In [1]:

```
import requests
# to send a request to the page which you are scrapping whether
# we can scrap or no
```

In [2]:

```
from bs4 import BeautifulSoup
# to import beautifulsoup from bs4 module. we use this to pars and fetch
# the details what we want
```

In [3]:

```
import pandas as pd
#to convert the data in a tabular form
```

In [4]:

```
url= 'https://www.worldometers.info/coronavirus/'
```

In [5]:

```
# get method helps us to find the reponse from the site whether its is
# scrapable or not (if we get 200 than we can scrap it otherwise we cannot)
page1=requests.get(url)
page1
```

Out[5]:

```
<Response [200]>
```

In [6]:

```
# beautifulsoup helps to extract data in html from the site and if the data
# is in readable form than we dont need to do any thing otherwise we need to use a function
# of beautifulsoup which is called as soup.prettify.
soup=BeautifulSoup(page1.text)
soup
```

Out[6]:

```


<!DOCTYPE html>

<!--[if IE 8]> <html lang="en" class="ie8"> <![endif]-->
<!--[if IE 9]> <html lang="en" class="ie9"> <![endif]-->
<!--[if !IE]><!-->
<html lang="en">
<!--<![endif]-->
<head>
<meta charset="utf-8"/>
<meta content="IE=edge" http-equiv="X-UA-Compatible"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<title>COVID Live - Coronavirus Statistics - Worldometer</title>
<meta content="Live statistics and coronavirus news tracking the number of confirmed cases, recovered patients,
tests, and death toll due to the COVID-19 coronavirus from Wuhan, China. Coronavirus counter with new cases, dea
ths, and number of tests per 1 Million population. Historical data and info. Daily charts, graphs, news and upda
tes" name="description"/>
<!-- Favicon -->
```

In [9]:

```
# since only one table so we use just find method, if there are more than 1
# to find than we use find_all function
table=soup.find('table', id='main_table_countries_today')
```

In [10]:

```
table=table.find('tbody')
```

In [11]:

```
table
```

Out[11]:

```
<tbody>
<tr class="total_row_world row_continent" data-continent="North America" style="display: none">
<td></td>
<td style="text-align:left;">
<nobr>North America</nobr>
</td>
<td>120,049,996</td>
<td></td>
<td>1,569,729</td>
<td></td>
<td>115,176,046</td>
<td></td>
<td>3,304,221</td>
<td>8,984</td>
<td></td>
<td></td>
<td></td>
<td></td>
```

In [12]:

```
rows=table.find_all('tr',style="")
rows
```

Out[12]:

```
[<tr class="total_row_world">
 <td></td>
 <td style="text-align:left;">World</td>
 <td>654,206,689</td>
 <td>+95,219</td>
 <td>6,660,216</td>
 <td>+49</td>
 <td>629,628,626</td>
 <td>+115,772</td>
 <td>17,917,847</td>
 <td>37,560</td>
 <td>83,929</td>
 <td>854.4</td>
 <td></td>
 <td></td>
 <td></td>
 <td data-continent="all" style="display:none">All</td>
 <!-- 1 Case every X -->
```

In [13]:

```
column_info=[]
for i in rows:
    cols=i.find_all('td')
    #print cols
    country_info=[c.text for c in cols]
    #print(country_info)
    column_info.append(country_info)
print(column_info)
```

```
[['', 'World', '654,206,689', '+95,219', '6,660,216', '+49', '629,628,626', '+115,772', '17,917,847', '37,560',
'83,929', '854.4', '', '', '', 'All', '\n', '', '', '', '', ''], ['1', 'USA', '101,369,163', '', '1,109,983
', '', '98,608,503', '', '1,650,677', '3,801', '302,771', '3,315', '1,145,374,855', '3,421,018', '334,805,269 ',
'North America', '3', '302', '0', '', '', '4,930'], ['2', 'India', '44,676,246', '', '530,658
', '', '44,140,592', '', '4,996', '698', '31,761', '377', '908,416,385', '645,810', '1,406,631,776 ', 'Asia', '3
1', '2,651', '2', '', '', '4'], ['3', 'France', '38,515,721', '', '159,870                                   ', '',
'37,283,076', '', '1,072,775', '869', '587,268', '2,438', '271,490,188', '4,139,547', '65,584,518 ', 'Europe',
'2', '410', '0', '', '', '16,357'], ['4', 'Germany', '36,812,671', '', '159,177
', '', '36,135,500', '+37,700', '517,994', '1,406', '438,854', '1,898', '122,332,384', '1,458,359', '83,883,596
', 'Europe', '2', '527', '1', '', '', '6,175'], ['5', 'Brazil', '35,659,520', '', '691,021
', '', '34,436,602', '', '531,897', '8,318', '165,586', '3,209', '63,776,166', '296,146', '215,353,593 ', 'South
America', '6', '312', '3', '', '', '2,470'], ['6', 'S. Korea', '27,841,001', '+86,852', '31,128
', '+29', '26,806,741', '+53,668', '1,003,132', '460', '542,393', '606', '15,804,065', '307,892', '51,329,899 ',
'Asia', '2', '1,649', '3', '1,692', '0.6', '19,543'], ['7', 'Japan', '26,132,159', '', '51,829
', '', '20,848,373', '', '5,231,957', '409', '208,084', '413', '83,692,489', '666,422', '125,584,838 ', 'Asia',
'5', '2,423', '2', '', '', '41,661'], ['8', 'Italy', '24,709,404', '', '182,419
', '', '24,003,910', '', '523,075', '335', '410,028', '3,027', '259,639,054', '4,308,449', '60,262,770 ', 'Europ
e', '2', '330', '0', '', '', '8,680'], ['9', 'UK', '24,053,576', '', '197,723                                    ',
'', '23,780,918', '+3,513', '74,935', '146', '351,158', '2,887', '522,526,476', '7,628,357', '68,497,907 ', 'Eur
```

In [16]:

```
number=[]
country=[]
total_cases=[]
new_case=[]
total_death=[]

for i in column_info:
    number.append(i[0])
    country.append(i[1])
    total_cases.append(i[2])
    new_case.append(i[3])
    total_death.append(i[4])
```

In [17]:

```
data={'Number':number,'Countries':country,'Total cases':total_cases,
      'New cases':new_case,'Total deaths':total_death}
```

In [18]:

```
covid=pd.DataFrame(data)
covid
```

Out[18]:

	Number	Countries	Total cases	New cases	Total deaths
0		World	654,206,689	+95,219	6,660,216
1	1	USA	101,369,163		1,109,983
2	2	India	44,676,246		530,658
3	3	France	38,515,721		159,870
4	4	Germany	36,812,671		159,177
...	...	...	...	...	...
218	222	Montserrat	1,403		8
219	223	Macao	1,040		6
220	224	Wallis and Futuna	761		7
221	226	Niue	191		
222	230	China	367,627	+2,315	5,235

223 rows × 5 columns

In [19]:

```
covid.to_csv('sneha_covid_data.csv',index=False)
```

In [ ]: