```
!pip install BS4
!pip install Requests
```

```
Requirement already satisfied: BS4 in /usr/local/lib/python3.7/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from BS4) (4.6.3)
Requirement already satisfied: Requests in /usr/local/lib/python3.7/dist-packages (2.23.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from Requests) (2021.5.30)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from Requests) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from Requests) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from Requests) (3.0.4)
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
import csv
import requests
from bs4 import BeautifulSoup
from IPython.display import HTML
```

```
url = 'https://www.worldometers.info/world-population/population-by-country/'
headers = {'User-Agent': "Chrome/54.0.2840.90"}
response = requests.get(url, headers=headers, verify=False)
html = response.content
```

```
soup = BeautifulSoup(html, "html.parser")
#print(soup)
table = soup.find('tbody')
```

```
tmpRow = (table.findAll('tr'))
# print(tmpRow)
# The <tr> tag specifies a row in an HTML table.
# Each table data/cell is defined with a <td> tag.
```

```
list_of_rows = []
try:
    outfile = open("./population_by_country.csv", "w")  # create an empty file in write mode
    writer = csv.writer(outfile)

    # Add header to the file
    writer.writerow(["No.", "Country (or dependency)", "Population(2020)", "Yearly Change", "Net Change", "Density(P/Km²)", "Land Area(Km²)", "Migrants(net)",  "Fert.Rate",
                     "Med.Age", "Urban Pop%", "World Share"])
    for row in table.findAll('tr'):  # Get all the table rows and each row get the values.
        list_of_cells = []
        for cell in row.findAll("td"):
            text = cell.text.replace(' ', '')   # nbsp - non breaking space
            list_of_cells.append(text)
        arrLength = len(list_of_cells)
        writer.writerow(list_of_cells)
finally:
    outfile.close()
```

```
# data cleaning
import pandas as pd
import numpy as np

df = pd.read_csv('./population_by_country.csv')
df.head(10)
```

	No.	Country (or dependency)	Population(2020)	Yearly Change	Net Change	Density(P/Km²)	Land Area(Km²)	Migrants(net)	Fert.Rate	Med.Age	Urban Pop%	World Share
0	1	China	1,439,323,776	0.39 %	5,540,090	153	9,388,211	-348,399	1.7	38	61 %	18.47 %
1	2	India	1,380,004,385	0.99 %	13,586,631	464	2,973,190	-532,687	2.2	28	35 %	17.70 %
2	3	United States	331,002,651	0.59 %	1,937,734	36	9,147,420	954,806	1.8	38	83 %	4.25 %
3	4	Indonesia	273,523,615	1.07 %	2,898,047	151	1,811,570	-98,955	2.3	30	56 %	3.51 %
4	5	Pakistan	220,892,340	2.00 %	4,327,022	287	770,880	-233,379	3.6	23	35 %	2.83 %
5	6	Brazil	212,559,417	0.72 %	1,509,890	25	8,358,140	21,200	1.7	33	88 %	2.73 %
6	7	Nigeria	206,139,589	2.58 %	5,175,990	226	910,770	-60,000	5.4	18	52 %	2.64 %
7	8	Bangladesh	164,689,383	1.01 %	1,643,222	1,265	130,170	-369,501	2.1	28	39 %	2.11 %
8	9	Russia	145,934,462	0.04 %	62,206	9	16,376,870	182,456	1.8	40	74 %	1.87 %
9	10	Mexico	128,932,753	1.06 %	1,357,224	66	1,943,950	-60,000	2.1	29	84 %	1.65 %

```
# delete the extra # column
df = df.drop(columns=['No.'])
df.head(10)
```

	Country (or dependency)	Population(2020)	Yearly Change	Net Change	Density(P/Km²)	Land Area(Km²)	Migrants(net)	Fert.Rate	Med.Age	Urban Pop%	World Share
0	China	1,439,323,776	0.39 %	5,540,090	153	9,388,211	-348,399	1.7	38	61 %	18.47 %
1	India	1,380,004,385	0.99 %	13,586,631	464	2,973,190	-532,687	2.2	28	35 %	17.70 %
2	United States	331,002,651	0.59 %	1,937,734	36	9,147,420	954,806	1.8	38	83 %	4.25 %
3	Indonesia	273,523,615	1.07 %	2,898,047	151	1,811,570	-98,955	2.3	30	56 %	3.51 %
4	Pakistan	220,892,340	2.00 %	4,327,022	287	770,880	-233,379	3.6	23	35 %	2.83 %
5	Brazil	212,559,417	0.72 %	1,509,890	25	8,358,140	21,200	1.7	33	88 %	2.73 %
6	Nigeria	206,139,589	2.58 %	5,175,990	226	910,770	-60,000	5.4	18	52 %	2.64 %
7	Bangladesh	164,689,383	1.01 %	1,643,222	1,265	130,170	-369,501	2.1	28	39 %	2.11 %
8	Russia	145,934,462	0.04 %	62,206	9	16,376,870	182,456	1.8	40	74 %	1.87 %
9	Mexico	128,932,753	1.06 %	1,357,224	66	1,943,950	-60,000	2.1	29	84 %	1.65 %

```
df.shape
```

```
(235, 11)
```

```
df = df.head(50)
```

```
df.to_csv("./cleanPopulationByCountry.csv")
```