 秒
倒计时结束
```

```python
'''【第二题】for 循环相关的练习


请使用 while 循环以及增强赋值 “-=” 实现功能：
依次打印出倒计时 5 秒、倒计时 4 秒、 …、倒计时 1 秒。
【提示】number -= 1 等价于 number = number - 1。'''

import time
for number in range(5,0,-1):

    print(f'倒计时 {number} 秒')
    time.sleep(1)


print('倒计时结束')
```

✓ 5.4s

```
倒计时 5 秒
倒计时 4 秒
倒计时 3 秒
倒计时 2 秒
倒计时 1 秒
倒计时结束
```

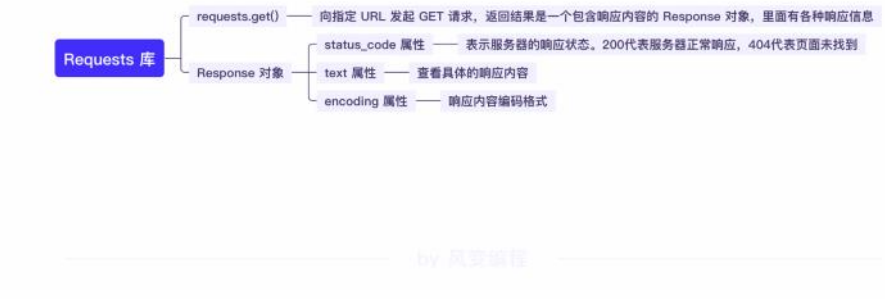
ok，复习就先到这里。

我们开始学习今天的项目代码。

无论是提取网站书籍列表页前 3 页的所有书籍名和书籍链接，还是提取书籍的详细信息，都需要我们先对网页所在服务器发送请求，这就需要网络爬虫的第一步：获取网页。

3. 获取网页

前面课程学过的 Requests 库内置了很多函数来帮我们实现各种方法的网络请求。像 request.get() 就是 Requests 库中用来发起 GET 请求的函数。通过 requests.get() 函数返回的 Response 对象调用属性，可以获得具体的响应信息。例如 Response.status_code 属性就是响应状态码；Response.text 属性用来查看具体的响应内容；Response.encoding 属性就是响应内容的编码格式。



在提取网站书籍列表页前 3 页的书籍名和书籍链接前，需要验证网页是否能够正常响应。
先看看网站书籍列表页前 3 页的网页链接，它们分别为：
第 1 页：https://wp.forchange.cn/resources/
第 2 页：https://wp.forchange.cn/resources/page/2/
第 3 页：https://wp.forchange.cn/resources/page/3/
都说自己动手，丰衣足食。
请你补充下面代码的第 10 ~ 12 、15 ~ 17 行，通过 requests.get() 函数分别请求网站书籍列表页前 3 页的网页链接，并打印一下它们对应的响应状态码。

```
# 导入模块 requests
import requests
# 设置网站前 3 页的网页链接
url_1 = 'https://wp.forchange.cn/'
url_2 = 'https://wp.forchange.cn/resources/page/2/'
url_3 = 'https://wp.forchange.cn/resources/page/3/'
# 分别请求网站前 3 页的网页链接


# 分别打印网站前 3 页的响应状态码
```

```
1  # 导入模块 requests
2  import requests
3
4  # 设置网站前 3 页的网页链接
5  url_1 = 'https://wp.forchange.cn/'
6  url_2 = 'https://wp.forchange.cn/resources/page/2/'
7  url_3 = 'https://wp.forchange.cn/resources/page/3/'
8
9  # 分别请求网站前 3 页的网页链接
10 url_1_res = requests.get(url_1)
11 url_2_res = requests.get(url_2)
12 url_3_res = requests.get(url_3)
13
14 # 分别打印网站前 3 页的响应状态码
15 print(url_1_res.status_code)
16 print(url_2_res.status_code)
17 print(url_3_res.status_code)
18
✓ 1.1s
200
200
200
```

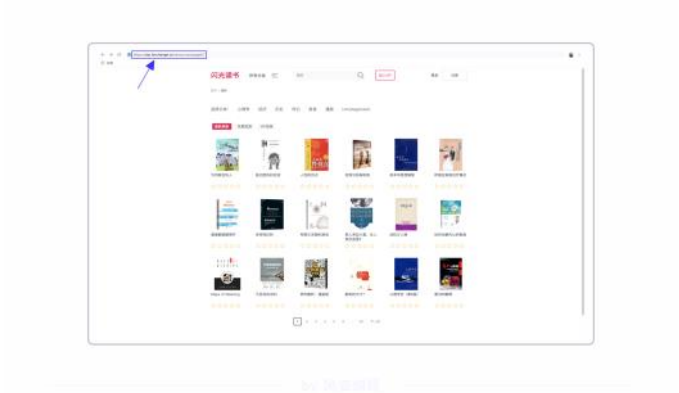
虽然成功取到了网页的响应状态码，但是这代码看上去就很冗余。
为了减轻代码量，我们一般会使用循环来替代。这里我推荐使用 while 循环语句来简化上方的代码。

仔细对比一下下方的三个网页链接：

```
url_1 = 'https://wp.forchange.cn/resources/'
url_2 = 'https://wp.forchange.cn/resources/page/2/'
url_3 = 'https://wp.forchange.cn/resources/page/3/'
```

很明显可以看出，三个网页链接的前半部分是一样的，都包含：`https://wp.forchange.cn/resources/` 。

三者的主要区别在于后半部分。其中 `url_2` 和 `url_3` 的区别在于链接尾部的数字不同，而 `url_1` 缺失这后半部分的内容。
我们可以做个假设：网页链接里 ’page/’ 后的数字决定了我们的页数。
然后做个验证：在 `url_1` 网页链接的尾部加上 ’page/1/’ 再去访问该网页，看看是否还是同一个网页：
https://wp.forchange.cn/resources/page/1/。



事实证明，网站书籍列表页第一页的链接尾部有没有加上 ’page/1/’，访问的都是同一个网页。以我的爬虫经验来说，有很多分页的网站都会省去第一页的
页码，所以遇到这种情况我们不妨可以大胆试试看。

现在给 `url_1` 补上 `page/1/` 以后，三个网页的规律就很明显了：

```
url_1 = 'https://wp.forchange.cn/resources/page/1/'
url_2 = 'https://wp.forchange.cn/resources/page/2/'
url_3 = 'https://wp.forchange.cn/resources/page/3/'
```

摸清网站书籍列表页前 3 页的规律后，我们就可以使用 `while` 循环语句配合字符串拼接的知识点，去实现批量获取书籍列表页前 3 页的网页链接并打印响
应状态码。运行我的代码看看吧：

```python
# sourcery skip: remove-empty-nested-block, remove-zero-from-range
import requests
for pagenumber in range(1,4):
    url = f'https://segmentfault.com/blog/track/{pagenumber}'

    url_res = requests.get(url)
    print(url_res.status_code)

```

✓ 0.4s

```
200
200
200
```

```python
import requests
page_number = 1
while page_number <=3:
    url = f'http://www.qiushibaike.com/hot/page/{page_number}'
    res = requests.get(url)
    page_number += 1
    print(res.status_code)
```

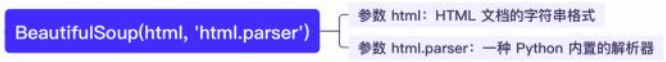
✓ 2.2s

```
200
200
200
```

上方代码中，我们设置了值为 1 的页码 page_number。
通过条件为 page_number < 4 的 while 循环语句，以及 page_number 的自增，使 while 循环语句的循环体执行了三次。
每次执行循环体的内容，会将页码与网址字符串拼接为一个完整的网页链接。
再通过 requests.get() 函数向网页发起请求，最后执行打印语句打印出三个网页的响应码。
从终端的显示内容可以看到，三个网页的响应状态码都是 200，即服务器正常响应请求。
到这里，我们已经成功获取网站书籍列表页前 3 页的网页链接，接下来可以对这几页中的书籍名和书籍链接进行提取，此时就到网络爬虫的第二步：解析网页。

4. 解析网页

要从 HTML 文档中提取数据，可以使用 bs4 库里的 BeautifulSoup 类对 HTML 文档进行解析。
在调用该类前我们需要先导入该类，其语法为：from bs4 import BeautifulSoup。
导入成功后，调用类 BeautifulSoup() 解析网页，得到 BeautifulSoup 对象。
调用该类时需要传入两个参数，第一个参数可以是一个字符串格式的 HTML 文档（Response.text）。第二个参数是解析器（本节课依旧使用解析器 html.parser）。

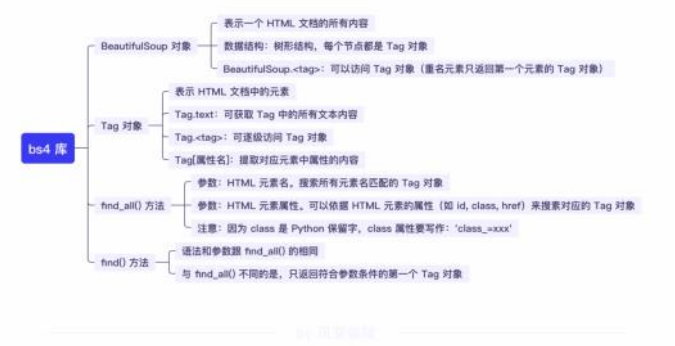


现在我们以网站书籍列表页的第 1 页（https://wp.forchange.cn/resources/page/1/）为例，请你按以下要求补充代码的第 10、12 行：
一、使用 BeautifulSoup 对网页内容进行解析；
二、把调用类得到的 BeautifulSoup 对象赋给变量 bs，并打印变量 bs。

```
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置网站书籍列表页第 1 页的链接
url = 'https://wp.forchange.cn/resources/page/1/'
# 请求网页
res = requests.get(url)
# 解析网页内容得到 BeautifulSoup 对象，赋给变量 bs


# 打印变量 bs
```

```
1  # 导入库 requests 以及类 BeautifulSoup
2  import requests
3  from bs4 import BeautifulSoup
4
5  # 设置网站书籍列表页第 1 页的链接
6  url = 'https://wp.forchange.cn/resources/page/1/'
7  # 请求网页
8  res = requests.get(url)
9  # 解析网页内容得到 BeautifulSoup 对象，赋给变量 bs
10 bs = BeautifulSoup(res.text, 'html.parser')
11
12 # 打印变量 bs
13 bs
✓ 0.7s
Output exceeds the size limit. Open the full output data in a text editor
<!DOCTYPE html>

<html lang="zh-CN">
<head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<title>所有资源 - 闪光读书</title>
<link href="https://wp.forchange.cn/feed/" rel="alternate" title="闪光读书
type="application/rss+xml"/>
<link href="https://wp.forchange.cn/comments/feed/" rel="alternate" title=
```

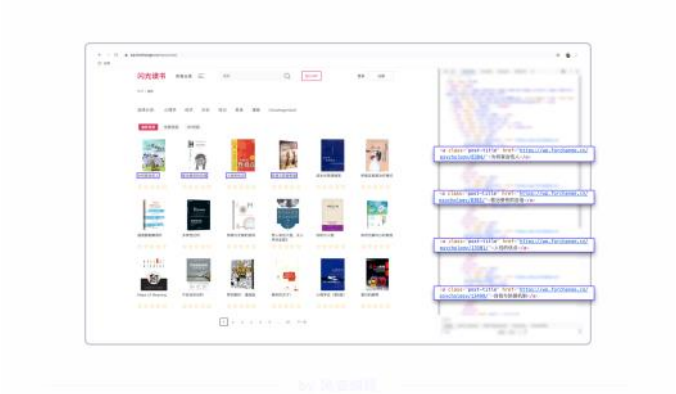
代码中的第 8 行，我们使用 requests.get() 函数向网站书籍列表页的第 1 页发起请求，把返回的 Response 对象赋给变量 res。
第 10 行执行 BeautifulSoup(res.text, 'html.parser') 得到 BeautifulSoup 对象，再赋给变量 bs。其中 res.text 得到网页的 HTML 文档内容，作为第一个参数传入，第二个参数依旧使用 html.parser 解析器。
程序运行后在终端显示出 BeautifulSoup 对象的内容，但我们的目的是提取书籍的名字和链接，很显然终端显示的并不是我们想要的最终数据。
此时可以使用 BeautifulSoup 对象的 find() 或 find_all() 方法，辅助我们定位书籍名和书籍链接的位置。
先给你简单回顾这两个方法：
find() 方法和 find_all() 方法有通用的两个参数。第一个参数 HTML 元素名用以搜索所有元素名匹配的 Tag 对象；第二个参数 HTML 元素属性可以依据 HTML 元素的属性搜索对应的 Tag 对象。
但 find_all() 方法和 find() 方法有个不同点在于，find() 方法只返回符合参数条件的第一个 Tag 对象。



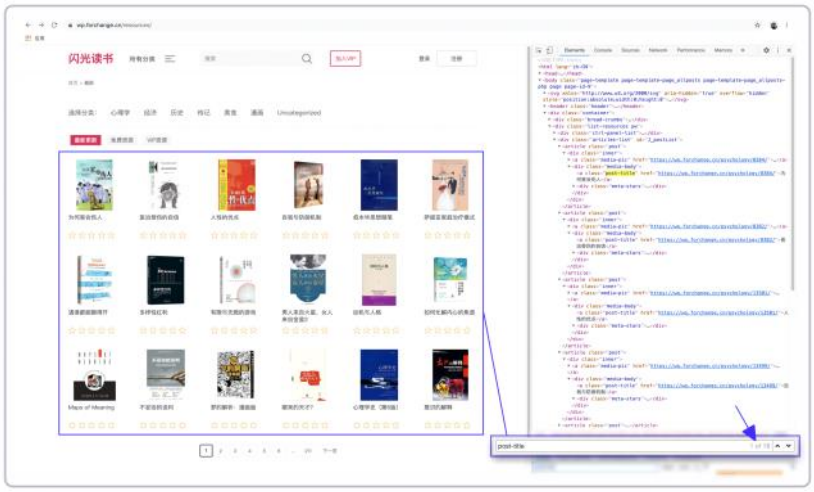
现在我们观察网页，看看使用哪一个方法爬取书籍名和书籍链接会更合适。

4.1 提取书籍名和书籍链接

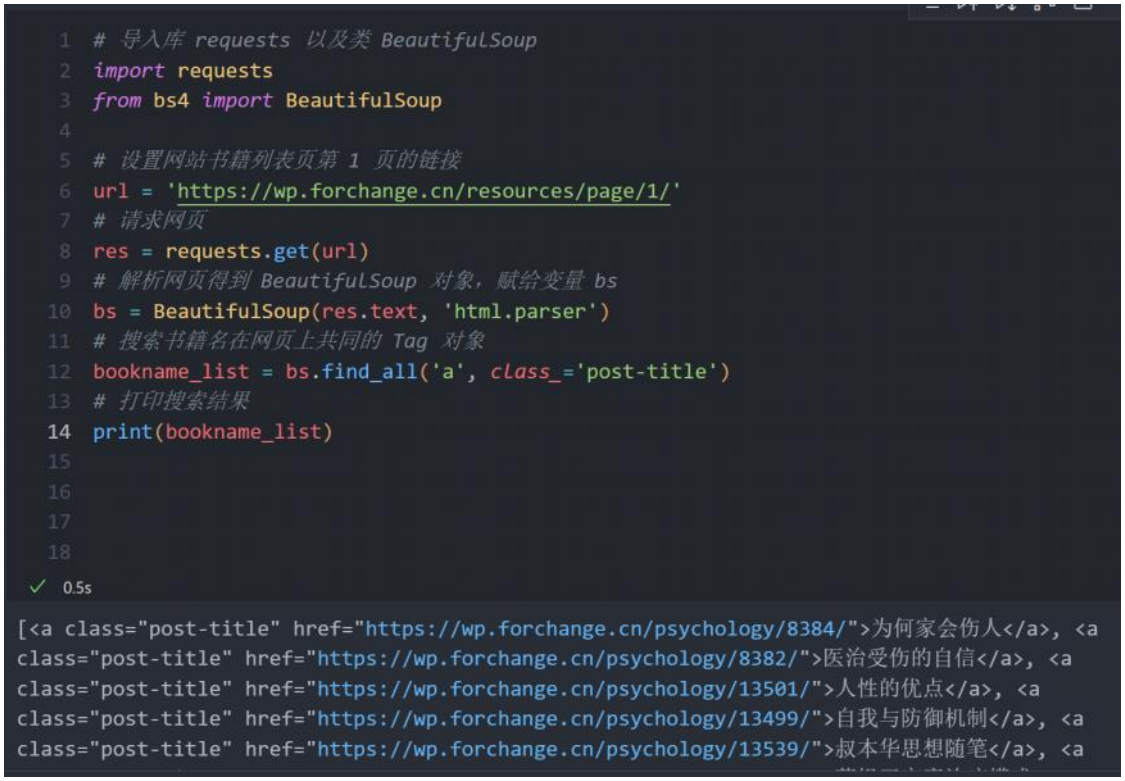
同样打开浏览器的开发者工具，并使用指针工具依次点击网页上的书名。



可以看到所有的书籍名和书籍链接都在属性为 class='post-title' 的 a 元素里。
这时我们再点击右侧的 HTML 文档，然后按 Ctrl+F（Mac 用户的快捷键为 command + F）打开搜索工具，在搜索框中输入 post-title。可以看到 'post-title' 的数量为 18 ，和该页面上书籍的数量一致。即 a 元素的 class='post-title' 属性定位到的全是书籍的信息。



所以，这里使用 find_all() 定位所有书籍名和书籍链接的位置，会更加方便。
你可以看看下方代码，主要留意代码中的第 12 行:



代码中第 12 行，通过调用 BeautifulSoup 对象的 find_all() 方法查询包含书籍名的元素。
上面我们也分析过，所有书籍名和书籍链接都在属性为 class='post-title' 的 a 元素内，所以 find_all() 方法内的第一个参数传入的是 a 元素，第二个参数传入元素的属性 class='post-title'。
程序运行后，可以看到终端显示了一个类似列表的数据，这个数据由 Tag 对象组成。
但数据还是没能精确显示出每个单独的书籍名和书籍链接，因为书籍名和书籍链接都在元素的内容中。
所以我们需要将得到的类似列表的数据通过 for 循环语句遍历，再使用 Tag.text 属性去获取书名、以及 Tag['属性'] 提取书籍链接。
请你根据下方代码的注释提示，补充第 15、17、19 行的代码吧:

```
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置网站书籍列表页第 1 页的链接
url = 'https://wp.forchange.cn/resources/page/1/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中所有包含书籍名和书籍链接的 Tag
bookname_list = bs.find_all('a', class_='post-title')
# 遍历搜索结果，提取并打印书籍名和书籍链接
```

```python
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup

# 设置网站书籍列表页第 1 页的链接
url = 'https://wp.forchange.cn/resources/page/1/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中所有包含书籍名和书籍链接的 Tag
bookname_list = bs.find_all('a', class_='post-title')

# 遍历搜索结果，提取并打印书籍名和书籍链接
for bookname in bookname_list:
    # 使用 Tag.text 属性提取书籍名，并打印书籍名
    print(bookname.text)
    # 使用 Tag['属性名'] 提取书籍链接，并打印书籍链接
    print(bookname['href'])
```

```
Output exceeds the size limit. Open the full output data in a text editor
为何家会伤人
https://wp.forchange.cn/psychology/8384/
医治受伤的自信
https://wp.forchange.cn/psychology/8382/
人性的优点
https://wp.forchange.cn/psychology/13501/
自我与防御机制
https://wp.forchange.cn/psychology/13499/
叔本华思想随笔
https://wp.forchange.cn/psychology/13539/
萨提亚家庭治疗模式
https://wp.forchange.cn/psychology/13497/
遇谁都能聊得开
https://wp.forchange.cn/psychology/13495/
多样性红利
https://wp.forchange.cn/psychology/13493/
有限与无限的游戏
https://wp.forchange.cn/psychology/13491/
男人来自火星，女人来自金星2
```

```python
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置网站书籍列表页第 1 页的链接
url = 'https://wp.forchange.cn/resources/page/1/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中所有包含书籍名和书籍链接的 Tag
bookname_list = bs.find_all('a', class_='post-title')
# 遍历搜索结果，提取并打印书籍名和书籍链接
for bookname in bookname_list:
    # 使用 Tag.text 属性提取书籍名，并打印书籍名
    print(bookname.text)
    # 使用 Tag['属性名'] 提取书籍链接，并打印书籍链接
    print(bookname['href'])
```

代码中第 17 行，将 for 循环语句遍历出的每个 Tag 对象调用 text 属性，提取网站书籍列表页第 1 页的所有书籍名。
第 19 行，由于书籍链接与书籍名在同一个 a 元素里，书籍链接存储在元素的属性 href 上，所以使用 Tag['属性名'] 提取属性的内容。
提取到书籍的链接，就可以使用 requests.get() 请求书籍详情页，提取书籍的详细数据。

4.2 提取书籍的详细信息

关于书籍详细信息的提取，相信你还有印象，现在我给你大致过一遍。

我们还是使用书籍《人性的优点》为例： https://wp.forchange.cn/psychology/13501/ ，看看该书籍详情页的 HTML 文档内容。

这里对书籍详细信息的提取，需要配合使用 find() 和 find_all() 方法。

书籍详细信息整体都在同一个 div 元素内，所以我们先使用 find('div', class_='res-attrs') 定位各项信息共同所在的元素。

而 div 元素里存在多个 dl 元素，dl 元素里存储着书籍的各项信息，我们可以使用 find_all('dl') 找到符合参数条件的所有 Tag。

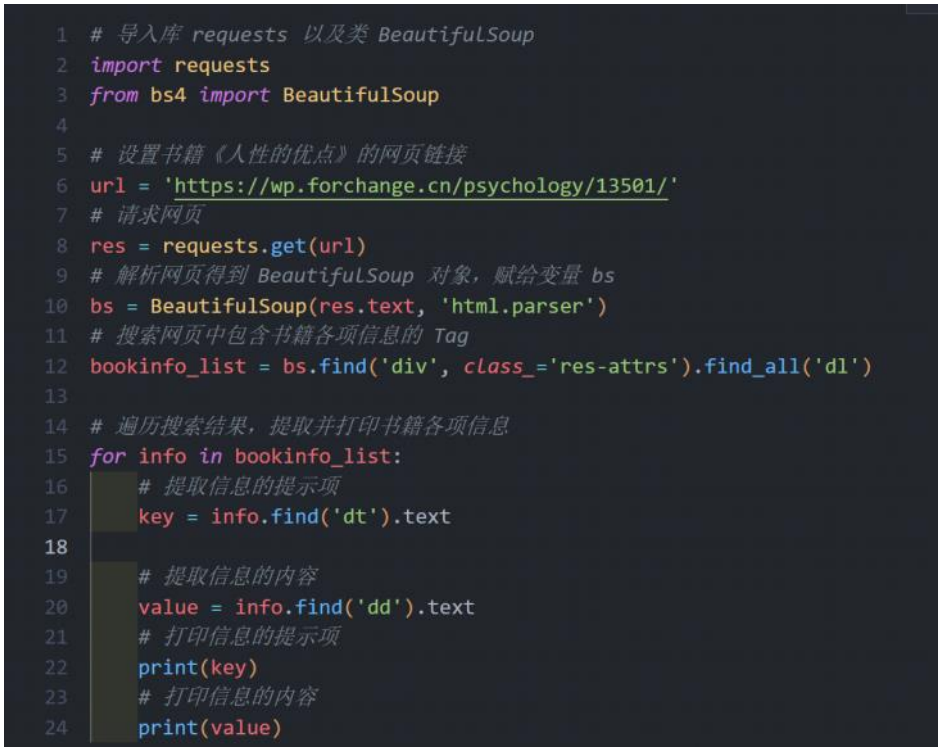
通过 for 循环语句遍历 find_all() 返回的结果，再使用 find() 方法分别定位 dt 元素和 dd 元素，可以提取书籍详细信息的提示项和信息的内容。

昨天刚学完的内容，相信你还有印象。请你来补充下方代码的第 12、17、19 行吧：

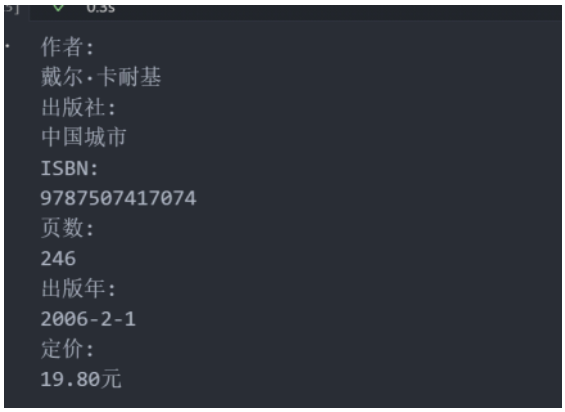
【提示】find() 和 find_all() 可以连用，例如 find('xxx').find_all('yyy')。

```
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list =
# 遍历搜索结果，提取并打印书籍各项信息
for info in bookinfo_list:
    # 提取信息的提示项
    key =
    # 提取信息的内容
    value =
    # 打印信息的提示项
    print(key)
    # 打印信息的内容
    print(value)
```

答案



```
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup

# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list = bs.find('div', class_='res-attrs').find_all('dl')

# 遍历搜索结果，提取并打印书籍各项信息
for info in bookinfo_list:
    # 提取信息的提示项
    key = info.find('dt').text

    # 提取信息的内容
    value = info.find('dd').text
    # 打印信息的提示项
    print(key)
    # 打印信息的内容
    print(value)
```

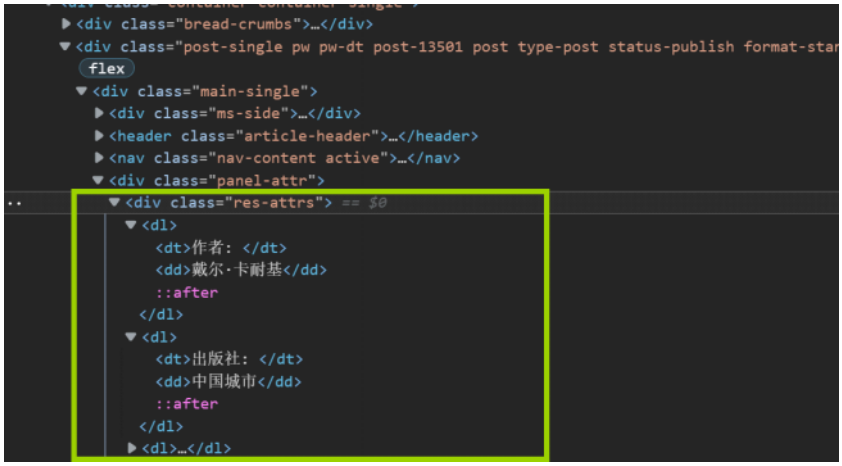


```
作者:
戴尔·卡耐基
出版社:
中国城市
ISBN:
9787507417074
页数:
246
出版年:
2006-2-1
定价:
19.80元
```


```
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list = bs.find('div', class_='res-attrs').find_all('dl')
# 遍历搜索结果，提取并打印书籍各项信息
for info in bookinfo_list:
    # 提取信息的提示项
    key = info.find('dt').text
    # 提取信息的内容
    value = info.find('dd').text
    # 打印信息的提示项
    print(key)
    # 打印信息的内容
    print(value)
```

```python
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup

# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list = bs.find('div', class_='res-attrs').find_all('dl')

# 遍历搜索结果，提取并打印书籍各项信息
for info in bookinfo_list:
    # 提取信息的提示项
    key = info.find('dt').text
    # 提取信息的内容
    value = info.find('dd').text
    # 打印信息的提示项
    print(key)
    # 打印信息的内容
    print(value)
```

返回了一个包含书籍各项信息的数据

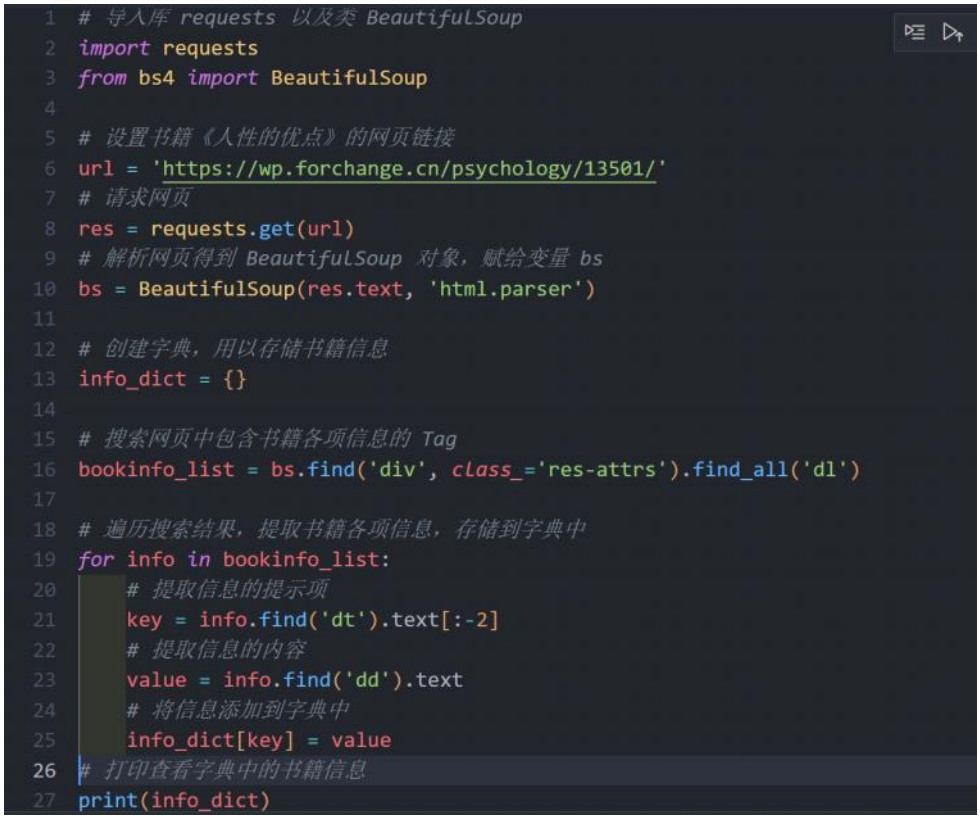


代码第 15 ~ 23 行为了精确提取各项信息的数据，我们使用 for 循环语句对 find_all() 的返回结果进行遍历。

我们也分析过，信息的提示项在 dt 元素中，我们使用 find('dt') 进行定位，再通过 text 属性取出。(小提示：这里的 [:-2] 是为了去除多余的内容：'：')

而信息的内容在 dd 元素中，我们同样使用 find() 方法以及 Tag.text 属性取出。

由于提取的数据间有对应关系，我们一般会使用字典来存储，以下是我的优化代码，你重点要看第 13、25 行，然后运行程序：



```python
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup

# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')

# 创建字典，用以存储书籍信息
info_dict = {}

# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list = bs.find('div', class_='res-attrs').find_all('dl')

# 遍历搜索结果，提取书籍各项信息，存储到字典中
for info in bookinfo_list:
    # 提取信息的提示项
    key = info.find('dt').text[:-2]
    # 提取信息的内容
    value = info.find('dd').text
    # 将信息添加到字典中
    info_dict[key] = value
# 打印查看字典中的书籍信息
print(info_dict)
```


```python
# 导入库 requests 以及类 BeautifulSoup
import requests
from bs4 import BeautifulSoup
# 设置书籍《人性的优点》的网页链接
url = 'https://wp.forchange.cn/psychology/13501/'
# 请求网页
res = requests.get(url)
# 解析网页得到 BeautifulSoup 对象，赋给变量 bs
bs = BeautifulSoup(res.text, 'html.parser')
# 创建字典，用以存储书籍信息
info_dict = {}
# 搜索网页中包含书籍各项信息的 Tag
bookinfo_list = bs.find('div', class_='res-attrs').find_all('dl')
# 遍历搜索结果，提取书籍各项信息，存储到字典中
for info in bookinfo_list:
    # 提取信息的提示项
    key = info.find('dt').text[:-2]
    # 提取信息的内容
    value = info.find('dd').text
    # 将信息添加到字典中
    info_dict[key] = value
# 打印查看字典中的书籍信息
print(info_dict)
```


第 13 行我们创建了一个空字典 info_dict，用以存储单本书籍的各项信息。
第 25 行，我们将书籍信息的提示项作为字典的键、信息的内容作为字典的值存储至字典 info_dict 中。

从终端的显示内容，可以看到我们已经成功将书籍《人性的优点》的信息打印出来。

至此我们已经学习了三个步骤：

1）批量获取网站前 3 页的网页链接；

2）提取网站书籍列表页第 1 页所有书籍的名字和书籍链接；

3）通过单本书籍的链接提取该书籍的详细信息。

现在我们需要将这三块的代码做一个合并，实现功能：提取网站第 3 页所有书籍的信息。

4.3 提取网站前 3 页的书籍信息

```python
import requests
from bs4 import BeautifulSoup

# 设置页码 page_number
page_number = 1

# while 循环的条件设置为 page_number 的值是否小于 4
while page_number < 4:
    # 设置要请求的网页链接
    url = 'https://wp.forchange.cn/resources/page/' + str(page_number)
    # 请求网页
    books_list_res = requests.get(url)
    # 解析请求到的网页内容
    bs = BeautifulSoup(books_list_res.text, 'html.parser')
    # 搜索网页中所有包含书籍名和书籍链接的 Tag
    href_list = bs.find_all('a', class_='post-title')

    # 使用 for 循环遍历搜索结果
    for href in href_list:
        # 创建字典，用以存储书籍信息
        info_dict = {}
        # 提取书籍名
        info_dict['书名'] = href.text
        # 提取书籍链接
        book_url = href['href']
        # 通过书籍链接请求书籍详情页
        book_list_res = requests.get(book_url)
```

```
        # 解析书籍详情页的内容
        new_bs = BeautifulSoup(book_list_res.text, 'html.parser')
        # 搜索网页中所有包含书籍各项信息的 Tag
        info_list = new_bs.find('div', class_='res-attrs').find_all('dl')

        # 使用 for 循环遍历搜索结果
        for info in info_list:
            # 提取信息的提示项
            key = info.find('dt').text[:-2]
            # 提取信息的内容
            value = info.find('dd').text
            # 将信息添加到字典中
            info_dict[key] = value

        # 打印字典中的书籍信息
        print(info_dict)

    # 页码 page_number 自增
    page_number += 1
```

✓ 17.9s

```
Output exceeds the size limit. Open the full output data in a text editor
{'书名': '为何家会伤人', '作者': '武志红', '出版社': '北京联合出版公司', 'ISBN': '9787550230491', '
数': '328', '出版年': '2014/7/1', '定价': '39.80元'}
{'书名': '医治受伤的自信', '作者': '[法]弗雷德里克•方热著', '出版社': '生活·读书·新知三联书店，生活书
版有限公司', 'ISBN': '9787807681427', '页数': '320', '出版年': '2016/8/1', '定价': '42'}
{'书名': '人性的优点', '作者': '戴尔·卡耐基', '出版社': '中国城市', 'ISBN': '9787507417074', '页数'
'246', '出版年': '2006-2-1', '定价': '19.80元'}
{'书名': '自我与防御机制', '作者': '[奥]安娜·弗洛伊德', '出版社': '华东师范大学出版社', 'ISBN':
'9787567575189', '页数': '160', '出版年': '2018-11-1', '定价': '25.00元'}
{'书名': '叔本华思想随笔', '作者': '[德]阿图尔·叔本华', '出版社': '上海人民出版社', 'ISBN':
'9787208081185', '页数': '328', '出版年': '2008-10-1', '定价': '34.00元'}
{'书名': '萨提亚家庭治疗模式', '作者': '（美）萨提亚', '出版社': '世界图书出版公司', 'ISBN':
'9787506286558', '页数': '349', '出版年': '2007-6-1', '定价': '36.00元'}
{'书名': '遇谁都能聊得开', '作者': '[美]莉尔·朗兹', '出版社': '上海社会科学院出版社', 'ISBN':
```

方法2:

```
import requests
from bs4 import BeautifulSoup


book_dict = {}

for page_number in range(1,12):

    book_url = f'https://wp.forchange class_: _Strainable umber}'
    res_book = requests.get(book_url)
    bs_book = BeautifulSoup(res_book. class_: _Strainable
    book_list = bs_book.find_all('a',class_='post-title')
    for book in book_list:
        name = book.text
        link = book['href']


        res_book_link = requests.get(link)
        bs_book = BeautifulSoup(res_book_link.text,'html.parser')
        book_info = bs_book.find('div',class_='res-attrs').find_all('dl')
        for book_info_item in book_info:
            dt = book_info_item.find('dt').text[:-2]
            dd = book_info_item.find('dd').text
            book_dict[dt] = dd
        print(book_dict)
```

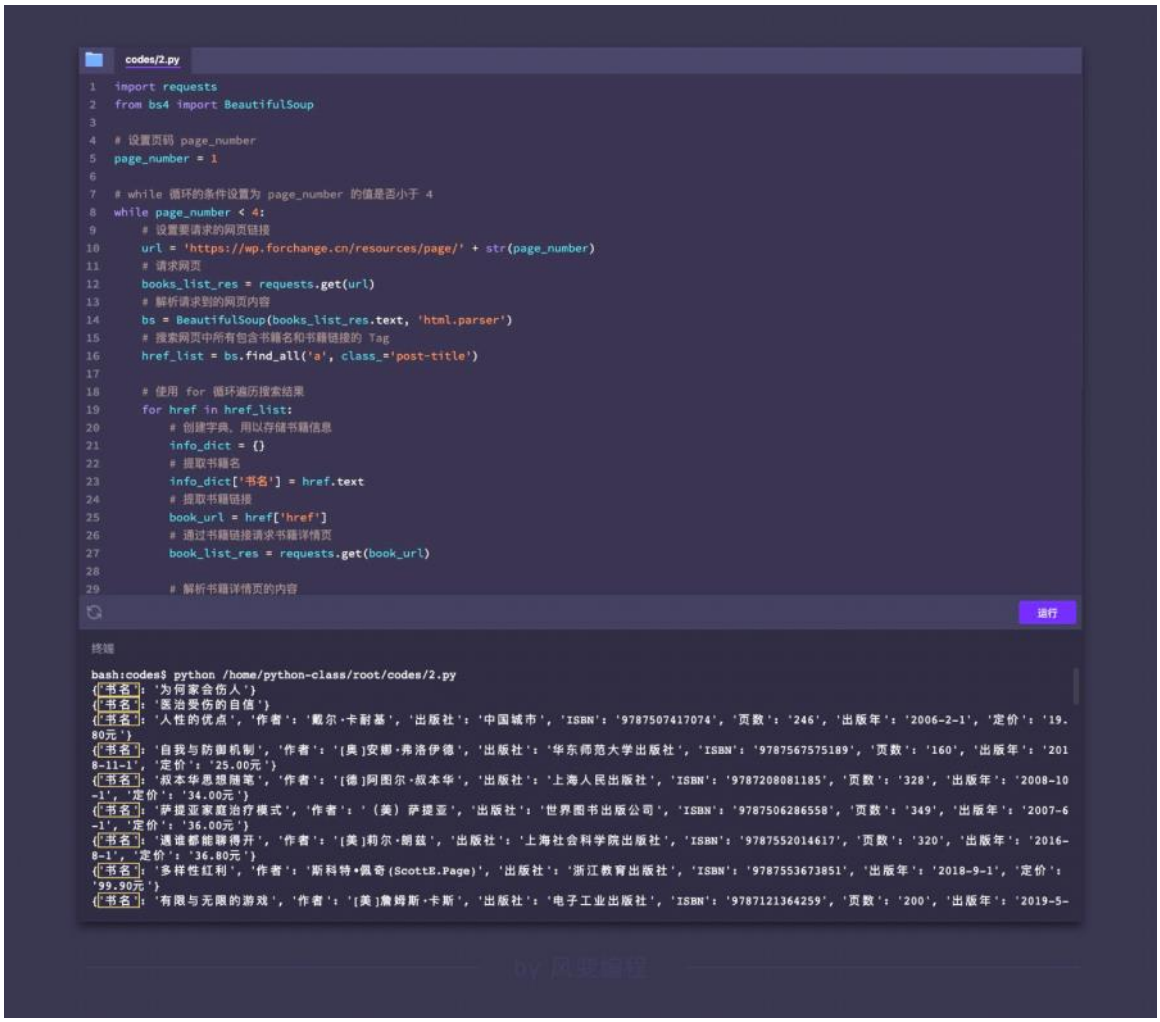
```
import requests
from bs4 import BeautifulSoup
```

```
book_dict = {}
for page_number in range(1,12):
    book_url = f'https://wp.forchange.cn/resources/{page_number}'
    res_book = requests.get(book_url)
    bs_book = BeautifulSoup(res_book.text,'html.parser')
    book_list = bs_book.find_all('a',class_='post-title')
    for book in book_list:
        name = book.text
        link = book['href']

        res_book_link = requests.get(link)
        bs_book = BeautifulSoup(res_book_link.text,'html.parser')
        book_info = bs_book.find('div',class_='res-attrs').find_all('dl')
        for book_info_item in book_info:
            dt = book_info_item.find('dt').text[:-2]
            dd = book_info_item.find('dd').text
            book_dict[dt] = dd
        print(book_dict)
```

5. 存储数据

网络爬虫的最后一步：存储数据。

从刚刚的显示结果可以看到，每本书籍的信息单独存放一个字典，但不同书籍信息的提示项都一样，且都是作为字典的键存储。



所以在存储上会更倾向于使用 csv 模块的 DictWriter()，接下来我们简单复习一下这个知识点。

调用 csv 模块中类 DictWriter 的语法为：csv.DictWriter(f, fieldnames)，执行后会得到一个 DictWriter 对象。语法中的**参数 f** 是 open() 函数打开的文件对象；**参数 fieldnames** 用来设置文件的表头。

得到的 DictWriter 对象可以调用 writeheader() 方法，将 fieldnames 写入 csv 的第一行。

再调用 writerows() 方法将多个字典写进 csv 文件中。

来道练习题让你复习一下。

已知两个列表：

① data_list = [{'姓名'\: '陈知枫', '性别': '男', '身高': '175.5', '部门': '运营部'}, {'姓名': '南大川', '性别': '男', '身高': '180', '部门': '技术部'}]

② headers = ['姓名', '性别', '身高', '部门']

请根据以下要求完成代码：

1）调用类 DictWriter() 的语法：DictWriter(f, fieldnames)；

2）使用 csv.DictWriter() 类的 writeheader() 方法将列表 headers 作为表头写入；

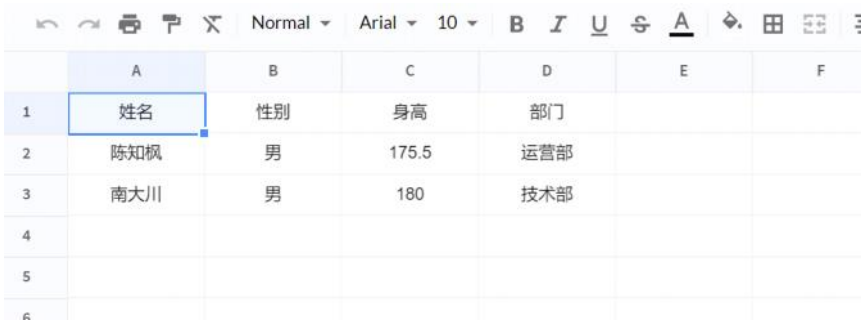
3）使用 csv.DictWriter() 类的 writerows() 方法把列表 data_list 中，各字典的所有值写进 info.csv 中。

```
# 导入 csv 模块
import csv
# 设置列表
data_list = [{'姓名': '陈知枫', '性别': '男', '身高': '175.5', '部门': '运营部'}, {'姓名': '南大川', '性别': '男', '身高': '180', '部门': '技术部'}]
# 设置表头
headers = ['姓名', '性别', '身高', '部门']
# 创建并打开 info.csv
with open(r'D:\PythonTest\风变python学习资料\Python爬虫\info.csv', 'w', encoding='utf-8-sig', newline='') as demo_file:
    # 补全代码下，实现题目要求
```

答案

```
# 导入 csv 模块
import csv

# 设置列表
data_list = [{'姓名': '陈知枫', '性别': '男', '身高': '175.5', '部门': '运营部'}, {'姓名': '南大川', '性别': '男', '身高': '180', '部门': '技术部'}]
# 设置表头
headers = ['姓名', '性别', '身高', '部门']

# 创建并打开 info.csv
with open(r'D:\PythonTest\风变python学习资料\Python爬虫\info.csv', 'w', encoding='utf-8-sig', newline='') as demo_file:
    # 补全代码下方代码，实现题目要求
    # 创建 csv 写入对象
    csv_writer = csv.DictWriter(demo_file, fieldnames=headers)
    csv_writer.writeheader()
    csv_writer.writerows(data_list)
```

```python
# 导入 csv 模块
import csv
# 设置列表
data_list = [{'姓名': '陈知枫', '性别': '男', '身高': '175.5', '部门': '运营部'}, {'姓名': '南大川', '性别': '男', '身高': '180', '部门': '技术部'}]
# 设置表头
headers = ['姓名', '性别', '身高', '部门']
# 创建并打开 info.csv
with open(r'D:\PythonTest\风变python学习资料\Python爬虫\info.csv', 'w', encoding='utf-8-sig', newline='') as demo_file:
    # 补全代码下方代码，实现题目要求
    # 创建 csv 写入对象
    csv_writer = csv.DictWriter(demo_file, fieldnames=headers)
    csv_writer.writeheader()
    csv_writer.writerows(data_list)
```



	A	B	C	D	E	F
1	姓名	性别	身高	部门		
2	陈知枫	男	175.5	运营部		
3	南大川	男	180	技术部		
4						
5						
6						

代码中的第 12 行调用 csv 模块中类的 DictWriter，将文件对象 demo_file 传给参数 f，并将列表 headers 作为文件的表头传给参数 fieldnames。

第 14 行将表头 headers 写入到 info.csv 文件中的第一行。
第 15 行将列表 data_list 中的各个字典的值，根据字典的键在 csv 文件中所在行写入。

现在我们要将爬取到的前 3 页书籍信息写入到 csv 文件中，完成最终代码。
我们先把提取网站书籍列表页前 3 页书籍信息的代码展示出来，再依次添加代码

```python
import requests
from bs4 import BeautifulSoup
import csv

book_item = []

for page_number in range(1,12):

    book_url = f'https://wp.forchange.cn/resources/{page_number}'
    res_book = requests.get(book_url)
    bs_book = BeautifulSoup(res_book.text,'html.parser')
    book_list = bs_book.find_all('a',class_='post-title')
    for book in book_list:
        book_dict = {}
        name = book.text
        link = book['href']
        book_dict['书名'] = name
        book_dict['链接'] = link
```

```python
        res_book_link = requests.get(link)
        bs_book = BeautifulSoup(res_book_link.text,'html.parser')
        book_info = bs_book.find('div',class_='res-attrs').find_all('dl')

        for book_info_item in book_info:
            dt = book_info_item.find('dt').text[:-2]
            dd = book_info_item.find('dd').text
            book_dict[dt] = dd
        print(book_dict)
        book_item.append(book_dict)
```

```python

with open(r'D:\PythonTest\风变python学习资料\Python爬虫\book_info.csv','w',encoding='utf-8_sig',newline='') as f:

    book_csv = csv.DictWriter(f,fieldnames=['书名','链接','作者', '出版社', 'ISBN', '页数', '出版年', '定价'])
    book_csv.writeheader()
    book_csv.writerows(book_item)

```

```
Output exceeds the size limit. Open the full output data in a text editor
{'书名': '为何家会伤人', '链接': 'https://wp.forchange.cn/psychology/8384/', '作者': '武志红', '出版社': '北京联合出版公司', 'ISBN': '9787550230491', '页数': '328', '出版年': '2014/7/1', '定价': '39.80元'}
{'书名': '医治受伤的自信', '链接': 'https://wp.forchange.cn/psychology/8382/', '作者': '[法]弗雷德里克•方热著', '出版社': '生活·读书·新知三联书店，生活书店出版有限公司', 'ISBN': '9787807681427', '页数': '320', '出版年': '2016/8/1', '定价': '42'}
{'书名': '人性的优点', '链接': 'https://wp.forchange.cn/psychology/13501/', '作者': '戴尔·卡耐基', '出版社': '中国城市', 'ISBN': '9787507417074', '页数': '246', '出版年': '2006-2-1', '定价': '19.80元'}
{'书名': '自我与防御机制', '链接': 'https://wp.forchange.cn/psychology/13499/', '作者': '[奥]安娜·弗洛伊德', '出版社': '华东师范大学出版社', 'ISBN': '9787567575189', '页数': '160', '出版年': '2018-11-1', '定价': '25.00元'}
{'书名': '叔本华思想随笔', '链接': 'https://wp.forchange.cn/psychology/13539/', '作者': '[德]阿图尔·叔本华', '出版社': '上海人民出版社', 'ISBN': '9787208081185', '页数': '328', '出版年': '2008-10-1', '定价': '34.00元'}
{'书名': '萨提亚家庭治疗模式', '链接': 'https://wp.forchange.cn/psychology/13497/', '作者': '（美）萨
```

```
import requests
from bs4 import BeautifulSoup
import csv
book_item = []
for page_number in range(1,12):
    book_url = f'https://wp.forchange.cn/resources/{page_number}'
    res_book = requests.get(book_url)
    bs_book = BeautifulSoup(res_book.text,'html.parser')
    book_list = bs_book.find_all('a',class_='post-title')
    for book in book_list:
        book_dict = {}
        name = book.text
        link = book['href']
        book_dict['书名'] = name
        book_dict['链接'] = link

        res_book_link = requests.get(link)
        bs_book = BeautifulSoup(res_book_link.text,'html.parser')
        book_info = bs_book.find('div',class_='res-attrs').find_all('dl')

        for book_info_item in book_info:
            dt = book_info_item.find('dt').text[:-2]
            dd = book_info_item.find('dd').text
            book_dict[dt] = dd
        print(book_dict)
        book_item.append(book_dict)

with open(r'D:\PythonTest\风变python学习资料\Python爬虫\book_info.csv','w',encoding='utf-8_sig',newline='') as f:
    book_csv = csv.DictWriter(f,fieldnames=['书名','链接','作者', '出版社', 'ISBN', '页数', '出版年', '定价'])
    book_csv.writeheader()
    book_csv.writerows(book_item)
```

6. 程序实现与总结

我们先回顾一下这个项目是要实现什么功能:

1）通过循环获取网站书籍列表页前 3 页的网页链接;

2）请求网页并解析网页内容，提取书籍名字和书籍链接;

3）通过书籍链接请求书籍详情页，提取书籍的详细信息;

4）将书籍的所有信息写进 csv 文件中。

ok，逻辑也梳理完了，下面请你独立完成这个项目吧。别担心，我会给你提示信息，辅助你完成项目。

6.2 知识归纳与总结

本节课主要学习了以下几个知识点:

1）网络爬虫的步骤

a.获取网页：顾名思义就是获取网页信息，在网络爬虫技术中这里获取的就是网页源代码;

b.解析网页：指的是从网页源代码中提取想要的数据;

c.存储数据：就是将提取到的数据存储下来。

2）Requests 库

a.requests.get() 函数向指定的 URL 发起 GET 请求，返回结果是一个包含响应内容的 Response 对象，里面有各种响应信息;

b.Response.status_code 属性就是响应状态码;

c.Response.text 属性用来查看具体的响应内容;

d.Response.encoding 属性就是响应内容的编码格式。

3）bs4 库

a.BeautifulSoup(html, 'html.parser') 语法中，第一个参数可以是一个内容为 HTML 文档的字符串，第二个参数是解析器，本节课依旧使用解析器 html.parser;

b.find_all() 方法和 find() 方法有通用的两个参数。第一个参数 HTML 元素名用以搜索所有元素名匹配的 Tag 对象；第二个参数 HTML 元素属性可以依据 HTML 元素的属性搜索对应的 Tag 对象;

c.Tag.text 可获取 Tag 中的所有文本内容;

d.Tag['属性名']可提取元素中对应属性的内容。

4）csv 模块

a.DictWriter(f, fieldnames)语法中，参数 f 为 open()函数打开的文件对象，参数 fieldnames 为必须参数，用来设置文件的表头；

b.writeheader()方法将 fieldnames 写入 csv 的第一行；

c.writerows(rows)方法中，参数 rows 是可迭代对象，该可迭代对象必须由字典组成。

以下是我们知识点的总结图：