K. J. Somaiya College of Engineering, Mumbai-77
(A Constituent College of Somaiya Vidyavihar University)
Department of Computer Engineering

Batch: H2-1	**Roll No:-** 16010122151
Experiment No. 9	

Title : To implement data scraping using Selenium/ ScraPy

Aim: To perform testing using web scrapers in Python

Expected Outcome of Experiment:
CO2: Application of Exploratory data analysis (EDA) on Real world problems.

Books/ Journals/ Websites referred:
1. https://realpython.com/python-web-scraping-practical-introduction/

Web Scrapping:

Web scraping is the process of collecting and parsing raw data from the Web.

Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured.

Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code.

Why is Web Scraping Used?

• Price Comparison: Services such as ParseHub use web scraping to collect data from online shopping websites and use it to compare the prices of products.

• Email address gathering: Many companies that use email as a medium for marketing, use web scraping to collect email ID and then send bulk emails.

• Social Media Scraping: Web scraping is used to collect data from Social Media websites such as Twitter to find out what's trending.

• Research and Development: Web scraping is used to collect a large set of data (Statistics, General Information, Temperature, etc.) from websites, which are analyzed and used to carry out Surveys or for R&D.

• Job listings: Details regarding job openings, interviews are collected from different websites and then listed in one place so that it is easily accessible to the user.

Why is Python Good for Web Scraping?

• Here is the list of features of Python which makes it more suitable for web scraping.

• Ease of Use: Python is simple to code. You do not have to add semi-colons ";" or curly-braces "{}" anywhere. This makes it less messy and easy to use.

• Large Collection of Libraries: Python has a huge collection of libraries such as Numpy, Matlplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.

• Dynamically typed: In Python, you don't have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.

How Do You Scrape Data From A Website?

• When you run the code for web scraping, a request is sent to the URL that you have mentioned. As a response to the request, the server sends the data and allows you to read the HTML or XML page. The code then, parses the HTML or XML page, finds the data and extracts it.

• To extract data using web scraping with python, you need to follow these basic steps:

- Find the URL that you want to scrape
- Inspecting the Page
- Find the data you want to extract
- Write the code
- Run the code and extract the data
- Store the data in the required format

Libraries used for Web Scraping

- **Selenium:** Selenium is a web testing library. It is used to automate browser activities.
- **BeautifulSoup:** Beautiful Soup is a Python package for parsing HTML and XML documents. It creates parse trees that is helpful to extract the data easily.
- **Pandas:** Pandas is a library used for data manipulation and analysis. It is used to extract the data and store it in the desired format.

Implementation:

```
import bs4
from bs4 import BeautifulSoup as bs import requests

link='https://www.flipkart.com/search?q=tv&as=on&as-
show=on&otracker=AS_Query_TrendingAutoSuggest_8_0_na_na_na&otracker
1=AS_Query_TrendingAutoSuggest_8_0_na_na_na&as-pos=8&as-
type=TRENDING&suggestionId=tv&requestId=9c9fa553-b7e5-454b-a65b-
bbb7a9c74a29'

page=requests.get(link)
page.content

soup=bs(page.content, 'html.parser')
# print(soup.prettify())

#extract name of product
name=soup.find('div',class_="_4rR01T")
print(name.text)

#extract rating of product
rating=soup.find('div',class_="_3LWZlK")
print(rating.text)

#extracting other details
details=soup.find('div',class_="fMghEO")
print(details.text)

#to get all details seperately
for each in details:
    spec=each.find_all('li',class_='rgWa7D')
    print(spec[0].text)
    print(spec[1].text)
    print(spec[2].text)
    print(spec[4].text)
    print(spec[5].text)
    print(spec[7].text)

#extract price of the product
```

```
price=soup.find('div',class_='_30jeq3 _1_WHN1')
print(price.text)
```

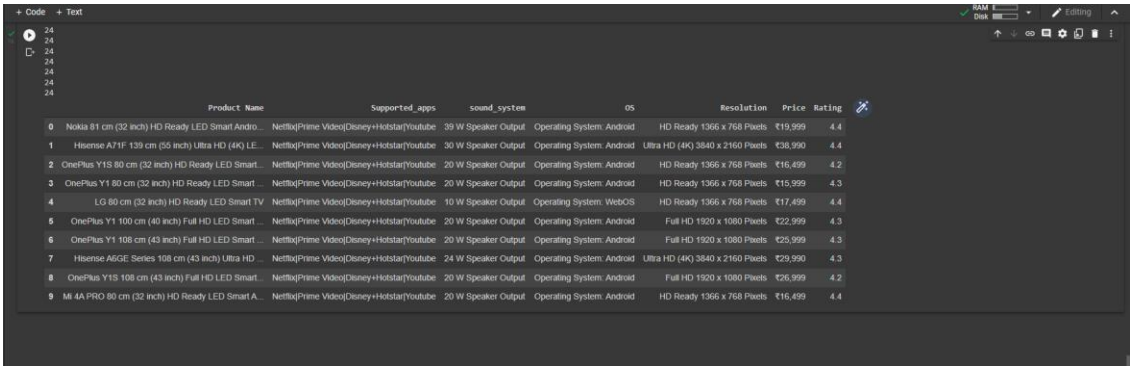
```
TOSHIBA E35KP 80 cm (32 inch) HD Ready LED Smart Android TV with DTS Virtual X (2022 Model)
4.2
Netflix|Prime Video|Disney+Hotstar|YoutubeOperating System: AndroidHD Ready 1366 x 768 Pixels16 W Speaker Output60 Hz Refresh Rate2 x HDMI | 2 x USBA+ Grade LED panel1 Year Warranty on Product From Toshiba
Netflix|Prime Video|Disney+Hotstar|Youtube
Operating System: Android
HD Ready 1366 x 768 Pixels
60 Hz Refresh Rate
2 x HDMI | 2 x USB
1 Year Warranty on Product From Toshiba
₹14,990
```

```
import bs4
from bs4 import BeautifulSoup as bs
import requests
import pandas as pd
link='https://www.flipkart.com/search?q=tv&as=on&as-
show=on&otracker=AS_Query_TrendingAutoSuggest_8_0_na_na_na&otracker
1=AS_Query_TrendingAutoSuggest_8_0_na_na_na&as-pos=8&as-
type=TRENDING&suggestionId=tv&requestId=9c9fa553-b7e5-454b-a65b-
bbb7a9c74a29'
page=requests.get(link)
page.content
soup=bs(page.content, 'html.parser')
# print(soup.prettify())
products=[] #List to store the name of the product
prices=[] #List to store price of the product
ratings=[] #List to store rating of the product
apps = [] #List to store supported apps

os = [] #List to store operating system
hd = [] #List to store resolution
sound = [] #List to store sound output
for data in soup.findAll('div',class_='_3pLy-c row'):
 names=data.find('div', attrs={'class':'_4rR01T'})
 price=data.find('div', attrs={'class':'_30jeq3 _1_WHN1'})
 rating=data.find('div', attrs={'class':'_3LWZlK'})
 specification = data.find('div', attrs={'class':'fMghEO'})

 for each in specification:
 col=each.find_all('li', attrs={'class':'rgWa7D'})
 app =col[0].text
 os_ = col[1].text
 hd_ = col[2].text
 sound_ = col[3].text
 products.append(names.text) # Add product name to list
```

```
 prices.append(price.text) # Add price to list
 apps.append(app)# Add supported apps specifications to list
 os.append(os_) # Add operating system specifications to list
 hd.append(hd_) # Add resolution specifications to list
 sound.append(sound_) # Add sound specifications to list
 ratings.append(rating.text) #Add rating specifications tolist
#printing the length of list
print(len(products))
print(len(ratings))
print(len(prices))
print(len(apps))
print(len(sound))
print(len(os))
print(len(hd))
df=pd.DataFrame({'Product
Name':products,'Supported_apps':apps,'sound_system':sound,'OS':os,"Resolu
tion":hd,'Price':prices,'Rating':ratings})
df.head(10)
```



	Product Name	Supported_apps	sound_system	OS	Resolution	Price	Rating
0	Nokia 81 cm (32 inch) HD Ready LED Smart Andro...	Netflix|Prime Video|Disney+Hotstar|Youtube	39 W Speaker Output	Operating System: Android	HD Ready 1366 x 768 Pixels	₹19,999	4.4
1	Hisense A71F 139 cm (55 inch) Ultra HD (4K) LE...	Netflix|Prime Video|Disney+Hotstar|Youtube	30 W Speaker Output	Operating System: Android	Ultra HD (4K) 3840 x 2160 Pixels	₹38,990	4.4
2	OnePlus Y1S 80 cm (32 inch) HD Ready LED Smart...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	HD Ready 1366 x 768 Pixels	₹16,499	4.2
3	OnePlus Y1 80 cm (32 inch) HD Ready LED Smart ...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	HD Ready 1366 x 768 Pixels	₹15,999	4.3
4	LG 80 cm (32 inch) HD Ready LED Smart TV	Netflix|Prime Video|Disney+Hotstar|Youtube	10 W Speaker Output	Operating System: WebOS	HD Ready 1366 x 768 Pixels	₹17,499	4.4
5	OnePlus Y1 100 cm (40 inch) Full HD LED Smart ...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	Full HD 1920 x 1080 Pixels	₹22,999	4.3
6	OnePlus Y1 108 cm (43 inch) Full HD LED Smart ...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	Full HD 1920 x 1080 Pixels	₹25,999	4.3
7	Hisense A6GE Series 108 cm (43 inch) Ultra HD ...	Netflix|Prime Video|Disney+Hotstar|Youtube	24 W Speaker Output	Operating System: Android	Ultra HD (4K) 3840 x 2160 Pixels	₹29,990	4.3
8	OnePlus Y1S 108 cm (43 inch) Full HD LED Smart...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	Full HD 1920 x 1080 Pixels	₹26,999	4.2
9	Mi 4A PRO 80 cm (32 inch) HD Ready LED Smart A...	Netflix|Prime Video|Disney+Hotstar|Youtube	20 W Speaker Output	Operating System: Android	HD Ready 1366 x 768 Pixels	₹16,499	4.4

Conclusion:

That's how we can scrape and store the data from the website. Here we have learned to scrape just one page, we can also perform the same on various pages and extract more data for comparison or analysis. The next step from here is to clean the data and perform analysis.

Post lab Questions:

1. What are the different HTTP response status codes?

HTTP response status codes are standardized codes that indicate the outcome of an HTTP request. They are sent by a server in response to a client's request. Here are some of the most common HTTP response status codes:

- 1xx Informational:
  - 100 Continue
  - 101 Switching Protocols
  - 102 Processing (WebDAV; RFC 2518)
- 2xx Success:
  - 200 OK
  - 201 Created
  - 202 Accepted
  - 204 No Content
  - 206 Partial Content
- 3xx Redirection:
  - 300 Multiple Choices
  - 301 Moved Permanently
  - 302 Found (previously "Moved Temporarily")
  - 304 Not Modified
  - 307 Temporary Redirect
  - 308 Permanent Redirect
- 4xx Client Error:
  - 400 Bad Request
  - 401 Unauthorized
  - 403 Forbidden
  - 404 Not Found
  - 405 Method Not Allowed
  - 409 Conflict
  - 410 Gone
  - 429 Too Many Requests
- 5xx Server Error:
  - 500 Internal Server Error
  - 501 Not Implemented

- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout
- 505 HTTP Version Not Supported

S.N.	Code and Description
1	**1xx: Informational** It means the request has been received and the process is continuing.
2	**2xx: Success** It means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** It means further action must be taken in order to complete the request.
4	**4xx: Client Error** It means the request contains incorrect syntax or cannot be fulfilled.
5	**5xx: Server Error** It means the server failed to fulfill an apparently valid request.

1xx: Information

Message	Description
100 Continue	Only a part of the request has been received by the server, but as long as it has not been rejected, the client should continue with the request.
101 Switching Protocols	The server switches protocol.

2xx: Successful

Message	Description
200 OK	The request is OK.
201 Created	The request is complete, and a new resource is created .
202 Accepted	The request is accepted for processing, but the processing is not complete.
203 Non-authoritative Information	The information in the entity header is from a local or third-party copy, not from the original server.
204 No Content	A status code and a header are given in the response, but there is no entity-body in the reply.
205 Reset Content	The browser should clear the form used for this transaction for additional input.
206 Partial Content	The server is returning partial data of the size requested. Used in response to a request specifying a *Range* header. The server must specify the range included in the response with the *Content-Range* header.

3xx: Redirection

Message	Description
300 Multiple Choices	A link list. The user can select a link and go to that location. Maximum five addresses .
301 Moved Permanently	The requested page has moved to a new url .
302 Found	The requested page has moved temporarily to a new url .
303 See Other	The requested page can be found under a different url .
304 Not Modified	This is the response code to an *If-Modified-Since* or *If-None-Match* header, where the URL has not been modified since the specified date.
305 Use Proxy	The requested URL must be accessed through the proxy mentioned in the *Location* header.
306 *Unused*	This code was used in a previous version. It is no longer used, but the code is reserved.
307 Temporary Redirect	The requested page has moved temporarily to a new url.

4xx: Client Error

Message	Description
400 Bad Request	The server did not understand the request.
401 Unauthorized	The requested page needs a username and a password.
402 Payment Required	*You can not use this code yet.*
403 Forbidden	Access is forbidden to the requested page.
404 Not Found	The server can not find the requested page.
405 Method Not Allowed	The method specified in the request is not allowed.
406 Not Acceptable	The server can only generate a response that is not accepted by the client.
407 Proxy Authentication Required	You must authenticate with a proxy server before this request can be served.
408 Request Timeout	The request took longer than the server was prepared to wait.
409 Conflict	The request could not be completed because of a conflict.
410 Gone	The requested page is no longer available .
411 Length Required	The "Content-Length" is not defined. The server will not accept the request without it .
412 Precondition Failed	The pre condition given in the request evaluated to false by the server.
413 Request Entity Too Large	The server will not accept the request, because the request entity is too large.
414 Request-url Too Long	The server will not accept the request, because the url is too long. Occurs when you convert a "post" request to a "get" request with a long query information .
415 Unsupported Media Type	The server will not accept the request, because the mediatype is not supported .
416 Requested Range Not Satisfiable	The requested byte range is not available and is out of bounds.
417 Expectation Failed	The expectation given in an Expect request-header field could not be met by this server.

5xx: Server Error

Message	Description
500 Internal Server Error	The request was not completed. The server met an unexpected condition.
501 Not Implemented	The request was not completed. The server did not support the functionality required.
502 Bad Gateway	The request was not completed. The server received an invalid response from the upstream server.
503 Service Unavailable	The request was not completed. The server is temporarily overloading or down.
504 Gateway Timeout	The gateway has timed out.
505 HTTP Version Not Supported	The server does not support the "http protocol" version.

2. How to get the Updated Daily News using Python

```
import requests
from bs4 import BeautifulSoup

url = 'https://www.bbc.com/news' response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser') headlines =
soup.find('body').find_all('h3')
unwanted = ['BBC World News TV', 'BBC World Service Radio', 'News
daily newsletter', 'Mobile app', 'Get in touch']

for x in list(dict.fromkeys(headlines)): if x.text.strip() not in
unwanted:
print(x.text.strip())
```