International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Data Extraction Using Web Scraping

Andrew Sharan R[1], Manoj C[2]

[1]UG Student, Computer Science, Sri Krishna Arts and Science College, Coimbatore, India
[2]UG Student, Computer Science, Sri Krishna Arts and Science College, Coimbatore, India
[1]andrewsharan67@gmail.com, [2]manojmj278@gmail.com

ABSTRACT

The internet is a huge collection of different kinds of data. It is the origin of extremely big information and data source from where one can get information about anything that present in the entire globe. Every information on the internet is in poorly structured form. It takes lot of time to search the useful data because not every information is useful and it becomes a challenge for the users how to get the useful data in a short period of time. One of the solutions is Web Scraping. With the help of Web Scraping, we can convert the unstructured data into a structured form. In this article, you will learn web scraping on brief and how to extract data from a website.

Keywords: Python, extract, data, website, web scraping

1.Introduction –

Web Scraping is an automated method used to extract data from websites. The data present in the websites are unstructured. It helps to collect these unstructured data and store it in a structured form. It is also called as web harvesting or web data extraction. There are several ways to scrap data from the websites. We'll be seeing how to work web scraping using Python.

edureka!
XML
CSV
Webpages
Web Scraping
Structured Data

Figure 1: Web Scraping

2.Uses of Web Scraping:

2.1 Price Comparison:

Web Scraping is used to collect data from online shopping websites and used to compare the prices of products.

2.2 Email address gathering:

Many companies that use email as medium for marketing use web scraping to gather email IDs and then send bulk emails.

2.3 Social Media Scraping:

It is used to collect data from social media websites like Twitter, Facebook, etc.

2.4 Research and Development:

It is used to collect large set of data from websites in the fields of Research and Development.

2.5Job listings:

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

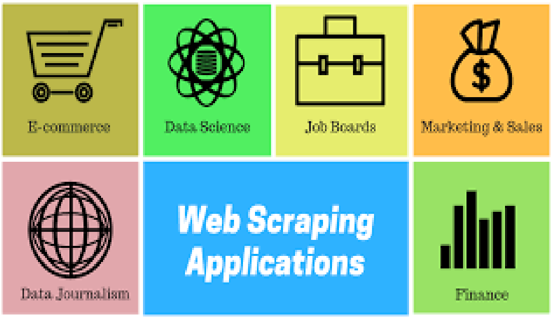


Fig 2: Applications of Web Scraping

3.Reasons why Python is good for Web Scraping:

- Python is ease of use because we do not have any semicolons or curly braces anywhere.
- Python has a large collection of libraries such as NumPy, Pandas, Matplotlib which are suitable for web scraping and for further manipulation of the extracted data.
- We can directly use the variables wherever required. There is no need to define the data types for variables.
- The syntax for Python is easily understandable for the users because the statements in Python are similar to normal English statements which looks very expressive.
- We can write small codes to perform huge tasks in order to use our time efficiently.
- Python community has one of the biggest and most active communities.

4.Tools used in Web Scraping:

There are various tools and techniques used in web scraping. Some of the tools are:

- ParseHub
- Scrapy
- Octoparse
- Scraper API
- Mozenda
- Webhose.io
- Content Grabber
- Common Crawl

5.Main tools used in Web Scraping:

5.1Requests:
It is a Python library used for making various types of HTTP requests like GET, POST, etc.
5.2 Beautiful Soup:
It is the most widely used Python library used for creating parse tree for parsing HTML and XML documents.
5.3 Selenium:
It is a Python library used for automated testing for web applications.
5.4 Pandas:
It is an open-source library used for working with relational or labeled data.

6.Steps to be followed whilescrapping data from a website

To extract data using web scraping with Python, we need to follow these basic steps:
Step 1: Find the website URL you want to scrap.
Step 2: Analyze the page.
Step 3: Find the data which you want to extract.
Step 4: Write the program.
Step 5: Run the program and extract the data.

Step 6: Store the data in the needed format.

6.1 Example:

We are going to scrap data from the leading online ecommerce site Flipkartto extract the name, price and rate of some laptops.

To scrap the data, we need some prerequisites:

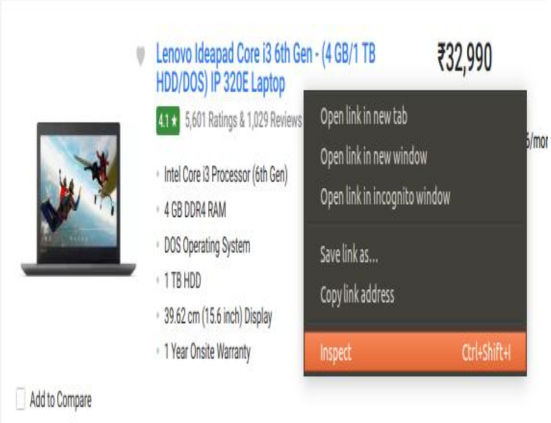
- Python 2.x or Python 3.x. Libraries such as Selenium, BeautifulSoup, Pandas should be installed.
- Google Chrome Browser and
- Ubuntu Operating System

Step 1: Find the website URL that you want to scrap

The website URL of the page is https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2

Step 2: Analyze the page

To analyze the page,right click on the element and click “Inspect”



Lenovo Ideapad Core i3 6th Gen - (4 GB/1 TB HDD/DOS) IP 320E Laptop

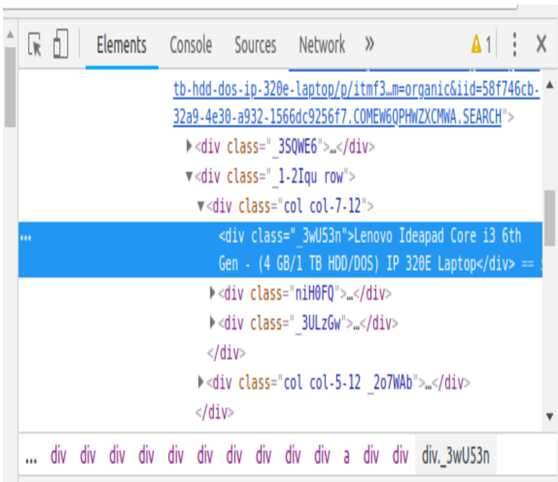
₹32,990

4.1★ 5,601 Ratings & 1,029 Reviews

- Intel Core i3 Processor (6th Gen)
- 4 GB DDR4 RAM
- DOS Operating System
- 1 TB HDD
- 39.62 cm (15.6 inch) Display
- 1 Year Onsite Warranty

Figure 3: Inspecting the page

When you click the “Inspect” tab, a “Browser Inspector Box” will be seen.



```
tb-hdd-dos-ip-320e-laptop/p/itmf3…m=organic&iid=58f746cb-
32a9-4e30-a932-1566dc9256f7.COMEW6QPHWZXCMWA.SEARCH">
  ▶<div class="_3SQWE6">…</div>
  ▼<div class="_1-2Iqu row">
    ▼<div class="col col-7-12">
        <div class="_3wU53n">Lenovo Ideapad Core i3 6th
        Gen - (4 GB/1 TB HDD/DOS) IP 320E Laptop</div> ==
      ▶<div class="niH0FQ">…</div>
      ▶<div class="_3ULzGw">…</div>
     </div>
    ▶<div class="col col-5-12 _2o7WAb">…</div>
   </div>
… div div div div div div div div div div a div div div._3wU53n
```

Fig 4: Browser Inspector Box

Step 3: Find the data which you want to extract

Extract price, name and rate which is in the “div” tag.

Step 4: Write the program

Create a Python file by openingthe terminal in Ubuntu and type gedit<file name> with .py extension. The file name is “web-s”

gedit web-s.py

Before writing the code, we should import all the necessary libraries:

from selenium import webdriver
from BeautifulSoup import BeautifulSoup

```
import pandas as pd
```

To installwebdriver to use Chrome browser, we have to set the path to chromedriver

```
driver = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver")
```

The code to open the URL is as follows:

```
products= []
prices= []
ratings= []
driver.get("<ahref="https://www.flipkart.com/laptops/">https://www.flipkart.com/laptops/</a>~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&amp;amp;amp;amp;amp;amp;amp;amp;amp;uniq")
```

After writing the code, we should extract the data from the website.The data we want to extract is nested in <div> tags. So, we should find the div tags with those respective class-names, extract the data and store the data in a variable.

```
content = driver.page_source
soup = BeautifulSoup(content)
for a in soup.findAll('a',href=True, attrs={'class':'_31qSD5'}):
name=a.find('div', attrs={'class':'_3wU53n'})
price=a.find('div', attrs={'class':'_1vC4OE _2rQ-NK'})
rating=a.find('div', attrs={'class':'hGSR34 _2beYZw'})
products.append(name.text)
prices.append(price.text)
ratings.append(rating.text)
```

Step 5: Run the program and extract the data

To run the code, use the command*python web-s.py*

Step 6: Store the data in a needed format

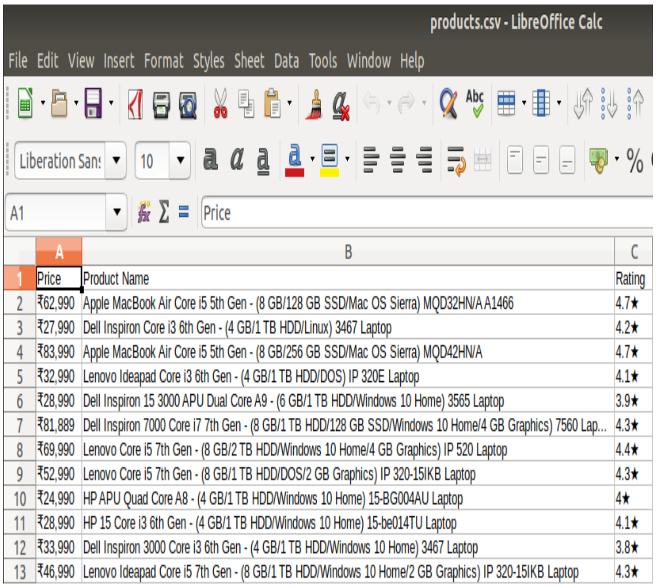
After extracting the data, we want to store it in a format. We can store the extracted data in a CSV (Comma Separated Value) format.

```
df = pd.DataFrame({'Product Name':products,'Price':prices,'Rating':ratings})

df.to_csv('products.csv', index=False, encoding='utf-8')
```

Run the whole code again.

A file named “products.csv”is created and this file contains the extracted data in an CSV format.


products.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10
A1
Price

	A	B	C
1	Price	Product Name	Rating
2	₹62,990	Apple MacBook Air Core i5 5th Gen - (8 GB/128 GB SSD/Mac OS Sierra) MQD32HN/A A1466	4.7★
3	₹27,990	Dell Inspiron Core i3 6th Gen - (4 GB/1 TB HDD/Linux) 3467 Laptop	4.2★
4	₹83,990	Apple MacBook Air Core i5 5th Gen - (8 GB/256 GB SSD/Mac OS Sierra) MQD42HN/A	4.7★
5	₹32,990	Lenovo Ideapad Core i3 6th Gen - (4 GB/1 TB HDD/DOS) IP 320E Laptop	4.1★
6	₹28,990	Dell Inspiron 15 3000 APU Dual Core A9 - (6 GB/1 TB HDD/Windows 10 Home) 3565 Laptop	3.9★
7	₹81,889	Dell Inspiron 7000 Core i7 7th Gen - (8 GB/1 TB HDD/128 GB SSD/Windows 10 Home/4 GB Graphics) 7560 Lap...	4.3★
8	₹69,990	Lenovo Core i5 7th Gen - (8 GB/2 TB HDD/Windows 10 Home/4 GB Graphics) IP 520 Laptop	4.4★
9	₹52,990	Lenovo Core i5 7th Gen - (8 GB/1 TB HDD/DOS/2 GB Graphics) IP 320-15IKB Laptop	4.3★
10	₹24,990	HP APU Quad Core A8 - (4 GB/1 TB HDD/Windows 10 Home) 15-BG004AU Laptop	4★
11	₹28,990	HP 15 Core i3 6th Gen - (4 GB/1 TB HDD/Windows 10 Home) 15-be014TU Laptop	4.1★
12	₹33,990	Dell Inspiron 3000 Core i3 6th Gen - (4 GB/1 TB HDD/Windows 10 Home) 3467 Laptop	3.8★
13	₹46,990	Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Windows 10 Home/2 GB Graphics) IP 320-15IKB Laptop	4.3★

Figure 5: Extracted data

7.Conclusion:

In this article, we have discussed about web scraping and how to scrap or extract useful data from the leading online ecommerce site Flipkart. We extracted data such as price, name and rating of laptops. Web Scraping is time efficient. Scraping plays a major role in every field either it is technology related field or not a technology related field. Web Scraping is both legal and illegal. Neither doing some illegal things, we can use it for a good purpose.

REFERENCES:

[1] Omkar S Hiremath – Web Scraping using Python
https://www.edureka.co/blog/web-scraping-with-python/amp/
[2]Web Scraping – Wikipedia https://en.m.wikipedia.org/wiki/Web_scraping
[3] Gabor Laszlo Hajba - Website Scraping with Python- https://doi.org/10.1007/978-1-4842-3925-4
[4]SeppevandenBroucke, Bart Baesens-Practical Web Scraping for Data Science –
https://doi.org/10.1007/978-1-4842- 3582-9