**Web Scrapping Documentation**

**Step by step breakdown to the code**

*from bs4 import beautifulSoup -> import beautiful library,which is used for parsing HTML content*

*import requests -> imports requests library,which is used to send HTTP GET request to website*

*import pandas as pd -> import pandas,used to data manipulation and analysis*

*url = “www.somewejsajdlkj.com/dsakdj” -> line defines url of the web page u want to scrape*

*page = request.get(url) -> this line sends an HTTP GET request to the specified URL using the ‘request.get()’ method and store the response in ‘page’ variable*

*soup = BeautifulSoup(page.text,’lxml’) - > create BeautifulSoup object named ‘soup’ by parsing HTML content of the web page obtained from page.text.the ‘lxml’ argument specifies the parser to USE*

*table = soup.find\_all(‘table’)[1] - > this line find all <table> element from the page,and select table index 1(second table)*

*world\_titles = table.find\_all(‘th’) -> find all th(table header) from selected table and stores it in world\_title variable*

*world\_table\_titles = [ title.text.strip() for title in world\_title ] -> extract the text content of ‘th’ element.strip() is for removing leading and trailing white space*

*df = pd.DataFrame(columns = world\_table\_titles) ->create empty pandas Data Frame named df with column named after the table header titles*

*column\_data = table.find\_all(‘tr’) -> find all table row elements and stored in column data*

*for row in column\_data[1:] -> this line start a for loop that iterate through each table row(starting from second row…first row is header)*

*row\_data = row.find\_all(‘td’) -> inside the loop this line find all table data element within the current table row and stores them in the ‘row data’ variable*

*individual\_row\_data = [data.text.strip() for data in row\_data] -> extract text content of each ‘td’ in the row\_data.*

*Length = len(df)*

*df.loc[length] = Individual\_row\_data -> this line calculate the current length of dataframe ‘df’ and uses it as an index the ‘individual\_row\_data’ as a new row in the DataFrame*

*df.to\_csv(‘text.csv’,index = False) -> save df as a csv in current directory using to\_csv() method.The index = False ensures that the DataFrame’s Index is not included on the CSV*