Webscraping Procedure :

**Scrape data from HTML tables into a DataFrame using BeautifulSoup and Pandas**

**Method 1**

1. Install the necessary libraries for python webscraping here are the list:

* !pip install bs4 (Beautiful Soup)
* !pip install request
* !pip install lxml
* !pip install html5lib

1. Import the required modules, function and libraries for the python webscraping:

* Import requests
* From bs4 import BeautifulSoap

1. Get the URL of the webpage, be sure to check the table of the page and its structure.
2. Retrieve the webpage in text format using the request.get().text function.
3. Create an instance of Beautiful Soup Object.
4. Perform some checking of the data using the len(), find\_all(), matching search and

find ()

1. Performa manual coding assignment of adding the element to the dataframe.

**Method 2**

1. Install the necessary libraries for python webscraping here are the list:

* !pip install bs4 (Beautiful Soup)
* !pip install request
* !pip install lxml
* !pip install html5lib

1. Import the required modules, function and libraries for the python webscraping:

* Import requests
* From bs4 import BeautifulSoap

1. Get the URL of the page.
2. Create a Beautiful Soap Object
3. Perform search and matching to determine the index of the table and locate a specific table.
4. After determining the index sample tables[index] load it to pandas read\_html() with the following structure:

*population\_data\_read\_html = pd.read\_html(str(tables[3]), flavor='bs4')[0]*

note*: understand the meaning of [0] at the end of the code because the display of the data affected if this part of the code is omitted.*

**Method 3**

1. Load the URL directly to the read\_html function.

Sample :

dataframe\_list = pd.read\_html(url, flavor='bs4')

*note: check the format before loading if it’s an html or not.*

1. Perform the Pandas Data Manipulation.
2. A matching parameter can be done in the following format:

*pd.read\_html(url, match="10 most densely populated countries", flavor='bs4')[0]*