**Digital Empowerment Pakistan**

# Python Programming

**Submitted By:**

**Farwa Rubab**

**Task 02:**

## Create a web scraper using libraries like Beautiful Soup and requests to extract data from a website and store it in a CSV file.

Following are the steps for this task**:**

### 1. Install Required Libraries:

Firstly,I have insatalled required libraries by this command:

**pip install requests beautifulsoup4 pandas**

### 2. Import Libraries:

Import the desired libraries by following command lines:

```
import requests

from bs4 import BeautifulSoup

import pandas as pd
```

In this step, we import the necessary libraries:

- requests for sending HTTP requests.
- BeautifulSoup from the bs4 module for parsing HTML.
- pandas for handling and manipulating data.

### 3. Send a Request to the Website:

For this example, let's scrape the table from the Wikipedia page for the list of countries and dependencies by population. The URL is: https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population.

```
url = 'https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population'

response = requests.get(url)

if response.status_code != 200:

    print('Failed to retrieve the webpage')

    exit()
```

Here, we:

- Define the URL of the Wikipedia page we want to scrape.
- Use requests.get to send a GET request to the URL.
- Check the response status code. If it is not 200 (OK), we print an error message and exit the script.

## 4. Parse the HTML Content:

**soup = BeautifulSoup(response.content, 'html.parser')**

In this step, we:

- Pass the content of the response to BeautifulSoup to create a soup object.
- Specify 'html.parser' as the parser to be used.

## 5. Extract Data from the Table

**# Step 3: Extract data**

**table = soup.find('table', {'class': 'wikitable'})**

**rows = table.find_all('tr')**

Here, we:

- Find the table with the class wikitable using soup.find.
- Retrieve all the rows in the table using table.find_all('tr').

## 6. Store Data in a List

**data = []**

**for row in rows:**

**cols = row.find_all(['th', 'td']) # Use 'th' for headers and 'td' for data**

**cols = [col.text.strip() for col in cols]**

**data.append(cols)**

In this step, we:

- Initialize an empty list data.

- Loop through each row in rows.
- For each row, find all header (th) and data (td) cells.
- Extract and strip the text from each cell.
- Append the list of cell texts to the data list.

## 7. Convert Data to a DataFrame and Save to CSV:

df = pd.DataFrame(data[1:], columns=data[0])  # Use the first row as headers

df.to_csv('output.csv', index=False)

print('Data has been successfully written to output.csv')

Here, we:

- Create a DataFrame from the data list, using the first row as headers and the subsequent rows as data.
- Save the DataFrame to a CSV file named output.csv without the index column.
- Print a confirmation message indicating that the data has been successfully written to the CSV file.

## 8.Read the Csv file:

This script reads the `output.csv` file and displays its contents.

**import pandas as pd**

**# Read the CSV file**

**df = pd.read_csv('output.csv')**

**# Display the contents of the DataFrame**

**print(df)**