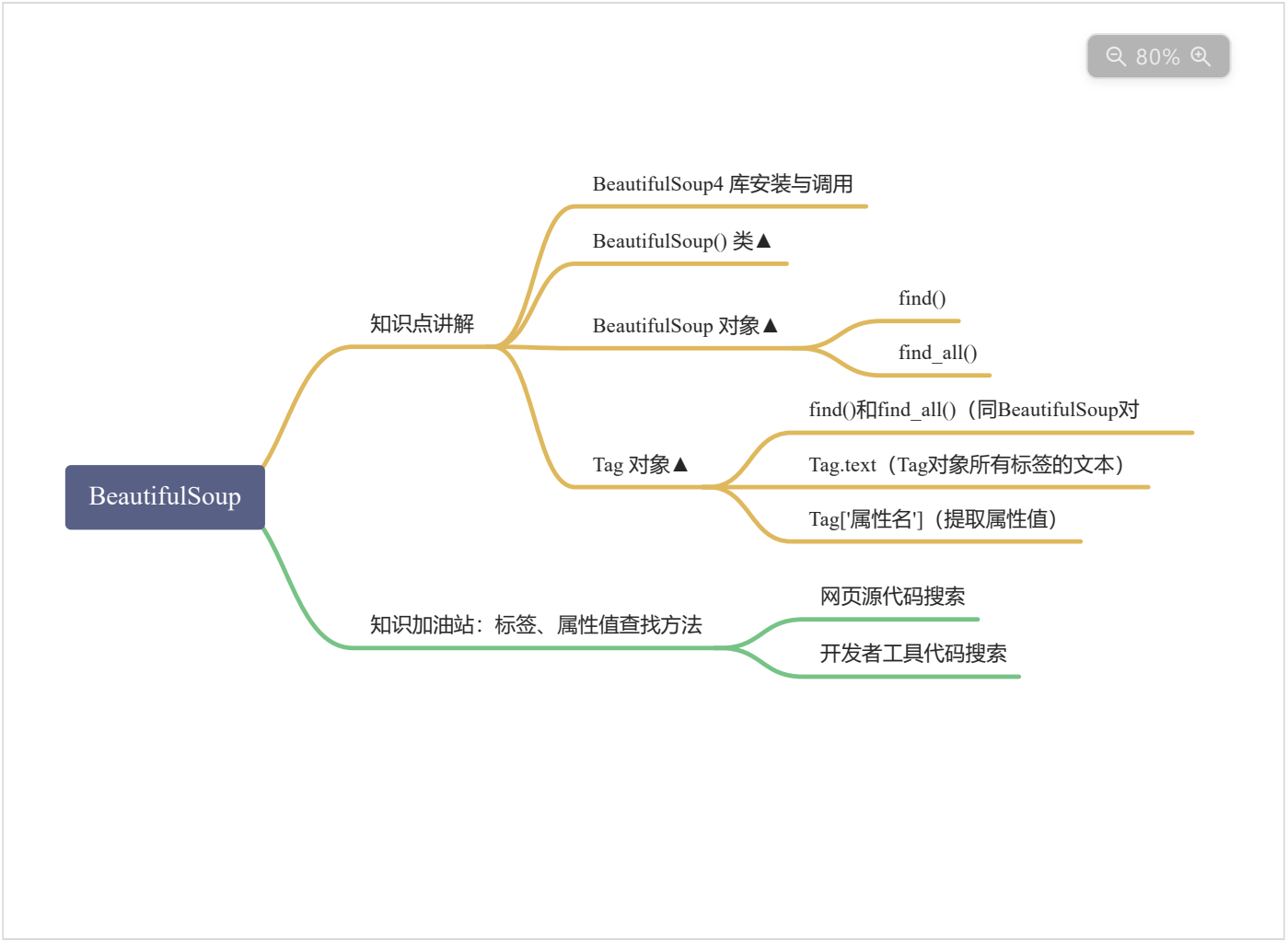
第2课 BeautifulSoup

一、课程结构导图

风变科技



注：▲为重点知识点。

二、知识点讲解

2.1 BeautifulSoup4 库安装

概念：BeautifulSoup4库是一个用于解析和提取网页数据的解析库。常用BeautifulSoup()类解析和提取网页内容。

安装语法：

语法

Windows：**pip install BeautifulSoup4**
Mac：**pip3 install BeautifulSoup4**

2.2 BeautifulSoup()类调用

应用场景：需要对请求到的数据进行解析和提取时，可调用BeautifulSoup() 类来实现。

调用方法：第一步：通过BeautifulSoup4 库调用BeautifulSoup() 类；第二步：给BeautifulSoup() 类传递两个参数，即**BeautifulSoup(html文本, '解析器')**，第0个参数是要被解析的文本，必须是**字符串**；第1个参数是解析器，我们使用Python的一个内置库：html.parser。 每次调用类之后会创建一个BeautifulSoup对象。

示例：

```
# 爬取《这个书苑不太冷》
import requests
from bs4 import BeautifulSoup   # 第一步，调用BeautifulSoup()类
res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/spider-men5.0.html') # 请求数据

soup = BeautifulSoup(res.text,'html.parser')    # 第二步，使用用BeautifulSoup()解析数据
print(type(soup))
# 结果：<class 'bs4.BeautifulSoup'>
```

2.3 BeautifulSoup对象

基本介绍：由BeautifulSoup()类生成的对象 ， 叫BeautifulSoup()对象，表示一个HTML文档的所有内容。（BeautifulSoup对象下文简称为BS或BS对象）

常用方法：

方法	功能	语法	示例	返回值
find()	提取满足需求的首个数据	BS对象.find(标签, 属性=属性值)	BS.find('div',class_='XXX')	Tag 对
find_all()	提取满足需求的所有数据	BS对象.find_all(标签, 属性=属性值)	BS.find_all('div',class_='XXX')	Tag 对
注意：示例中的class_，这里有一个下划线，是为了和python语法中的类 class区分，避免程序冲突；方法中的标签和属性，如果只用其中一个可以定位到，可以不用另外一个。				

注意：find_all() 方法的返回对象（<class 'bs4.element.ResultSet'>）是Tag对象集合，形如[tag1, tag2,tag3,.....]，可以使用遍历列或索引取值方法进行取值。

示例：

```
# 爬取《这个书苑不太冷》
import requests
from bs4 import BeautifulSoup
res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/spider-men5.0.html')
soup = BeautifulSoup(res.text,'html.parser')

# 通过匹配标签和属性提取我们想要的数据
item = soup.find(class_='books')        # 单独使用属性可以定位到数据，可以不用标签。
print(type(item))       # 打印item的数据类型

items = soup.find_all(class_='books')    # 单独使用属性可以定位到数据，可以不用标签。
print(type(items))      # 打印items的数据类型

item_0 = items[0]
print(type(item_0))     # 打印item_0的数据类型
```

```
# 结果:
# <class 'bs4.element.Tag'>
# <class 'bs4.element.ResultSet'>
# <class 'bs4.element.Tag'>
```

2.4 Tag 对象

基本介绍：一开始通过BS.find() 和BS.find_all() 获取Tag对象，获取到Tag 对象之后，也可以通过Tag.find()方法和Tag.find_all()方法获取数据内容。

常用方法：

方法	功能	语法	示例	返回值
find()	提取满足需求的首个数据	Tag对象.find(标签, 属性=属性值)	Tag.find('div',class_='XXX')	Tag 对
find_all()	提取满足需求的所有数据	Tag对象.find_all(标签, 属性=属性值)	Tag.find_all('div',class_='XXX')	Tag 对
注意：示例中的class_，这里有一个下划线，是为了和python语法中的类 class区分，避免程序冲突；方法中的标签和属性，如果只用其中一个可以定位到，可以不用另外一个。				

注意：find_all() 方法返回的对象（<class 'bs4.element.ResultSet'>）是Tag 对象集合，形如[tag1, tag2,tag3,.....]，可以使用列表的取值方法进行取值。

示例：

```python
# 爬取《这个书苑不太冷》
import requests
from bs4 import BeautifulSoup
res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/spider-men5.0.html')
soup = BeautifulSoup(res.text,'html.parser')    # BS对象

# 通过匹配标签和属性提取我们想要的数据
item = soup.find(class_='books')    # Tag对象；单独使用属性可以定位到数据，可以不用方法。

h2 = item.find('h2')    # Tag对象；单独使用标签可以定位到数据，可以不用属性。
print(type(h2))         # 打印h2的数据类型

a = item.find_all('a')
print(type(a))          # 打印a的数据类型

a_0 = item.find_all('a')[0]
print(type(a_0))        # 打印a_0的数据类型

# 结果:
# <class 'bs4.element.Tag'>
# <class 'bs4.element.ResultSet'>
# <class 'bs4.element.Tag'>
```

常用属性1：Tag.text 属性，该属性可以提取Tag 对象中所有标签中的字符串，可以通过父级提取子级的字符串。

示例1：

风变科技 风变科技

```python
# 爬取《这个书苑不太冷》
import requests
from bs4 import BeautifulSoup
res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/spider-men5.0.html')
soup = BeautifulSoup(res.text,'html.parser')

# 通过匹配标签和属性提取我们想要的数据
item = soup.find(class_='books')    # Tag对象

print(item.text)    # 打印Tag对象下所有字符
# 结果:
```

```
# \n科幻小说\n《奇点遗民》\n本书精选收录了刘宇昆的科幻佳作共22篇。《奇点遗民》融入了科幻艺术吸引人的几大元素：数字化生命、影像化记忆、人工智能、外星访客……刘宇昆的独特之处在于，他写的不是科幻探险或英雄奇幻，而是数据时代里每个人的生活和情感变化。透过这本书，我们看到的不仅是未来还有当下。\n\n\n\n\n
```

常用属性2：Tag['属性名']，通过属性名可以提取属性值。提取属性值时，需要定位到属性值对应的标签才能提取到，不能在父级中取子级的属性值。

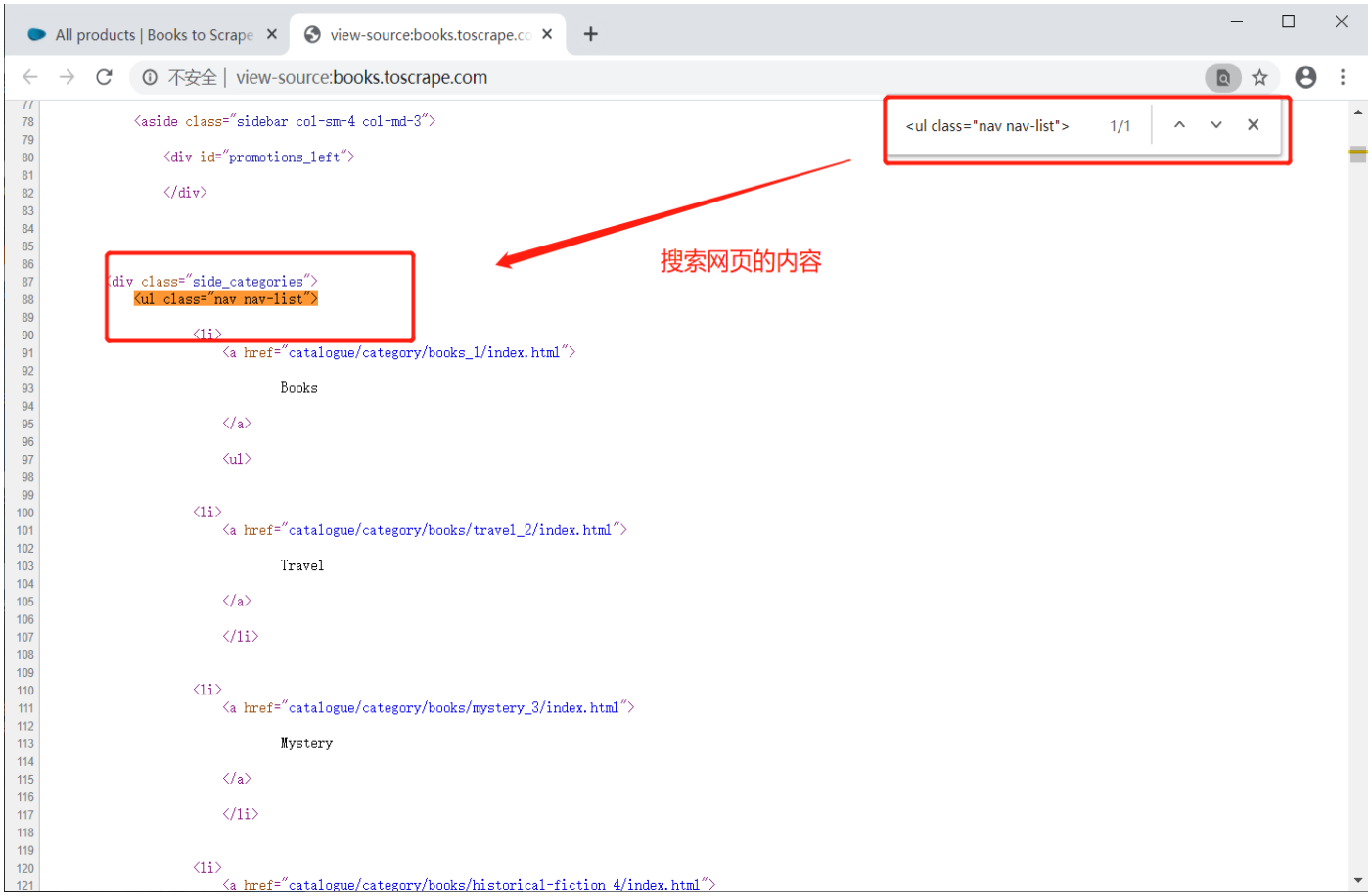
示例2：

```python
# 爬取《这个书苑不太冷》
import requests
from bs4 import BeautifulSoup
res = requests.get('https://localprod.pandateacher.com/python-manuscript/crawler-html/spider-men5.0.html')
soup = BeautifulSoup(res.text,'html.parser')

# 通过匹配标签和属性提取我们想要的数据
item = soup.find(class_='books')    # Tag对象

print(item['class'])    # 获取class的属性值，以列表返回
# 结果：['books']
print(item.find('img')['src'])  # 当获取的是连接的时候，以字符串返回
# 结果：./spider-men5.0_files/s29492583.jpg
```

三、知识加油站：标签、属性值查找方法

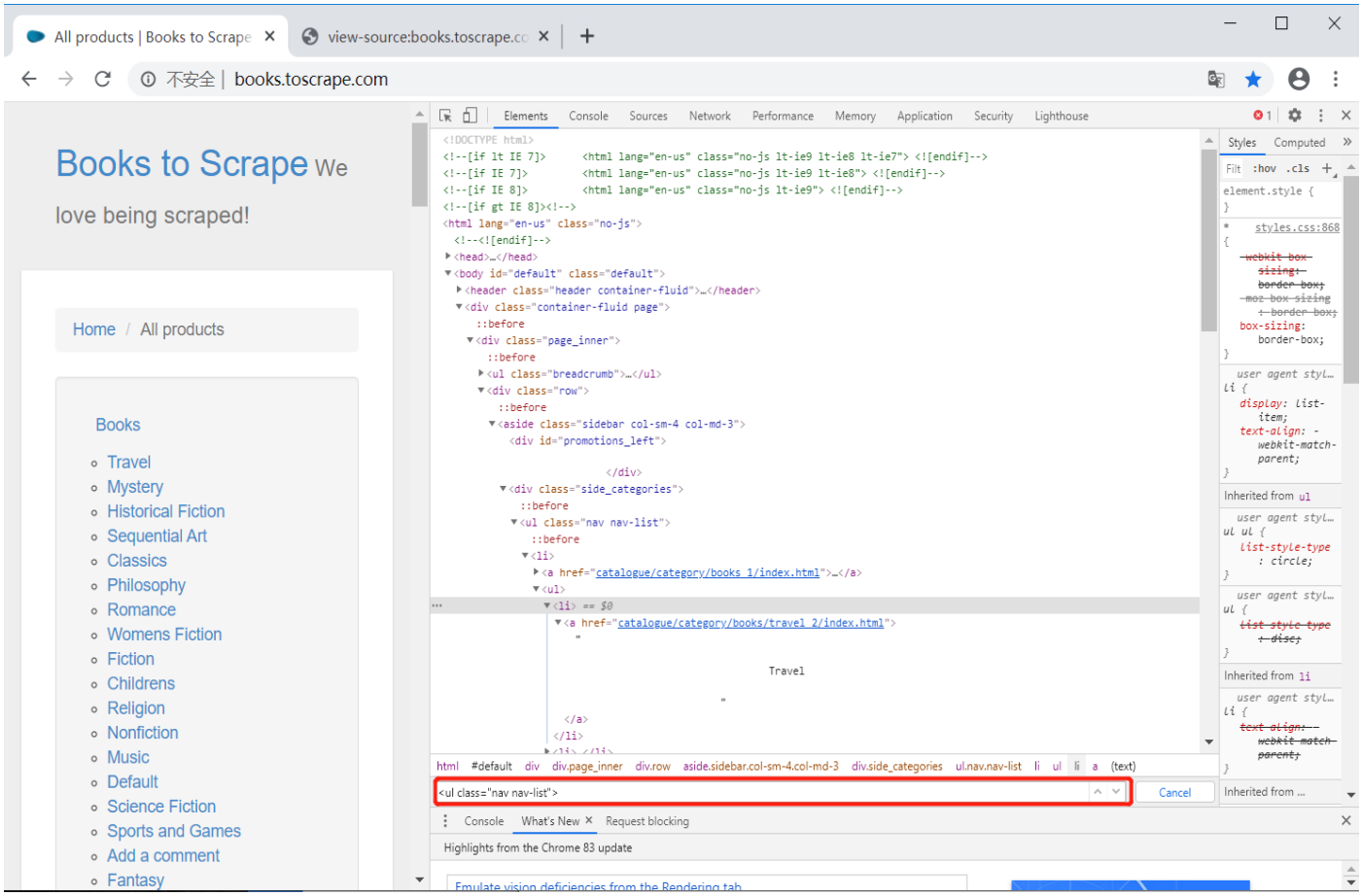
3.1 网页源代码搜索

打开网页的源代码可以使用【标签+属性+属性值】的形式来搜索。

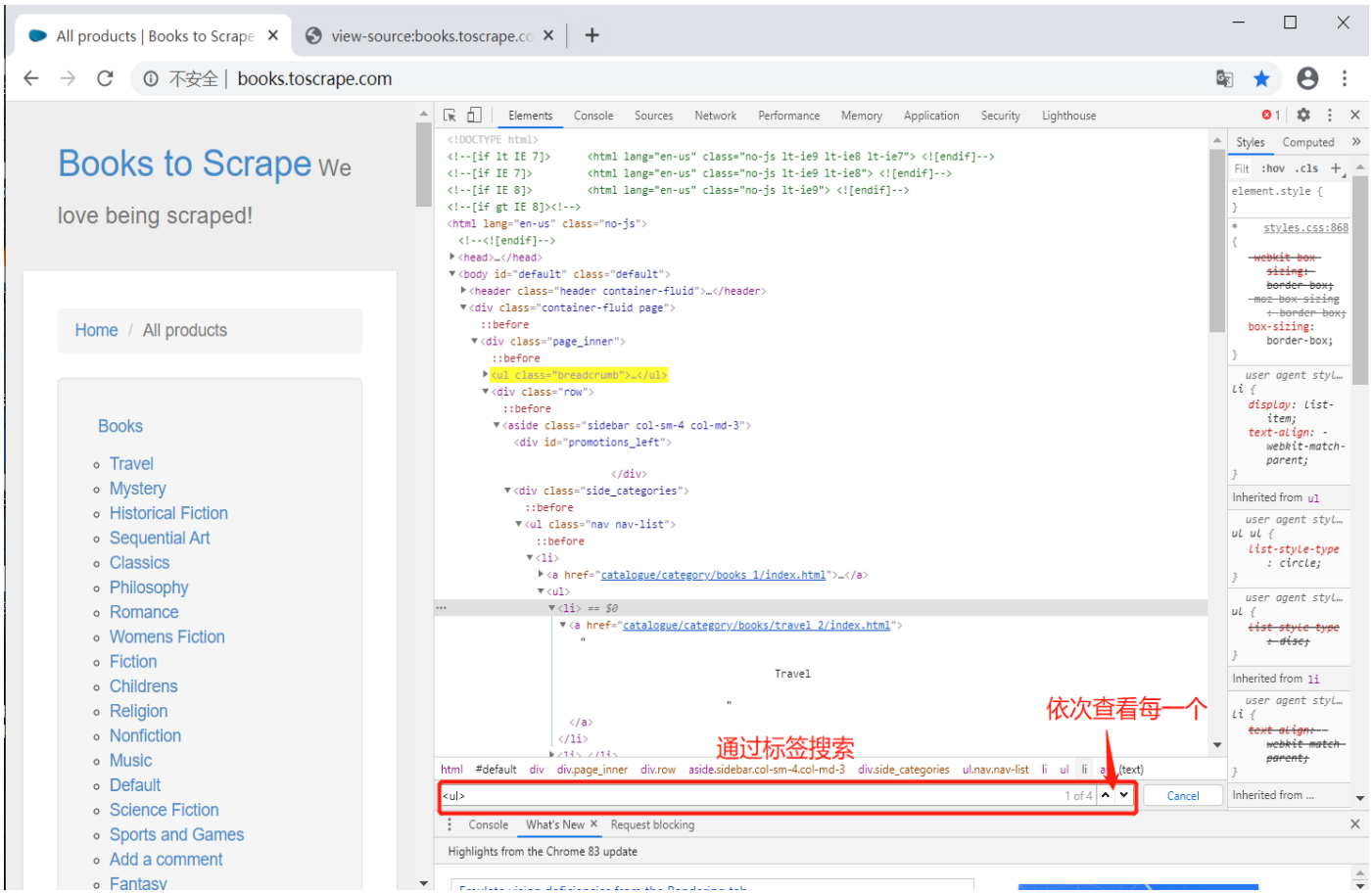


3.2 开发者工具代码搜索

在开发者工具中，**不可以**使用【标签+属性+属性值】的形式来搜索，只能单独搜索标签或者属性搜索。

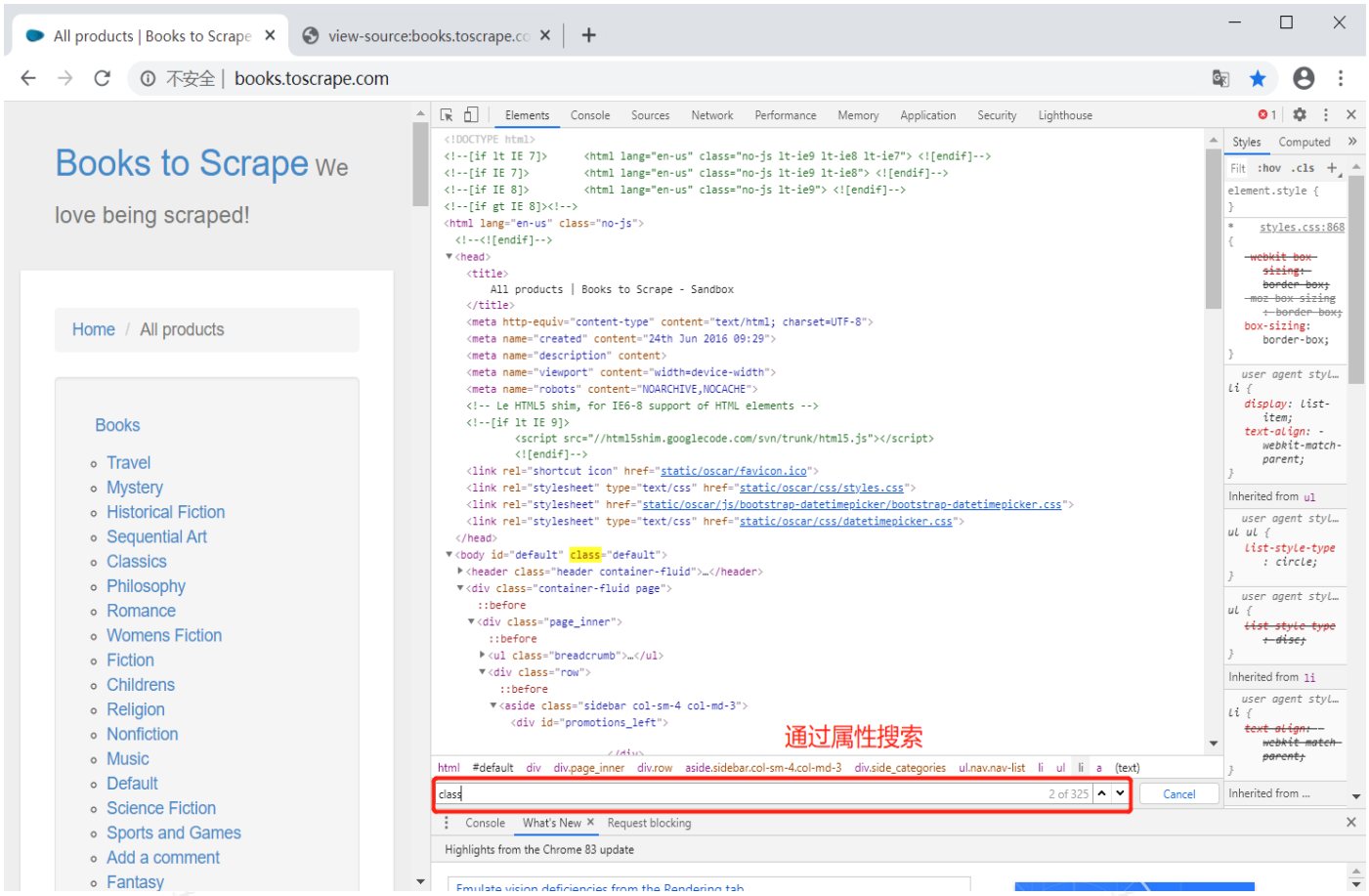

All products | Books to Scrape
view-source:books.toscrape.com
不安全 | books.toscrape.com
Books to Scrape We
love being scraped!
Home / All products
Books
Travel
Mystery
Historical Fiction
Sequential Art
Classics
Philosophy
Romance
Womens Fiction
Fiction
Childrens
Religion
Nonfiction
Music
Default
Science Fiction
Sports and Games
Add a comment
Fantasy
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Styles
Computed
<ul class="nav nav-list">
Cancel
Console
What's New
Request blocking
Highlights from the Chrome 83 update

- **通过【标签】搜索**

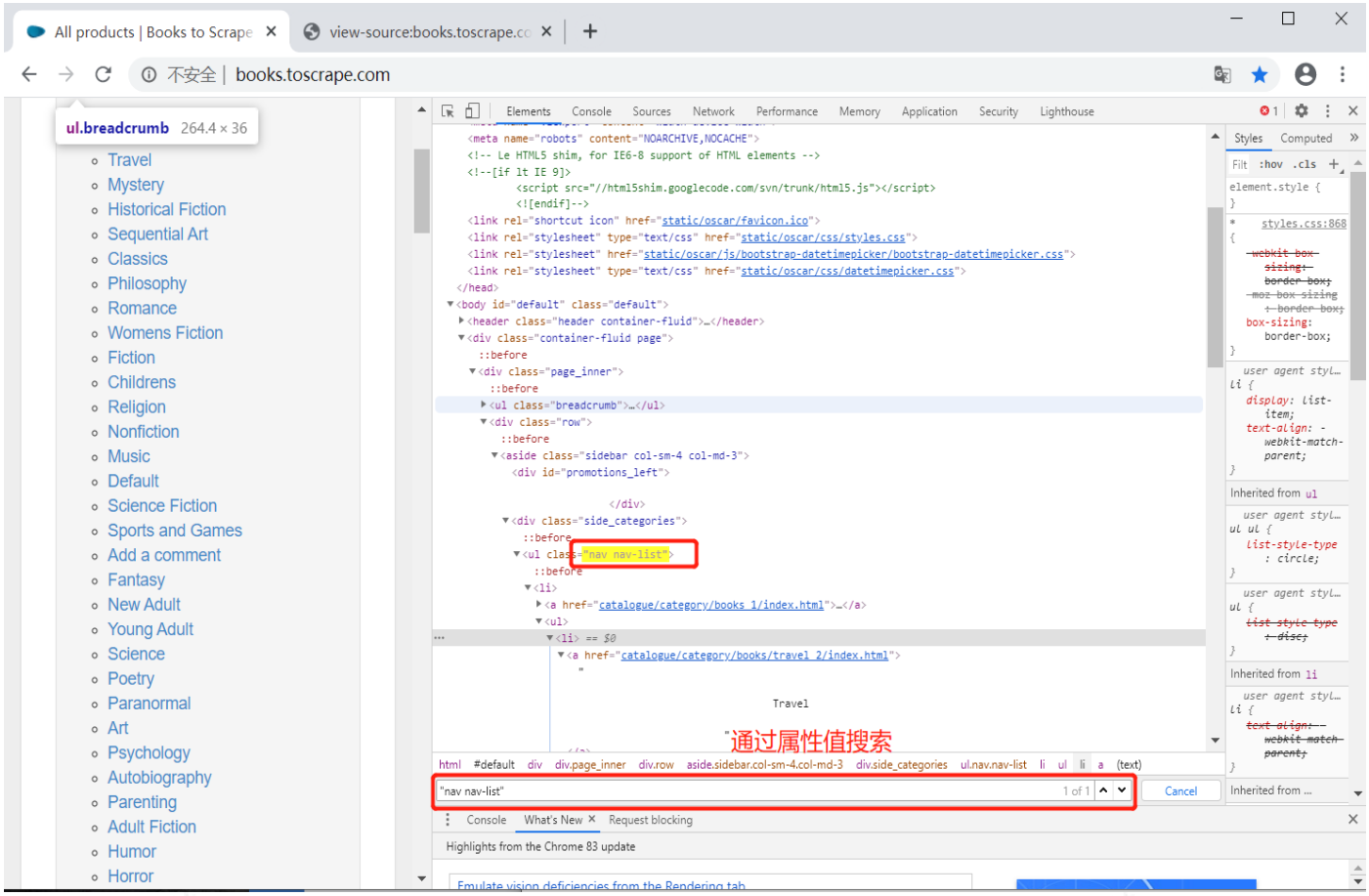

All products | Books to Scrape
view-source:books.toscrape.com
不安全 | books.toscrape.com
Books to Scrape We
love being scraped!
Home / All products
Books
Travel
Mystery
Historical Fiction
Sequential Art
Classics
Philosophy
Romance
Womens Fiction
Fiction
Childrens
Religion
Nonfiction
Music
Default
Science Fiction
Sports and Games
Add a comment
Fantasy
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Styles
Computed

1 of 4
Cancel
Console
What's New
Request blocking
Highlights from the Chrome 83 update
通过标签搜索
依次查看每一个

- **通过【属性】搜索**



- **通过【属性值】搜索**



属性一般可能会有很多个，所以一般不用属性来搜索，一般使用**标签和属性值**进行搜索。

常用查找技巧，提高搜索定位的准确性：

①搜索标签的时候，带上尖括号。

②搜索属性值的时，带上双引号。