第三章

1.实践案例

import requests

from bs4 import BeautifulSoup

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}

for x in range(10):

url = 'https://movie.douban.com/top250?start=' + str(x\*25) + '&filter='

res = requests.get(url, headers=headers)

bs = BeautifulSoup(res.text, 'html.parser')

tag\_num = bs.find\_all('div', class\_="item")

tag\_comment = bs.find\_all('div', class\_='star')

tag\_word = bs.find\_all('span', class\_='inq')

list\_all = [ ]

for x in range(len(tag\_num)):

if tag\_num[x].text[2:5] ==' ' or tag\_num[x].text[2:5] ==' ' or x >= len(tag\_word):

list\_movie=[tag\_num[x].text[2:5],tag\_num[x].find('img')['alt'], tag\_comment[x].text[2:5], tag\_num[x].find('a')['href'] ]

else:

list\_movie=[tag\_num[x].text[2:5],tag\_num[x].find('img')['alt'], tag\_comment[x].text[2:5], tag\_word[x].text, tag\_num[x].find('a')['href']]

list\_all.append(list\_movie)

print(list\_all)

2.实验一

##首先需要安装“requests”和“BeautifulSoup4”两个2库，再导入调用所需的库。

import requests

from bs4 import BeautifulSoupres = requests.get('http://books.toscrape.com/')

##利用requests对象的get方法，传入请求地址。

html = res.text

soup = BeautifulSoup(html,'html.parser')

##解析数据

items = soup.find('ul',class\_='nav').find('ul').find\_all('li')

##需要逐层挖掘，首先找到第一层<ul class=”nav nav-list”>;再往下找到第二层<ul>;依次找到第三层<li>如下图所示for item in items:

kind = item.find('a')

##找到所需数据的最终层<a>

print(kind.text.strip(),'\n')

##利用strip（）去除特殊字符串，如空格，\n,\t等等

2.实验二

import requests

from bs4 import BeautifulSoup

res\_bookstore=requests.get('http://books.toscrape.com/catalogue/category/books/travel\_2/index.html')

bs\_bookstore = BeautifulSoup(res\_bookstore.text,'html.parser')

list\_books = bs\_bookstore.find\_all(class\_='product\_pod') ##第一层级for tag\_books in list\_books:

tag\_name = tag\_books.find('h3').find('a') ## 找到a标签需要提取两次，详细如图5

list\_star = tag\_books.find('p',class\_="star-rating")

## 这个p标签的class属性有两种：一种是"star-rating"，另外一种是具体的几星比如"Two"。详细如图6

tag\_price = tag\_books.find('p',class\_="price\_color")

print(tag\_name['title']) ## 这里用到了tag['属性名']提取属性值，提取出‘title’的属性值，详细如图7

print('star-rating:',list\_star['class'][1])

# 同样是用属性名提取属性值，用list\_star['class']提取出来之后是一个由两个值组成的列表，如："['star-rating', 'Two']"，我们最终要提取的是这个列表的第1个值："Two"。

print('Price:',tag\_price.text, end='\n'+'------'+'\n')

3.实验三

headers={'user-agent':'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}

for x in range(10):

url = 'https://movie.douban.com/top250?start=' + str(x\*25) + '&filter='

res = requests.get(url, headers=headers)

bs = bs4.BeautifulSoup(res.text, 'html.parser')

bs = bs.find('ol', class\_="grid\_view")

for titles in bs.find\_all('li'):

num = titles.find('em',class\_="").text

title = titles.find('span', class\_="title").text

comment = titles.find('span',class\_="rating\_num").text

url\_movie = titles.find('a')['href'] if titles.find('span',class\_="inq") != None:

tes = titles.find('span',class\_="inq").text

sheet.append([num, title, comment, tes, url\_movie])

# 把num, title, comment, tes和url\_movie写成列表，用append函数多行写入Excel

print(num + '.' + title + '——' + comment + '\n' + '推荐语：' + tes +'\n' + url\_movie)

else:

sheet.append([num, title, comment, None,url\_movie])

print(num + '.' + title + '——' + comment + '\n' +'\n' + url\_movie)

wb.save('movieTop250.xlsx')

# 最后保存并命名这个Excel文件