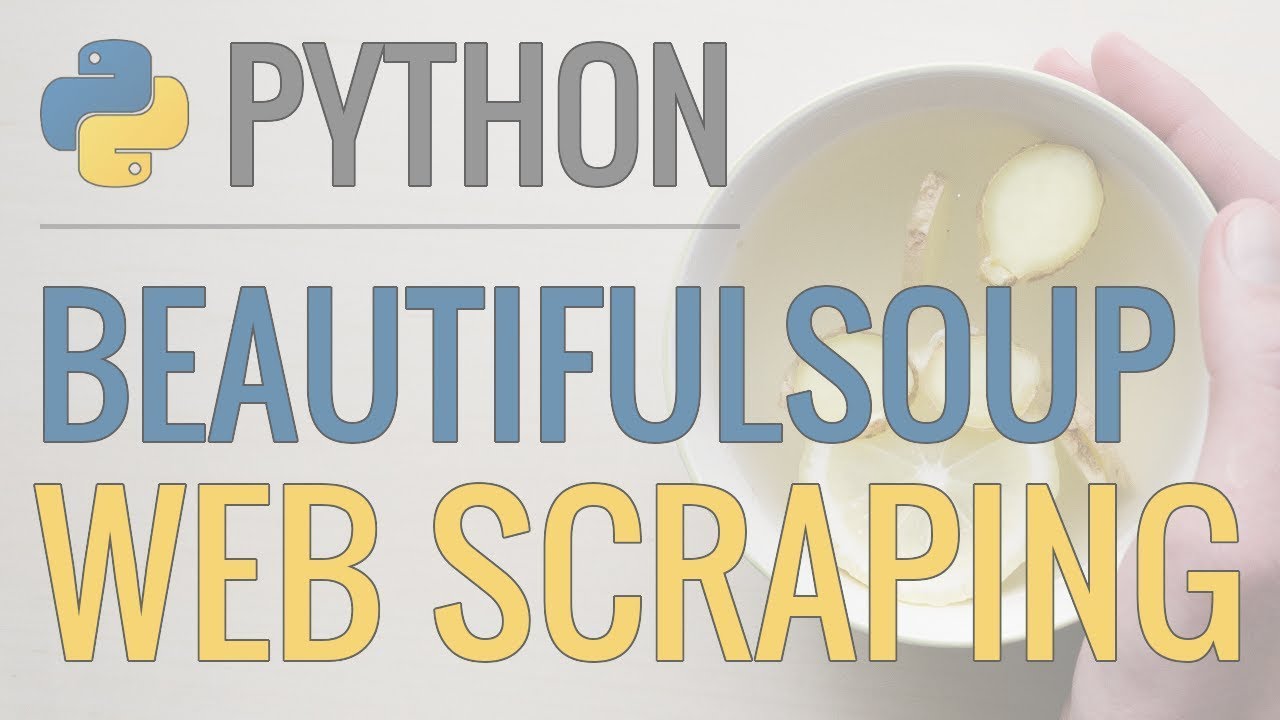
**Introduction to Web Scraping using Python**

**What is Web Scraping?**

**Web Scraping** or web data extraction is the process of extracting data from different websites through HTTP (Hyper Text Transfer Protocol) or through web browser. Web scraping is an automated process which extracts the web data and a data analyst or a data scientist can parse through the gathered data and create cluster of important data or quality data.

Web scraping using python can be done using BeautifulSoup library.

**Web Scraping with BeautifulSoup**



**BeautifulSoup** is a Python Library which converts the incoming data to Unicode format and outgoing data to UTF-8 format. BeautifulSoup parses everything you pass to it and creates tree structure on its own.

**Basics while performing Web Scraping**

While performing web scraping, we work with html tags and use elements to fetch the data. Thus one should have a good understanding of HTML.

HTML code starts with the <html> tag and ends with </html> tag.

The visible part of HTML document is between <body> and </body> tag.

To make raw HTML data more pretty, we need to parse the data by using different parsers. e.g.

1) HTML5 lib

2) XML parser

3) HTML parser

**Security**

Web scraping shouldn’t be performed without the permission of system administrator, else it may lead to illegal activity. Data Scientists many times perform web scraping on their own web pages or company data. It should be taken care that only open source sites or free sites should be used to perform web scraping.

**Working with BeautifulSoup**

Let’s try a simple code using Python3 and **BeautifulSoup**. When a web page is visited the browser sends a request to a web server. This will be a **GET** request as we are getting the file from server. Server then sends the file as response and tells the browser how to render the page. There are following types of files:

1) HTML (Hypertext Mark-up Language) – It contains the main content part of the page.

2) CSS (Cascading Style Sheet) – It makes the page look nice.

3) JS (Java Script) – They are used for web page interaction.

**HTML**

It is a mark-up language that tells the web browser how to layout the content of the page.HTML code is made up of HTML tags. The main part of html tag is head and body. e.g.:

<html>

<head>

</head>

<body>

<p>

This is paragraph of text

</p>

<p>

This is second paragraph of text

</p>

</body>

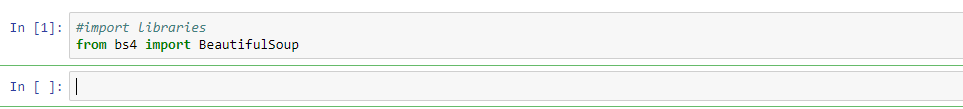
</html>

BeautifulSoup can be installed using following command,

*pip install BeautifulSoup4*

Once, the BeautifulSoup library has been installed, it’s time to use it to web scrape our data based on the following steps.

* First, the necessary libraries needs to be imported in the program using the following commands



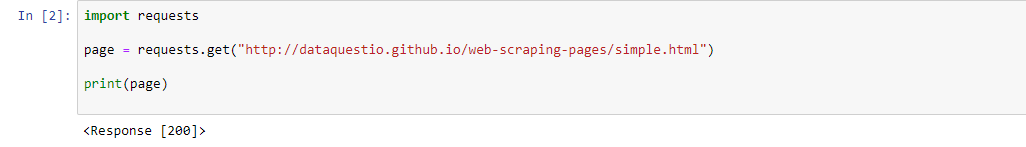
* **Requests library**

To scrape the page we need to download it and this part can be done using Python’s requests

Library. Requests makes a GET request to the web server which downloads the HTML

Contents of the page. e.g.:

Here, we have requested from [*http://dataquestio.github.io/web-scraping-pages/simple.html*](http://dataquestio.github.io/web-scraping-pages/simple.html)



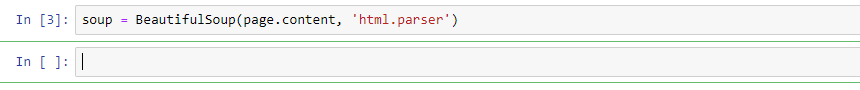
After running the above code you will get a status code which tells you whether the page is downloaded successfully or not.

**<Response [200]>**

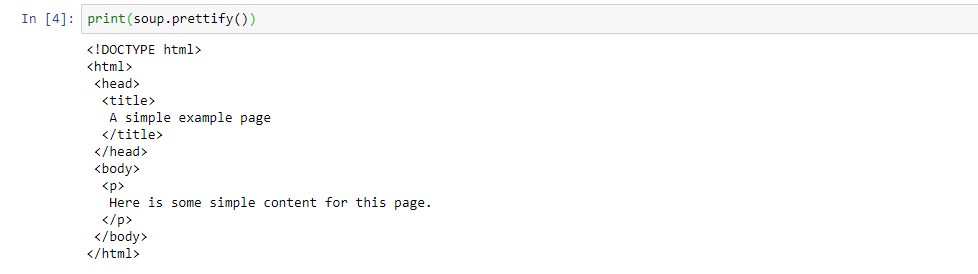
**200** status code means that the page is downloaded successfully. Status code that starts with 2 generally indicates success whereas 4 or 5 indicates error.

* **Parsing the page using BeautifulSoup**

We need to import the library in a python file and fetch the text from <p> tag.



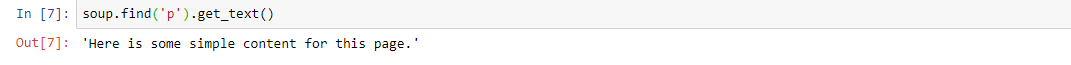
We can also print the html content of the page with the help of the soup object.



* Finding all instances of tag at once.

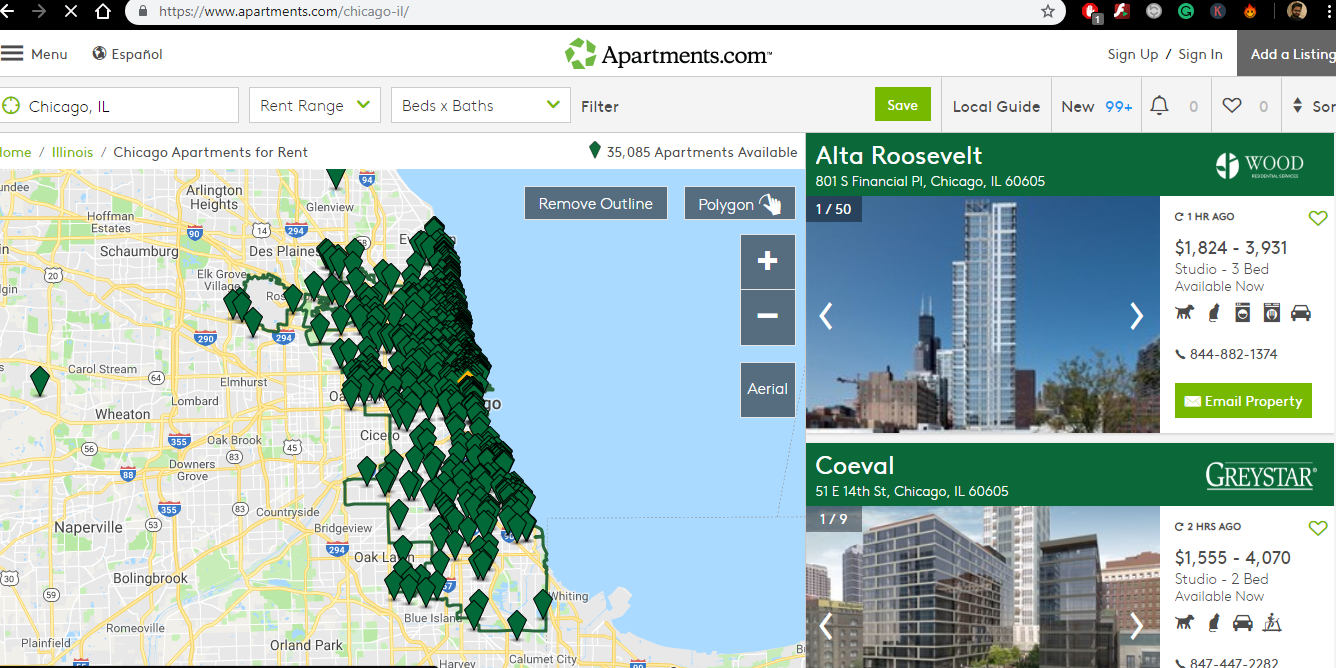


* To get text without the html tags.



**Web Scraping Example**

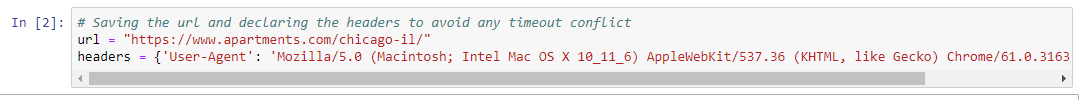
Let’s, look into another example where we would web-scrape data from the <https://www.apartments.com/chicago-il/> website as below.



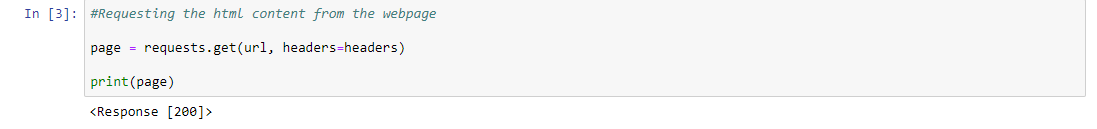
* First, the necessary libraries needs to be imported.



* Saving the URL and declaring the headers as *{'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_11\_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36'}* to avoid my TIMEDOUT conflict.



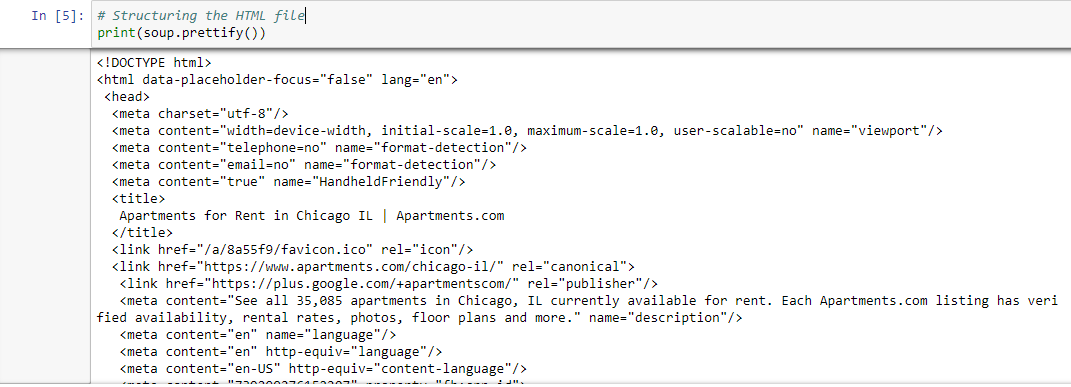
* The HTML content is requested using the GET command.



* Parsing the HTML content.

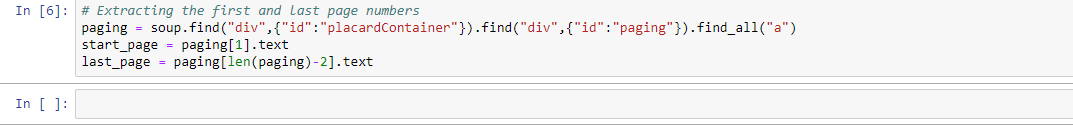


* Using prettify(), to structure our HTML content in a proper format.

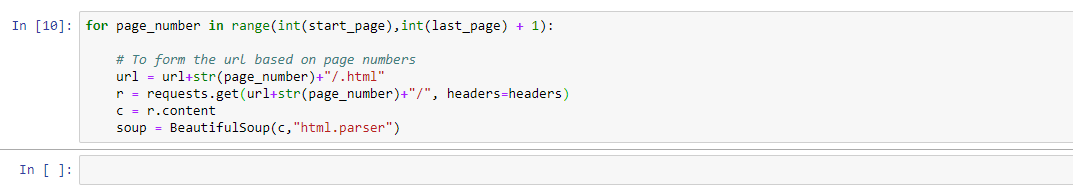


* Once, the HTML content, we would find the first and the last page of the website to get the entire data across all the pages.

After analysing this content, it is found the Page Number falls under the div id *paging* inside the div id *placardContainer*



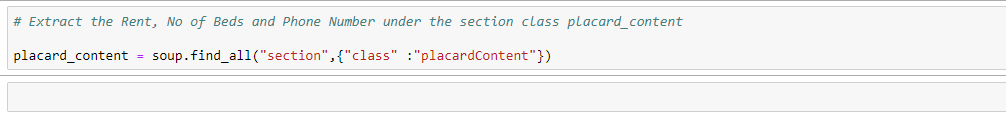
* Looping through the pages to create links from the page numbers.



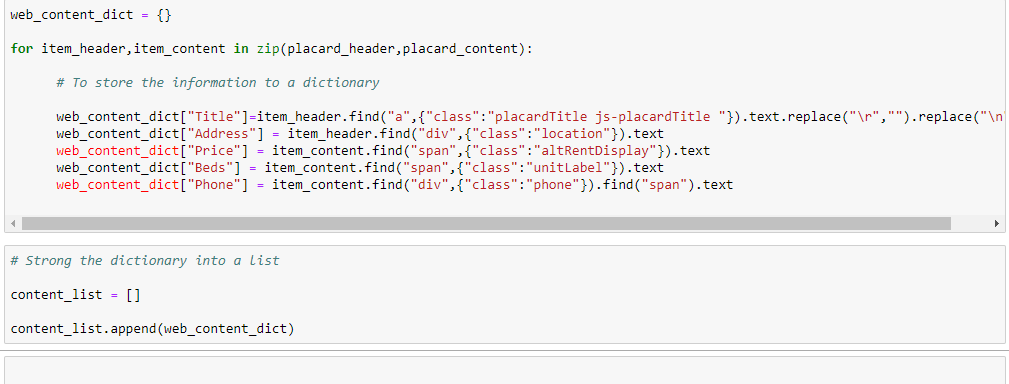
* Now, we would get the name of the property and its location.



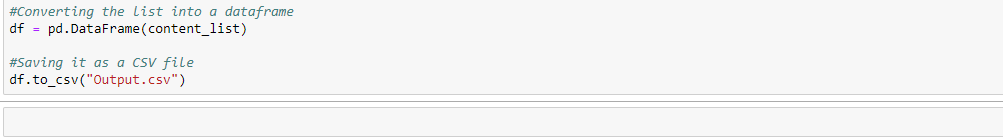
* Other details like rent, phone number.



* Getting all the placard\_header and placard\_content data in a single frame.



* The content is saved in a Data Frame and uploaded to a CSV file.



**Business examples of Web Scraping**

1) Real estate data gathering is a huge market which is growing very rapidly where the businesses are using already listed properties of different apartment, sites, farm house etc. Many real estate agents use it.

2) Email address gathering is one of the key part of B2B and B2C business. This is used by lot of business companies where they use the email-id of a person to contact and send the information of their product. Email-id is also used to get more details of a person, for e.g.: using email-id of a person, we can get his/her professional details by searching him on LinkedIn.

3) Social media data is the most valuable data in today’s century. Collecting data from different social media platform and using that data to attract more customers is one of the best business model in today’s world. Providing quality data to an app company will help them to fetch more customer by giving the customer what they really want and not what the company is providing. e.g.:

A food app can get to know a customer’s favourite dish and provide him the same dish for cheaper price with attractive discounts.

**Advantages of Web Scraping**

1) It reduces the cost of obtaining information from the web.

2) Easy to learn and implement.

3) The accuracy of data extracted using BeautifulSoup is very much accurate compared to other API’s. The accuracy is very important when data is extracted from websites which deal with sales price, real estate price, stock price or any other kind of financial data because even slight amount of change in the number may cause huge loss to a company.

**Disadvantages of Web Scraping**

1) Scraping may cause slight confusion while processing the data because of the html tags.

2) Some websites don’t allow users to scrape the data which may lead to illegal activity.

3) BeautifulSoup mostly works using html tag, this can also be considered as a disadvantage because if a developer has scraped the data of a website based on the previous version.

**Conclusion**

Web-Scraping is an important concept to master these days due to the enormous amount of unstructured data that’s available to us.

The understanding of regular expression with Python is crucial for web-scraping tasks - <https://www.w3schools.com/python/python_regex.asp>

Additional resource - <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>