Week-5 Assignment

Web Scraping

The aim for this asssignment is to scrap a website of book seller : [Books to Scrap]

From this website, we need to create a dataframe table with following columns:

- `title`
- `rating`
- `price`
- `link`

We need to use multiple libraries for this assignment which are: `requests` , `BeautifulSoup` and `pandas`

Import the Libraries

First we will import the libraries required to scrap the data from the website.
The libraries required for the operation:

- `requestd` : For handling HTTP Request
- `BeautifulSoup` : Used for parsing and extracting the elements
- `csv` : To handle manipulate the .csv files
- `pandas` : For Data Manipulation

In [1]:
```
import requests                                  # For HTTP request handling
from bs4 import BeautifulSoup as bs              # Used for HTML parsing
import csv                                       # In order to handle CSV file
import pandas as pd                              # Data manipulation
```

Get URL and send GET request

In order to get the data from the rquired website:

- We will use the `requests` library to send request to the website
- If request is accepted, then show the status

In [2]:
```
url = "http://books.toscrape.com/"
response = requests.get(url)        # Sending a request to the specified URL

if response.status_code == 200:     # Checking if the request was successful
    print("Request Successful")     # Printing a success message if the status code is 200
else:
    print("Request Failed")         # Printing a failure message if the status code is not 200
```

```
Request Successful
```

Parse the HTML Content

After successfully getting the data, we will first view the format of `HTML` text, till 1000 characters.

In [3]:
```
print(response.text[:1000])         # Printing the first 1000 characters of the response text
```

```
<!DOCTYPE html>
<!--[if lt IE 7]>      <html lang="en-us" class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html lang="en-us" class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html lang="en-us" class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html lang="en-us" class="no-js"> <!--<![endif]-->
    <head>
        <title>
    All products | Books to Scrape - Sandbox
</title>

        <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
        <meta name="created" content="24th Jun 2016 09:29" />
        <meta name="description" content="" />
        <meta name="viewport" content="width=device-width" />
        <meta name="robots" content="NOARCHIVE,NOCACHE" />

        <!-- Le HTML5 shim, for IE6-8 support of HTML elements -->
        <!--[if lt IE 9]>
        <script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
        <![endif]-->


            <link rel="shortcut icon" href="static/oscar/favicon.
```

After viewing, we will sparse the `HTML` file using `BeautifulSoup` library.

In [4]:
```
soup = bs(response.text, "html.parser")       # Creating a BeautifulSoup object for HTML parsing
print(type(soup))                             # Printing the type of the 'soup' object
```

```
<class 'bs4.BeautifulSoup'>
```

Extract Details for 1 Book

In order to get successful result, we will follow the following steps:

1. *Scrap the data of 1 Book*
2. *Scrap the data of all the books in 1 page*
3. *Scrap the data pf all the books of all 50 pages*

First we will find all the <article> tags in the website.
Then we will print and view the first content of <article> tag.

In [5]:
```
books = soup.find_all('article', class_='product_pod')      # Finding all HTML elements with the specified class
single_book = books[0]                                       # Accessing the first book element
single_book                                                  # Printing the details of the first book element
```

Out[5]:
```
<article class="product_pod">
<div class="image_container">
<a href="catalogue/a-light-in-the-attic_1000/index.html"><img alt="A Light in the Attic" class="thumbnail" src="medi
a/cache/2c/da/2cdad67c44b002e7ead0cc35693c0e8b.jpg"/></a>
</div>
<p class="star-rating Three">
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
<i class="icon-star"></i>
</p>
<h3><a href="catalogue/a-light-in-the-attic_1000/index.html" title="A Light in the Attic">A Light in the ...</a></h3>
<div class="product_price">
<p class="price_color">Â£51.77</p>
<p class="instock availability">
<i class="icon-ok"></i>

        In stock

</p>
<form>
<button class="btn btn-primary btn-block" data-loading-text="Adding..." type="submit">Add to basket</button>
</form>
</div>
</article>
```

Now we will extract the `title` attribute value from the first book element of <anchor> tag.

In [6]:
```
title = single_book.find('a', title=True)['title']  # Extracting the 'title' attribute value from the first book ele
ment
title
```

Out[6]: 'A Light in the Attic'

Now we will extract the `star-rating` class value from the first book element of <paragraph> tag.

In [7]:
```
rating = single_book.find('p', class_='star-rating')['class'][1]  # Extracting the rating class value from the first
book element
rating
```

Out[7]: 'Three'

Now we will extract and clean the `price_color` class value from the first book element of <paragraph> tag.

In [8]:
```
price = single_book.find('p', class_='price_color').text.strip().strip('Â')  # Extracting and cleaning the price of
the first book
price
```

Out[8]: '£51.77'

Now we will extract the `href` attribute value from the first book element of <anchor> tag.
After that, we will concatenate the initial `url` to `book_url`.

In [9]:
```
book_url = single_book.find('a')['href']          # Extracting the URL for the first book
link = url + book_url                             # Creating the complete URL for the book
link
```

Out[9]: 'http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html'

Extract Book Details for 1 Page

Using BeautifulSoup to find and extract book details from a single webpage:

1. Finds all `HTML` elements representing individual books.
2. Initializes an empty list to store book details.
3. For each book:
   - Extracting the book's title.
   - Extracting the book's rating.
   - Cleaning and extracting the book's price.
   - Extracting the book's URL and creating a complete link.
   - Appending all these details to a list.

In [10]:
```
books = soup.find_all('article', class_='product_pod')  # Finding all book elements
books_data = []                                          # List to store book details

for book in books:                                             # Iterating through each book element
    title = book.find('a', title=True)['title']                # Extracting the title of the book
    rating = book.find('p', class_='star-rating')['class'][1]  # Extracting the rating of the book
    price = book.find('p', class_='price_color').text.strip('Â')      # Extracting and cleaning the price
    book_url = book.find('a')['href']                          # Extracting the URL for the book
    link = url + book_url                                      # Creating the complete URL for the bo
ok

    books_data.append([title, rating, price, link])            # Appending book details to the list
```

Creating a DataFrame using `Pandas`.

In [11]:
```
page = pd.DataFrame(books_data, columns=["title","rating","price","link"])     # Creating a DataFrame from books_data
page
```

Out[11]:

	title	rating	price	link
0	A Light in the Attic	Three	£51.77	http://books.toscrape.com/catalogue/a-light-in...
1	Tipping the Velvet	One	£53.74	http://books.toscrape.com/catalogue/tipping-th...
2	Soumission	One	£50.10	http://books.toscrape.com/catalogue/soumission...
3	Sharp Objects	Four	£47.82	http://books.toscrape.com/catalogue/sharp-obje...
4	Sapiens: A Brief History of Humankind	Five	£54.23	http://books.toscrape.com/catalogue/sapiens-a-...
5	The Requiem Red	One	£22.65	http://books.toscrape.com/catalogue/the-requie...
6	The Dirty Little Secrets of Getting Your Dream...	Four	£33.34	http://books.toscrape.com/catalogue/the-dirty-...
7	The Coming Woman: A Novel Based on the Life of...	Three	£17.93	http://books.toscrape.com/catalogue/the-coming...
8	The Boys in the Boat: Nine Americans and Their...	Four	£22.60	http://books.toscrape.com/catalogue/the-boys-i...
9	The Black Maria	One	£52.15	http://books.toscrape.com/catalogue/the-black-...
10	Starving Hearts (Triangular Trade Trilogy, #1)	Two	£13.99	http://books.toscrape.com/catalogue/starving-h...
11	Shakespeare's Sonnets	Four	£20.66	http://books.toscrape.com/catalogue/shakespear...
12	Set Me Free	Five	£17.46	http://books.toscrape.com/catalogue/set-me-fre...
13	Scott Pilgrim's Precious Little Life (Scott Pi...	Five	£52.29	http://books.toscrape.com/catalogue/scott-pilg...
14	Rip it Up and Start Again	Five	£35.02	http://books.toscrape.com/catalogue/rip-it-up-...
15	Our Band Could Be Your Life: Scenes from the A...	Three	£57.25	http://books.toscrape.com/catalogue/our-band-c...
16	Olio	One	£23.88	http://books.toscrape.com/catalogue/olio_984/i...
17	Mesaerion: The Best Science Fiction Stories 18...	One	£37.59	http://books.toscrape.com/catalogue/mesaerion-...
18	Libertarianism for Beginners	Two	£51.33	http://books.toscrape.com/catalogue/libertaria...
19	It's Only the Himalayas	Two	£45.17	http://books.toscrape.com/catalogue/its-only-t...

Extract Book Details for All 50 Pages

Firstly, using a `for` loop to iterate over page numbers from `1 to 50 (inclusive)`.
Then constructing the `URL` for each page, using f-string formatting.
After that we will print and display each generated page `URL` during the loop execution.

In [12]:
```
for page_num in range(1, 51):                                                     # Looping through pages from 1 to 50
    page_url = f'http://books.toscrape.com/catalogue/page-{page_num}.html'  # Generating the URL for each page
    print(page_url)                                                         # Printing and viewing the generated pag
e URL
```

```
http://books.toscrape.com/catalogue/page-1.html
http://books.toscrape.com/catalogue/page-2.html
http://books.toscrape.com/catalogue/page-3.html
http://books.toscrape.com/catalogue/page-4.html
http://books.toscrape.com/catalogue/page-5.html
http://books.toscrape.com/catalogue/page-6.html
http://books.toscrape.com/catalogue/page-7.html
http://books.toscrape.com/catalogue/page-8.html
http://books.toscrape.com/catalogue/page-9.html
http://books.toscrape.com/catalogue/page-10.html
http://books.toscrape.com/catalogue/page-11.html
http://books.toscrape.com/catalogue/page-12.html
http://books.toscrape.com/catalogue/page-13.html
http://books.toscrape.com/catalogue/page-14.html
http://books.toscrape.com/catalogue/page-15.html
http://books.toscrape.com/catalogue/page-16.html
http://books.toscrape.com/catalogue/page-17.html
http://books.toscrape.com/catalogue/page-18.html
http://books.toscrape.com/catalogue/page-19.html
http://books.toscrape.com/catalogue/page-20.html
http://books.toscrape.com/catalogue/page-21.html
http://books.toscrape.com/catalogue/page-22.html
http://books.toscrape.com/catalogue/page-23.html
http://books.toscrape.com/catalogue/page-24.html
http://books.toscrape.com/catalogue/page-25.html
http://books.toscrape.com/catalogue/page-26.html
http://books.toscrape.com/catalogue/page-27.html
http://books.toscrape.com/catalogue/page-28.html
http://books.toscrape.com/catalogue/page-29.html
http://books.toscrape.com/catalogue/page-30.html
http://books.toscrape.com/catalogue/page-31.html
http://books.toscrape.com/catalogue/page-32.html
http://books.toscrape.com/catalogue/page-33.html
http://books.toscrape.com/catalogue/page-34.html
http://books.toscrape.com/catalogue/page-35.html
http://books.toscrape.com/catalogue/page-36.html
http://books.toscrape.com/catalogue/page-37.html
http://books.toscrape.com/catalogue/page-38.html
http://books.toscrape.com/catalogue/page-39.html
http://books.toscrape.com/catalogue/page-40.html
http://books.toscrape.com/catalogue/page-41.html
http://books.toscrape.com/catalogue/page-42.html
http://books.toscrape.com/catalogue/page-43.html
http://books.toscrape.com/catalogue/page-44.html
http://books.toscrape.com/catalogue/page-45.html
http://books.toscrape.com/catalogue/page-46.html
http://books.toscrape.com/catalogue/page-47.html
http://books.toscrape.com/catalogue/page-48.html
http://books.toscrape.com/catalogue/page-49.html
http://books.toscrape.com/catalogue/page-50.html
```

For collects book details from a website consisting of 50 Webpages, we will use two links:

- `primary_url` : A starting link used to build complete book URLs.
- `page_url` : A link to specify the directory of multiple webpages.

To extract data from 50 webpages, we will:

- First iterate through page numbers from 1 to 50.
- Construct the URL for each page on the website.
- Send a request to the page URL to get its content.
- Parsing the HTML content using BeautifulSoup.
- Find all elements representing individual books on the page.
- For each book, extracts its title, rating, price, and URL.
- Constructs the complete book URL by combining the primary URL with the book's specific URL.
- Gathers all these details into a list called `books_50_data`.

In [13]:
```
primary_url = "http://books.toscrape.com/"                                # Link to concatenate later
books_50_data = []                                                         # List to store book details from multip
le pages

for page_num in range(1, 51):                                              # Looping through page numbers from 1 to
50
    page_url = f'http://books.toscrape.com/catalogue/page-{page_num}.html'  # Generating the URL for each page
    response = requests.get(page_url)                                      # Sending a request to the page URL
    soup_page = bs(response.text, "html.parser")                           # Creating a BeautifulSoup object for HT
ML parsing
    books = soup_page.find_all('article', class_='product_pod')            # Finding all book elements on the page

    for book in books:                                                     # Iterating through each book element
        title = book.find('a', title=True)['title']                        # Extracting the title of the book
        rating = book.find('p', class_='star-rating')['class'][1]          # Extracting the rating of the book
        price = book.find('p', class_='price_color').text.strip().strip('Â')  # Extracting and cleaning the price
        book_url = book.find('a')['href']                                  # Extracting the URL for the book
        link = primary_url + book_url                                      # Creating the complete URL for the book

        books_50_data.append([title, rating, price, link])                 # Appending book details to the list
```

Creating a DataFrame using `Pandas`.

In [14]:
```
page_50 = pd.DataFrame(books_50_data, columns=["title","rating","price","link"])     # Creating a DataFrame from book
s_50_data
page_50
```

Out[14]:

	title	rating	price	link
0	A Light in the Attic	Three	£51.77	http://books.toscrape.com/a-light-in-the-attic...
1	Tipping the Velvet	One	£53.74	http://books.toscrape.com/tipping-the-velvet_9...
2	Soumission	One	£50.10	http://books.toscrape.com/soumission_998/index...
3	Sharp Objects	Four	£47.82	http://books.toscrape.com/sharp-objects_997/in...
4	Sapiens: A Brief History of Humankind	Five	£54.23	http://books.toscrape.com/sapiens-a-brief-hist...
...	...	...	...	...
995	Alice in Wonderland (Alice's Adventures in Won...	One	£55.53	http://books.toscrape.com/alice-in-wonderland-...
996	Ajin: Demi-Human, Volume 1 (Ajin: Demi-Human #1)	Four	£57.06	http://books.toscrape.com/ajin-demi-human-volu...
997	A Spy's Devotion (The Regency Spies of London #1)	Five	£16.97	http://books.toscrape.com/a-spys-devotion-the-...
998	1st to Die (Women's Murder Club #1)	One	£53.98	http://books.toscrape.com/1st-to-die-womens-mu...
999	1,000 Places to See Before You Die	Five	£26.08	http://books.toscrape.com/1000-places-to-see-b...

1000 rows × 4 columns

Export the Data

Saving the final data to `books_scraped.csv` file in local machine.

In [15]:
```
page_50.to_csv("books_scraped.csv", index = False)  # Saving the DataFrame to a CSV file without including the index
print("Data saved to .csv")                          # Printing a message confirming the data has been saved
```

```
Data saved to .csv
```