```
# Import necessary libraries
import requests
from bs4 import BeautifulSoup as bs
import pandas as pd
import csv


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
```

ChatGPT

```
print(response.text[:1000])
```

```
<!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.
```

```
soup = bs(response.text, "html.parser")     # Creating a BeautifulSoup object for HTML parsing
print(type(soup))
```

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

```
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
```

```
<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="media/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>
```

```
title = single_book.find('a', title=True)['title']  # Extracting the 'title' attribute value from the first book element
title
```

```
'A Light in the Attic'
```

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

```
'Three'
```

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

```
'£51.77'
```

```
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
```

```
'http://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html'
```

```
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().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 book

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


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

	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...

Next steps: Generate code with page View recommended plots

```
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 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

```
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])


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

	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-...