**WEB SCRAPING PROJECT**

* Web scraping is an automated method used to extract large amount of data from websites.
* Data on website is unstructured. Web scraping helps collects these unstructured data and store it in the structured form.
* To know whether a website allows web scraping or not, you can look at the website’s “robots.txt” file. You can find this file by appending “/robots.txt” to the URL that you want to scrape.

**How does Web Scraping work?**

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:

1. Find the URL that you want to scrape
2. Inspecting the Page
3. Find the data you want to extract
4. Write the code
5. Run the code and extract the data
6. Store the data in the required format

## ****Libraries used for Web Scraping****

As we know, Python is used for various applications and there are different libraries for different purposes. In our further demonstration, we will be using the following libraries:

* **Selenium**:  Selenium is a web testing library. It is used to automate browser activities.
* **Beautiful Soup**: 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.
* **LXML** is the most feature-rich and easy-to-use library for processing XML and HTML in the Python language. We use LXML to parse HTML content downloaded from web pages by converting it into a Tree Like structure that can be navigated programmatically using semi structured Query Languages like XPaths or CSS Selectors.

Install it using

pip install lxml

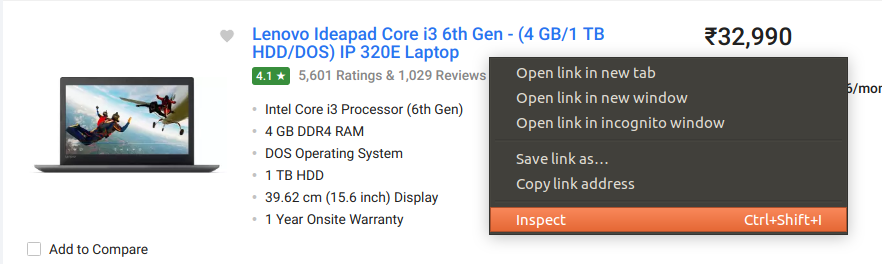
### **STEPS TO SCRAPE A WEBSITE**

### **Step 1: Find the URL that you want to scrape**

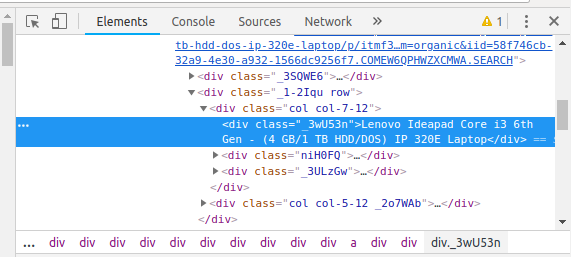
For this example, we are going scrape **Flipkart** website to extract the Price, Name, and Rating of Laptops. The URL for this page is <https://www.flipkart.com/laptops/~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniqBStoreParam1=val1&wid=11.productCard.PMU_V2>.

**Step 2: Inspecting the Page**

The data is usually nested in tags. So, we inspect the page to see, under which tag the data we want to scrape is nested. To inspect the page, just right click on the element and click on “Inspect”.



When you click on the “Inspect” tab, you will see a “Browser Inspector Box” open.



### **Step 3: Find the data you want to extract**

Let’s extract the Price, Name, and Rating which is nested in the “div” tag respectively.

### **Step 4: Write the code**

First, let’s create a Python file. To do this, open the terminal in Windows and type gedit <your file name> with .py extension.

I am going to name my file “web-s”. Here’s the command:

|  |  |
| --- | --- |
| 1 | gedit web-s.py |

Now, let’s write our code in this file.

First, let us import all the necessary libraries:

|  |  |
| --- | --- |
| 1  2  3 | from selenium import webdriver  from BeautifulSoup import BeautifulSoup  import pandas as pd |

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

|  |  |
| --- | --- |
| 1 | driver = webdriver.Chrome("/usr/lib/chromium-browser/chromedriver") |

Refer the below code to open the URL:

|  |  |
| --- | --- |
| 1  2  3  4 | products=[] #List to store name of the product  prices=[] #List to store price of the product  ratings=[] #List to store rating of the product  driver.get("<https://www.flipkart.com/laptops/>~buyback-guarantee-on-laptops-/pr?sid=6bo%2Cb5g&uniq") |

Now that we have written the code to open the URL, it’s time to extract the data from the website. As mentioned earlier, the data we want to extract is nested in <div> tags. So, I will find the div tags with those respective class-names, extract the data and store the data in a variable. Refer the code below:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | 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 code and extract the data**

To run the code, use the below command:

|  |  |
| --- | --- |
| 1 | python web-s.py |

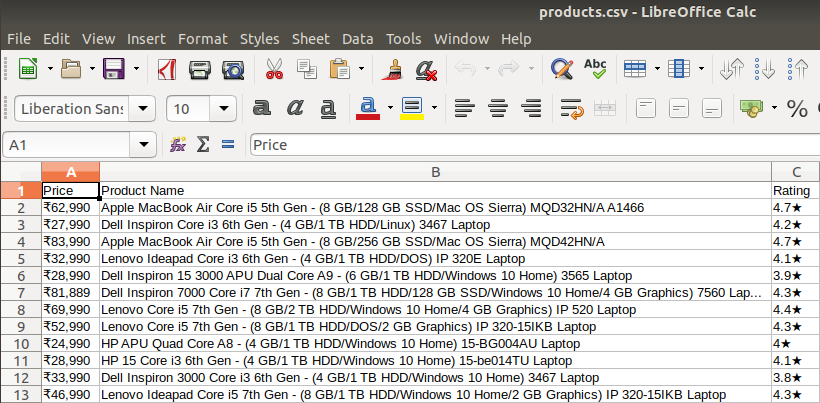
### **Step 6: Store the data in a required format**

After extracting the data, you might want to store it in a format. This format varies depending on your requirement. For this example, we will store the extracted data in a CSV (Comma Separated Value) format. To do this, I will add the following lines to my code:

|  |  |
| --- | --- |
| 1  2 | df = pd.DataFrame({'Product Name':products,'Price':prices,'Rating':ratings})  df.to\_csv('products.csv', index=False, encoding='utf-8') |

Now, I’ll run the whole code again.

A file name “products.csv” is created and this file contains the extracted data.

[](https://www.edureka.co/blog/wp-content/uploads/2018/11/output-2-1.png)

**Why Web Scraping?**

Web scraping is used to collect large information from websites. But why does someone have to collect such large data from websites? To know about this, let’s look at the applications of web scraping:

* **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.