

COVID-19 tracker

<https://www.worldometers.info/coronavirus> (<https://www.worldometers.info/coronavirus>)

Worldometers has a credible sources of COVID19 data.

In this exercise, we will learn how to get data from Worldometers website

We will save the data to a csv file, and practice how to plot the data.

```
In [1]: # import required libraries

#pip install requests==2.25.1
#pip install BeautifulSoup4==4.9.3

import requests as req
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
```

1. Get data

```
In [2]: # request data from website
html = req.get("https://www.worldometers.info/coronavirus")
```

```
In [3]: # check downloaded content
html.content
```

2. Parse html

```
In [4]: # parse html with BeautifulSoup
html_parsed = BeautifulSoup(html.content)
```

```
In [5]: # search for the required table
table = html_parsed.find('table', attrs={'id': 'main_table_countries_today'})
```

```
In [6]: # check result
table
```

```
In [7]: # get all the rows
rows = table.find_all("tr")
```

```
In [8]: # check result
rows[0]
```

```
Out[8]: <tr>
<th width="1%">#</th>
<th width="100">Country,<br>Other</br></th>
<th width="20">Total<br>Cases</br></th>
<th width="30">New<br>Cases</br></th>
<th width="30">Total<br>Deaths</br></th>
<th width="30">New<br>Deaths</br></th>
<th width="30">Total<br>Recovered</br></th>
<th width="30">New<br>Recovered</br></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><th width="30">1 Death<br></>every X pp
l</th><th width="30">1 Test<br></>every X ppl</th>
</tr>
```

```
In [9]: rows[0].text.strip()
```

```
Out[9]: '#\nCountry,Other\nTotalCases\nNewCases\nTotalDeaths\nNewDeaths\nTotalRecovered\nNewRecovered\nActiveCases\nSerious,Critical\nTot\xa0Cases/1M pop\nDeaths/1M p
op\nTotalTests\nTests/\n1M pop\n\nPopulation\nContinent\n1 Caseevery X ppl1 Dea
thevery X ppl1 Testevery X ppl'
```

```
In [10]: # tokenization
rows[9].text.strip().split("\n")
```

```
Out[10]: ['1',
'USA',
'20,216,991',
'',
'350,778 ',
'',
'11,998,794',
'',
'7,867,419',
'29,312',
'60,900',
'1,057',
'251,765,894',
'758,397',
'331,970,957 ',
'North America',
'169461']
```

3. Store data

```
In [11]: # store rows into list (data).
data = []
for x in rows:
    data.append(x.text.strip().split("\n")[1:5]) # get only the first 9 columns
```

```
In [12]: # convert List into DataFrame
df = pd.DataFrame(data)
```

```
In [13]: # check the DataFrame
df.head()
```

Out[13]:

	0	1	2	3
0	Country,Other	TotalCases	NewCases	TotalDeaths
1		23,185,914	+13,327	510,697
2		20,621,667	+23,236	336,548
3		13,112,940	+1,485	361,101
4		23,468,089	+88,571	540,394

```
In [14]: # set the first row as the header, and remove the second row
df = pd.DataFrame(data[9:], columns=data[0])
```

```
In [15]: # check the DataFrame
df.head()
```

Out[15]:

	Country,Other	TotalCases	NewCases	TotalDeaths
0	USA	20,216,991		350,778
1	India	10,267,283		148,774
2	Brazil	7,619,970		193,940
3	Russia	3,159,297	+27,747	57,019
4	France	2,600,498		64,381

```
In [16]: # save as csv file
df.to_csv('covid19.csv')
```

4. visualize

```
In [17]: # get the required columns.
df_plot=df[['Country,Other','TotalCases']]

# get first 10 rows
df_plot = df_plot[:10]
```

```
In [18]: # check the DataFrame
df_plot.head()
```

Out[18]:

	Country,Other	TotalCases
0	USA	20,216,991
1	India	10,267,283
2	Brazil	7,619,970
3	Russia	3,159,297
4	France	2,600,498

```
In [19]: # remove commas in digits, and convert string to int
df_plot['TotalCases'] = df_plot['TotalCases'].apply(lambda x: x.replace(',',''))
```

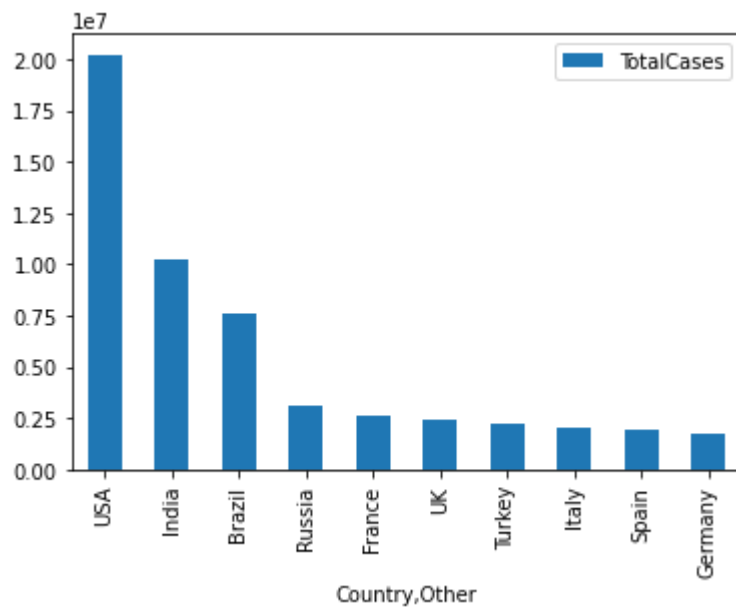
```
In [20]: # check DataFrame
df_plot.head()
```

Out[20]:

	Country,Other	TotalCases
0	USA	20216991
1	India	10267283
2	Brazil	7619970
3	Russia	3159297
4	France	2600498

```
In [21]: # plot
df_plot.plot(kind='bar',x='Country,Other',y='TotalCases')
```

```
Out[21]: <AxesSubplot:xlabel='Country,Other'>
```



```
In [ ]:
```