

## Week-5 Assignment

### Web Scrapping

By [Arijit Dhali](#) [Linkedin](#)

The aim for this assignment 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 [153]: 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 required website:

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

```
In [154]: 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 [155]: print(response.text[:1000]) # Printing the first 1000 characters of the response text

<!DOCTYPE html>
<!--[if lt IE 7]>
<!--[if IE 7]>
<!--[if IE 8]>
<!--[if gt IE 8]><!-->
<html lang="en-us" class="no-js lt-ie9 lt-ie8 lt-ie7"><![endif]-->
<html lang="en-us" class="no-js lt-ie9 lt-ie8"><![endif]-->
<html lang="en-us" class="no-js lt-ie9"><![endif]-->
<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="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
```

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

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

```
In [156]: 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 of 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 [157]: 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
print(single_book) # Printing the details of the first book element
```

```
Out[157]: <article class="product_pod">
<div class="image_container">
<a href="catalogue/a-light-in-the-attic_1000/index.html"></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">£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 [158]: title = single_book.find('a', title=True)['title'] # Extracting the 'title' attribute value from the first book element
print(title)
```

Out[158]: 'A Light in the Attic'

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

```
In [159]: rating = single_book.find('p', class_='star-rating')['class'][1] # Extracting the rating class value from the first book element
print(rating)
```

Out[159]: 'Three'

Now we will extract and clean the price\_color class value from the first book element of <paragraph> tag.

```
In [160]: price = single_book.find('p', class_='price_color').text.strip().strip('Â') # Extracting and cleaning the price of the first book
print(price)
```

Out[160]: '£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 [161]: 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
print(link)
```

Out[161]: '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 [162]: books = soup.find_all('article', class_='product_pod') # Finding all book elements
books_data = [] # List to store book details

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

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

Creating a DataFrame using Pandas .

```
In [163]: page = pd.DataFrame(books_data, columns=["title", "rating", "price", "link"]) # Creating a DataFrame from books_data
print(page)
```

Out[163]:

	title	rating	price	link
0	A Light in the Attic	Three	£51.77	http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html
1	Tipping the Velvet	One	£53.74	http://books.toscrape.com/catalogue/tipping-the-velvet_990/index.html
2	Soumission	One	£50.10	http://books.toscrape.com/catalogue/soumission_998/index.html
3	Sharp Objects	Four	£47.82	http://books.toscrape.com/catalogue/sharp-objects_997/index.html
4	Sapiens: A Brief History of Humankind	Five	£54.23	http://books.toscrape.com/catalogue/sapiens-a-brief-history-of-humankind_996/index.html
5	The Requiem Red	One	£22.65	http://books.toscrape.com/catalogue/the-requiem-red_995/index.html
6	The Dirty Little Secrets of Getting Your Dream...	Four	£33.34	http://books.toscrape.com/catalogue/the-dirty-little-secrets-of-getting-your-dream-true_994/index.html
7	The Coming Woman: A Novel Based on the Life...	Three	£17.93	http://books.toscrape.com/catalogue/the-coming-woman_993/index.html
8	The Boys in the Boat: Nine Americans and Their...	Four	£22.60	http://books.toscrape.com/catalogue/the-boys-in-the-boat_992/index.html
9	The Black Maria	One	£52.15	http://books.toscrape.com/catalogue/the-black-maria_991/index.html
10	Starving Hearts (Triangular Trade Trilogy, #1)	Two	£13.99	http://books.toscrape.com/catalogue/starving-hearts_990/index.html
11	Shakespeare's Sonnets	Four	£20.66	http://books.toscrape.com/catalogue/shakespeares-sonnets_989/index.html
12	Set Me Free	Five	£17.46	http://books.toscrape.com/catalogue/set-me-free_988/index.html
13	Scott Pilgrim's Precious Little Life (Scott Pilgrim vs. the World, #1)	Five	£52.29	http://books.toscrape.com/catalogue/scott-pilgrims-precious-little-life_987/index.html
14	Rip it Up and Start Again	Five	£35.02	http://books.toscrape.com/catalogue/rip-it-up-and-start-again_986/index.html
15	Our Band Could Be Your Life: Scenes from the ...	Three	£57.25	http://books.toscrape.com/catalogue/our-band-could-be-your-life_985/index.html
16	Olio	One	£23.88	http://books.toscrape.com/catalogue/olio_984/index.html
17	Mesaerion: The Best Science Fiction Stories 18...	One	£37.59	http://books.toscrape.com/catalogue/mesaerion-the-best-science-fiction-stories-1890s_983/index.html
18	Libertarianism for Beginners	Two	£51.33	http://books.toscrape.com/catalogue/libertarianism-for-beginners_982/index.html
19	It's Only the Himalayas	Two	£45.17	http://books.toscrape.com/catalogue/its-only-the-himalayas_981/index.html

### 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 [164]: for page_num in range(1, 51):
    page_url = f'http://books.toscrape.com/catalogue/page-{page_num}.html' # Looping through pages from 1 to 50
    print(page_url) # Generating the URL for each page
# Printing and viewing the generated page 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 [165]: primary_url = "http://books.toscrape.com/" # Link to concatenate later
books_50_data = [] # List to store book details from multiple 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 HTML 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 [166]: page_50 = pd.DataFrame(books_50_data, columns=["title", "rating", "price", "link"]) # Creating a DataFrame from books_50_data
print(page_50)
```

Out[166]:

	title	rating	price	link
0	A Light in the Attic	Three	£51.77	http://books.toscrape.com/a-light-in-the-attic_1000/index.html
1	Tipping the Velvet	One	£53.74	http://books.toscrape.com/tipping-the-velvet_990/index.html
2	Soumission	One	£50.10	http://books.toscrape.com/soumission_998/index.html
3	Sharp Objects	Four	£47.82	http://books.toscrape.com/sharp-objects_997/index.html
4	Sapiens: A Brief History of Humankind	Five	£54.23	http://books.toscrape.com/sapiens-a-brief-history-of-humankind_996/index.html
...	...	...	...	...
995	Alice in Wonderland (Alice's Adventures in Wonderland, #1)	One	£55.53	http://books.toscrape.com/alice-in-wonderland-alice-in-wonderland_995/index.html
996	Ajin: Demi-Human, Volume 1 (Ajin: Demi-Human #1)	Four	£57.06	http://books.toscrape.com/ajin-demi-human-volume-1_996/index.html
997	A Spy's Devotion (The Regency Spies of London #1)	Five	£16.97	http://books.toscrape.com/a-spys-devotion-the-regency-spies-of-london_997/index.html
998	1st to Die (Women's Murder Club #1)	One	£53.98	http://books.toscrape.com/1st-to-die-womens-murder-club_998/index.html
999	1000 Places to See Before You Die	Five	£26.08	http://books.toscrape.com/1000-places-to-see-before-you-die_999/index.html

1000 rows × 4 columns

### Export the Data

Saving the final data to books\_scraped.csv file in local machine.

```
In [167]: 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
```