Python script to scrap web data

In [1]:

```python
# Importing required libraries
from bs4 import BeautifulSoup
import requests
import pandas as pd
from IPython.display import Image
```

In [2]:

```python
# Importing the source data via url.
source_data_url = 'https://www.worldometers.info/coronavirus/#main_table
page = requests.get(source_data_url)
soup = BeautifulSoup(page.text, 'html')
```

In [3]:

```python
# Extracting the title of the webpage
title = soup.find("title")
title.text
```

Out[3]:

```
'COVID - Coronavirus Statistics - Worldometer'
```

'COVID - Coronavirus Statistics - Worldometer'

In [4]:

```python
Image(url="https://www.worldometers.info/img/worldometers-fb.jpg", heigh
```

Out[4]:

In [5]:

```python
# Extracting table of reported cases and deaths by country or territory
table = soup.find("table")
```

In [6]:

```
# Printing table to get class of the table
table #(class="table table-bordered table-hover main_table_countries")
```

Out[6]:

```
<table class="table table-bordered table-hover main_table_countries"
id="main_table_countries_today" style="width:100%;margin-top: 0px !im
portant;display:none;">
<thead>
<tr>
<th width="1%">#</th>
<th width="100">Country,<br/>Other</th>
<th width="20">Total<br/>Cases</th>
<th width="30">New<br/>Cases</th>
<th width="30">Total<br/>Deaths</th>
<th width="30">New<br/>Deaths</th>
<th width="30">Total<br/>Recovered</th>
<th width="30">New<br/>Recovered</th>
<th width="30">Active<br/>Cases</th>
<th width="30">Serious,<br/>Critical</th>
<th width="30">Tot Cases/<br/>1M pop</th>
<th width="30">Deaths/<br/>1M pop</th>
<th width="30">Total<br/>Tests</th>
<th width="30">Tests/<br/>
```

In [7]:

```
# Extracting the table to get the daily trend of reported cases
table= soup.find('table', class_ = 'table table-bordered table-hover mai
table
```

Out[7]:

```
<table class="table table-bordered table-hover main_table_countries"
id="main_table_countries_today" style="width:100%;margin-top: 0px !im
portant;display:none;">
<thead>
<tr>
<th width="1%">#</th>
<th width="100">Country,<br/>Other</th>
<th width="20">Total<br/>Cases</th>
<th width="30">New<br/>Cases</th>
<th width="30">Total<br/>Deaths</th>
<th width="30">New<br/>Deaths</th>
<th width="30">Total<br/>Recovered</th>
<th width="30">New<br/>Recovered</th>
<th width="30">Active<br/>Cases</th>
<th width="30">Serious,<br/>Critical</th>
<th width="30">Tot Cases/<br/>1M pop</th>
<th width="30">Deaths/<br/>1M pop</th>
<th width="30">Total<br/>Tests</th>
<th width="30">Tests/<br/>
```

In [8]:

```
# Extracting headers from the table
worldOmeter_columns = table.find_all('th')
worldOmeter_columns
```

Out[8]:

```
[<th width="1%">#</th>,
 <th width="100">Country,<br/>Other</th>,
 <th width="20">Total<br/>Cases</th>,
 <th width="30">New<br/>Cases</th>,
 <th width="30">Total<br/>Deaths</th>,
 <th width="30">New<br/>Deaths</th>,
 <th width="30">Total<br/>Recovered</th>,
 <th width="30">New<br/>Recovered</th>,
 <th width="30">Active<br/>Cases</th>,
 <th width="30">Serious,<br/>Critical</th>,
 <th width="30">Tot Cases/<br/>1M pop</th>,
 <th width="30">Deaths/<br/>1M pop</th>,
 <th width="30">Total<br/>Tests</th>,
 <th width="30">Tests/<br/>
 <nobr>1M pop</nobr>
 </th>,
 <th width="30">Population</th>,
 <th style="display:none" width="30">Continent</th>,
 <th width="30">1 Case<br/>every X ppl</th>,
```

In [9]:

```
# Creating a list of headers to get columns of the data frame
worldOmeter_columns = [title.text.strip() for title in worldOmeter_colum
worldOmeter_columns
```

Out[9]:

```
['#',
 'Country,Other',
 'TotalCases',
 'NewCases',
 'TotalDeaths',
 'NewDeaths',
 'TotalRecovered',
 'NewRecovered',
 'ActiveCases',
 'Serious,Critical',
 'Tot\xa0Cases/1M pop',
 'Deaths/1M pop',
 'TotalTests',
 'Tests/\n1M pop',
 'Population',
 'Continent',
 '1 Caseevery X ppl',
 '1 Deathevery X ppl',
 '1 Testevery X ppl',
 'New Cases/1M pop',
 'New Deaths/1M pop',
 'Active Cases/1M pop']
```

In [10]:

```python
# Extracting table rows after the 8th row. The actual dataset starts aft
table_row=table.find_all('tr')[8:]
table_row
```

Out[10]:

```
[<tr class="total_row_world">
 <td></td>
 <td style="text-align:left;">World</td>
 <td>702,690,536</td>
 <td>+718</td>
 <td>6,978,469</td>
 <td>0</td>
 <td>673,565,917</td>
 <td>+11,159</td>
 <td>22,146,150</td>
 <td>36,260</td>
 <td>90,149</td>
 <td>895.3</td>
 <td></td>
 <td></td>
 <td></td>
 <td data-continent="all" style="display:none">All</td>
 <!-- 1 Case every X -->
 <td>
```

In [11]:

```python
# Create an empty data frame with columns using headers
df=pd.DataFrame(columns=worldOmeter_columns)
df
```

Out[11]:

```
#   Country,Other   TotalCases   NewCases   TotalDeaths   NewDeaths   TotalRecovered   New
```

0 rows × 22 columns

In [12]:

```
# Extracting data from each row and adding them to the data frame
for row in table_row:
        # find all the data from the  row
        row_data = row.find_all('td')
        # Creating a list of the data from the row
        individual_row_data=[data.text.strip() for data in row_data]
        # intializing length by len(df)=0
        length=len(df)
        # Adding the individual_row_data as a new row in the data frame
        df.loc[length]= individual_row_data

df
```

Out[12]:

	#	Country,Other	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovere
0		World	702,690,536	+718	6,978,469	0	673,565,91
1	1	USA	110,848,567		1,195,303		108,565,52
2	2	India	45,025,792		533,451		N/
3	3	France	40,138,560		167,642		39,970,91
4	4	Germany	38,809,615		181,918		38,240,60
...	...	...	...	...	...	...	
235		Total:	69,687,559		1,365,132		66,628,55
236		Total:	14,790,096	+718	32,484		14,555,42
237		Total:	12,859,144		258,884		12,089,89
238		Total:	721		15		70

In [13]:

```
# drop the serial number
df.drop(columns="#",axis=1,inplace=True)
df
```

Out[13]:

	Country,Other	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	Ne
0	World	702,690,536	+718	6,978,469	0	673,565,917	
1	USA	110,848,567		1,195,303		108,565,520	
2	India	45,025,792		533,451		N/A	
3	France	40,138,560		167,642		39,970,918	
4	Germany	38,809,615		181,918		38,240,600	
...	...	...	...	...	...	...	
235	Total:	69,687,559		1,365,132		66,628,557	
236	Total:	14,790,096	+718	32,484		14,555,421	
237	Total:	12,859,144		258,884		12,089,893	
238	Total:	721		15		706	
239	Total:	702,690,536	+718	6,978,469	0	673,565,917	

240 rows × 21 columns

In [14]:

```
df.tail(10)
```

Out[14]:

```
         Country,Other   TotalCases  NewCases  TotalDeaths  NewDeaths  TotalRecovered
230         MS Zaandam            9                      2                          7
231              China      503,302                  5,272                    379,053
232             Total:  130,877,517              1,670,561                126,345,833
233             Total:  221,425,324              1,552,773                205,582,218
234             Total:  253,050,175              2,098,620                248,363,289
235             Total:   69,687,559              1,365,132                 66,628,557
236             Total:   14,790,096      +718       32,484                 14,555,421
237             Total:   12,859,144                258,884                 12,089,893
238             Total:          721                     15                        706
239             Total:  702,690,536      +718    6,978,469          0     673,565,917
```

In [15]:

```
# Rows from index 232 to 239 are the total of columns continentwise.
# Renaming total  via continent name in Country column
df.iloc[232]["Country,Other"]="North America"
df.iloc[233]["Country,Other"]="Asia"
df.iloc[234]["Country,Other"]="Europe"
df.iloc[235]["Country,Other"]="South America"
df.iloc[236]["Country,Other"]="Australia/Oceania"
df.iloc[237]["Country,Other"]="Africa"
df.iloc[238]["Country,Other"]="Not_Availbale"
df.iloc[239]["Country,Other"]="World"
```

In [16]:

```
df.tail(10)
```

Out[16]:

```
         Country,Other   TotalCases  NewCases  TotalDeaths  NewDeaths  TotalRecovered
230         MS Zaandam            9                      2                          7
231              China      503,302                  5,272                    379,053
232      North America  130,877,517              1,670,561                126,345,833
233               Asia  221,425,324              1,552,773                205,582,218
234             Europe  253,050,175              2,098,620                248,363,289
235      South America   69,687,559              1,365,132                 66,628,557
236  Australia/Oceania   14,790,096      +718       32,484                 14,555,421
237             Africa   12,859,144                258,884                 12,089,893
238      Not_Availbale          721                     15                        706
239              World  702,690,536      +718    6,978,469          0     673,565,917

10 rows × 21 columns
```

In [17]:

```
# exporting data frame into csv/excel file
df.to_csv(r'D:\Python_project\COVID.csv', index = False)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: