**Session – 29: Project Web Scrapping Project**

1. Install required module i.e.
   1. pandas
   2. request
   3. datetime
   4. bs4 import BeautifulSoup
2. Firstly, request for response to get data access i.e. url = <https://en.wikipedia.org./wiki/List_of_countries_and_dependencies_by_population>
3. Then, read url using request module and save in variable response (if return answer 200 than you get access of specific website).

*(****Handling Different Data Types while Web Scrapping***

*Web pages can contain various types of data, such as text, numbers, dates, and even images and videos. When we scrape data from a web page, we need to be aware of the data types we are dealing with.*

***Text:*** *Text is the most common form of data you’ll extract from a web page. In BeautifulSoup, you can use the* ***.text*** *or* ***.get\_text()*** *methods to extract text from a tag. You can also use the* ***.string*** *attribute to get the exact string within a tag.*

***Numbers:*** *Numbers are usually represented as text in HTML. After extracting the test, you can convert it to an integer or float using the* ***int()*** *or* ***float()*** *functions.*

***Dates:*** *Dates can be tricky because they can be in different formats. You might need to use the* ***datetime*** *module to parse and format dates.*

***Images and Videos:*** *To scape images or videos, you typically extract the URL of the image or video file rather than the content itself. In BeautifulSoup, you can get the* ***src*** *attribute of an* ***img*** *or* ***video*** *tag to get the* ***URL.***

*Remember; it’s important to inspect the HTML content of the web page to understand the structure and data types before you start scraping.*

***The are two types of errors we typically encounter when web scraping.***

***Connection Errors:*** *These occur when there is a network problem like a DNS failure (when the domain name cannot be converted into its corresponding IP address) or a refused connection (when the server refuses to respond). A common exception to this is* ***requests.exceptions.RequestException***

***HTTP Errors:*** *These occur when an HTTP request returns an unsuccessful status code. For example, a 404 Not Found error mean that the requested resource could not be found on the server, and a 500 Internal Server Error means that the server encountered an unexpected condition. A common exception for this is* ***requests.exceptions.HTTPError****.)*

1. Extract the html\_text.
2. Create instance of BeautifulSoup and parse with html\_parser and save in the variable namely soup.
3. Now, extract the table.
   1. As we know table is the rows and columns. But rows further classify into th = table headers and td = table data. For example, header is the title row, while data is the remaining bottom rows.
   2. Moreover, to know about table tag, then you can inspect and check the table tag.
4. Find out specific table class define with using soup.find and save it in the variable namely table.
5. Show table.
6. Firstly, extract all table rows of the table with the help of method find\_all and pass params “tr” and save it in the variable namely rows.
7. Second, extract all the table th = table headers from the rows. Use for loop in rows and find out table header using find\_all method params ‘th’ and save it in th\_lst, and once again using for loop from th\_lst and find out the headers and it append in the empty headers list with Data cleaning i.e. beginner and ending spaces remove.
   1. Print the array of headers.
   2. Fill the null value index with Rank and Notes respectively.
8. Third, extract all the table data = table data from the rows. Use for loop in rows and find out table data using find\_all method params ‘td’ and save it in variable namely td, and create a empty list namely table\_data then once again using for loop from td and find out the table data and it extract with .text attribute then it append in the empty table\_data list, then all table data list save in the empty list namely final\_data but before it append we can do the Data cleaning i.e. beginner and ending spaces remove.
   1. Print the array of final\_data.
9. Now check if you have any empty list than you can slicing.
10. And check the first column of data i.e. ranking data if any data misplace than set the ranking.
11. Now convert all the data into pandas data frame using pd.DataFrame and pass params the data set which extracted and save it in variable namely pop.
12. Now, we assign our header list to the datafram using .columns attribute.
13. **Data Cleaning:**
    1. Convert the **population** column into integer using for loop and save it in empty list namely popul and update column with data cleaning.
    2. Convert the **% of world** column into float with data cleaning.
    3. Check info.
    4. Formatting the date.
       1. Firstly, check datetime type and data.
       2. Formatting the datetime to create a function name convert\_date and params date then use if else to check the date length that if length of my date is less than 10 than pass to try and except, else apply full format with dt.datetime.strptime and parse it with date demo and format, and save in the variable namely date and return date.
       3. Standardized the Date column, then apply convert\_date function to the data frame with selecting date column.
       4. Then updated the date column.
       5. Check data info
14. Save all the data with convergent method to use pandas method to\_csv with name population.csv, and if I don’t want to use index than second params in the to\_csv method of index=False.
15. Now, your file created / Saved.