**Web Scraping In Python:**

* 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.
* The Python libraries **requests** and **Beautiful Soup** are powerful tools for the job.

**Is Web Scraping Legal?**

* Talking about whether web scraping is legal or not, some websites allow web scraping and some don’t.
* 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.
* For this example, I am scraping Flipkart website. So, to see the “robots.txt” file, the URL is [www.flipkart.com/robots.txt.](http://www.flipkart.com/robots.txt)

Why Python Is good for Web Scraping:

* The list of features of Python which makes it more suitable for web scraping.
  + Ease of use
  + Large collection of libraries (numpy, pandas, matplotlib and seaborn)
  + Dynamically typed
  + Easily understandable syntax
  + Community

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:

Python has various applications and there are different libraries for different purposes.

* **Selenium**:  Selenium is a web testing library. It is used to automate browser activities.

In our demonstration, we will be using the following libraries:

* **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.

## Web Scraping Example : Scraping Flipkart 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”.

## 

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

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

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

## **Jump into colab notebook.**

Resources:

<https://www.edureka.co/blog/web-scraping-with-python/>