

PYTHON PROJECT on WEB SCRAPPING

Web scraping is a technique to fetch data from websites. While surfing on the web, many websites don't allow the user to save data for personal use. One way is to manually copy-paste the data, which both tedious and time-consuming. Web Scraping is the automation of the data extraction process from websites. This event is done with the help of web scraping software known as web scrapers. They automatically load and extract data from the websites based on user requirements. These can be custom built to work for one site or can be configured to work with any website.

What is Web Scraping?

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data.

Web scraping requires two parts, namely the crawler and the scraper. The crawler is an artificial intelligence algorithm that browses the web to search for the particular data required by following the links across the internet. The scraper, on the other hand, is a specific tool created to extract data from the website. The design of the scraper can vary greatly according to the complexity and scope of the project so that it can quickly and accurately extract the data.

How web Scrapers Work?

Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it's best if you specify the data you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.

PYTHON PROJECT on WEB SCRAPING

So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and JavaScript elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.

Different Types of web Scrapers:

Web Scrapers can be divided on the basis of many different criteria, including Self-built or Pre-built Web Scrapers, Browser extension or Software Web Scrapers, and Cloud or Local Web Scrapers.

You can have **Self-built Web Scrapers** but that requires advanced knowledge of programming. And if you want more features in your Web Scraper, then you need even more knowledge. On the other hand, pre-built **Web Scrapers** are previously created scrapers that you can download and run easily. These also have more advanced options that you can customize.

Browser extensions Web Scrapers are extensions that can be added to your browser. These are easy to run as they are integrated with your browser, but at the same time, they are also limited because of this. Any advanced features that are outside the scope of your browser are impossible to run on Browser extension Web Scrapers. But **Software Web Scrapers** don't have these limitations as they can be downloaded and installed on your computer. These are more complex than Browser web scrapers, but they also have advanced features that are not limited by the scope of your browser.

Cloud Web Scrapers run on the cloud, which is an off-site server mostly provided by the company that you buy the scraper from. These allow your computer to focus on other tasks as the computer resources are not required to scrape data from websites. **Local Web Scrapers**, on the other hand, run on your computer using local resources. So, if the Web scrapers require more CPU or RAM, then your computer will become slow and not be able to perform other tasks.

PYTHON PROJECT on WEB SCRAPPING

Why is Python a popular programming language for Web Scraping?

It is the most popular language for web scraping as it can handle most of the processes easily. It also has a variety of libraries that were created specifically for Web Scraping. **Scrapy** is a very popular open-source web crawling framework that is written in Python. It is ideal for web scraping as well as extracting data using APIs. **Beautiful soup** is another Python library that is highly suitable for Web Scraping. It creates a parse tree that can be used to extract data from HTML on a website. Beautiful soup also has multiple features for navigation, searching, and modifying these parse trees.

What is Web Scraping used for?

Web Scraping has multiple applications across various industries. It finds many uses both at a professional and personal level. Having different needs at different levels, some popular uses of web scraping are.

1. Price Monitoring

Web Scraping can be used by companies to scrap the product data for their products and competing products as well to see how it impacts their pricing strategies. Companies can use this data to fix the optimal pricing for their products so that they can obtain maximum revenue.

2. Market Research

Web scraping can be used for market research by companies. High-quality web scraped data obtained in large volumes can be very helpful for companies in analysing consumer trends and understanding which direction the company should move in the future.

3. News Monitoring

PYTHON PROJECT on WEB SCRAPPING

Web scraping news sites can provide detailed reports on the current news to a company. This is even more essential for companies that are frequently in the news or that depend on daily news for their day-to-day functioning. After all, news reports can make or break a company in a single day!

4. Sentiment Analysis

If companies want to understand the general sentiment for their products among their consumers, then Sentiment Analysis is a must. Companies can use web scraping to collect data from social media websites such as Facebook and Twitter as to what the general sentiment about their products is. This will help them in creating products that people desire and moving ahead of their competition.

5. Email Marketing

Companies can also use Web scraping for email marketing. They can collect Email IDs from various sites using web scraping and then send bulk promotional and marketing Emails to all the people owning these Email ID's.

6. Brand Monitoring and Competition Analysis

Web Scraping is used to get customer feedback regarding a particular service or product so as to understand how a customer feels regarding that particular thing. It is also used to extract competitor data in a structural, usable format.

7. Machine Learning

Machine Learning is a process of Artificial Intelligence in which the machine is allowed to learn and improve with its experience rather than being explicitly programmed. For that, a large amount of data is required from millions of sites which is extracted through web scraping software.

8. Financial Data Analysis

Web Scraping is used to keep a record of the stock market in a usable format and hence employ the same for insights.

PYTHON PROJECT on WEB SCRAPING

9. Social Media Analysis

It is used to extract data from social media sites to gauge customer trends, and how they react to the campaign.

10. SEO monitoring

Search Engine Optimization is the optimization of the visibility and ranking of a website among different search engines like Google, Yahoo, Bing, etc. Web scraping is used to understand how the ranking of the content over time.

And there are so many other reasons to use Web Scraping

Techniques of Web Scraping: There are two ways of extracting data from websites, the Manual extraction technique, and the automated extraction technique.

Manual Extraction Techniques: Manually copy-pasting the site content comes under this technique. Though tedious, time taking and repetitive it is an effective way to scrap data from the sites having good anti-scraping measures like bot detection.

Automated Extraction Techniques: Web scraping software is used to automatically extract data from sites based on user requirement.

HTML Parsing: Parsing means to make something understandable to be analysing it part by part. To wit, it means to convert the information in one form to another form that is easy to that is easier to work on with. HTML parsing means taking in the code and extracting relevant information from it based on the user requirement. Mainly executed using JavaScript, the target as the name suggests are HTML pages.

DOM Parsing: The Document Object Model is the official recommendation of the World Wide Web Consortium. It defines an interface that enables a user to modify and update the style, structure, and content of the XML document.

Web Scraping Software: Nowadays, many web scraping tools are available or are custom build on users need to extract required desiring information from millions of websites.

PYTHON PROJECT on WEB SCRAPING

Challenges to Web Scraping: Besides the challenge of the legality of web scraping, there are also other problems that pose a challenge to web scraping.

Data Warehousing: Data extraction at a scale will generate a large amount of information to be stored. If the data warehousing infrastructure is not properly built then the searching, storing and exporting of this data will become a cumbersome task. Hence, for large-scale data extraction, there needs to be a perfect data warehousing system without any flaws and faults.

Website Structure Changes: Every website periodically updates its user interface to improve its attractiveness and experience. This requires various structural changes too. Since the web scrapers are set up according to the code elements of the website at that time, they require changes too. So, they require changes weekly too to target the correct website for data scraping as incomplete information regarding the website structure will lead to improper scraping of data.

Anti-Scraping Technologies: Some websites use anti-scraping technologies that thwart away any scraping attempt. They apply a dynamic coding algorithm to prevent any bot intervention and use the IP blocking mechanism. It requires a lot of time and money to work around such anti-scraping technologies.

Quality of Data Extracted: Records that do not meet the quality of information required will affect the overall integrity of the data. Making sure that the Data Scraped meets the quality guidelines is a difficult task as it needs to be done in real-time.

Future of Data Scraping: As there are some challenges and opportunities for data scraping, it can be fairly deemed that the unintended data-scraping practitioners are prone to create a moral hazard where they target the companies and retrieve their data. However, since we are on the verge of data transformation, data-scraping in combination with big data can provide the company's market intelligence and help them identify critical trends and patterns and identify the best opportunities and solutions. Hence, it won't be wrong to say that Data scraping can be upgraded to the better soon.

PYTHON PROJECT on WEB SCRAPPING

Web scrapping using python:

Requests Module:

Requests library is used for making HTTP requests to a specific URL and returns the response. Python requests provide inbuilt functionalities for managing both the request and response.

```
pip install requests
```

Making a Request:

Python requests module has several built-in methods to make HTTP requests to specified URI using GET, POST, PUT, PATCH, or HEAD requests. A HTTP request is meant to either retrieve data from a specified URI or to push data to a server. It works as a request-response protocol between a client and a server. Here we will be using the GET request.

GET method is used to retrieve information from the given server using a given URI. The GET method sends the encoded user information appended to the page request.

BeautifulSoup Library:

BeautifulSoup is used extract information from the HTML and XML files. It provides a parse tree and the functions to navigate, search or modify this parse tree.

Beautiful Soup is a Python library used to pull the data out of HTML and XML files for web scraping purposes. It produces a parse tree from page source code that can be utilized to drag data hierarchically and more legibly.

It was first presented by Leonard Richardson, who is still donating to this project, and this project is also supported by Tide lift (a paid subscription tool for open-source supervision).

Beautiful soup3 was officially released in May 2006, Latest version released by Beautiful Soup is 4.9.2, and it supports Python 3 and Python 2.4 as well.

PYTHON PROJECT on WEB SCRAPPING

Installation:

To install BeautifulSoup on Windows, Linux, or any operating system, one would need pip package. To check how to install pip on your operating system, check out – PIP Installation – Windows || Linux. Now run the below command in the terminal.

```
pip install beautifulsoup4
```

Here

[COVID - Coronavirus Statistics - Worldometer](https://www.worldometers.info/coronavirus/)

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

data is scraped from above website in python- Jupyter notebook with framework BeautifulSoup and the same data is fetched in SQL - XAMPP with SQL to PYTHON connectivity.

PYTHON PROJECT on WEB SCRAPPING

```
In [ ]: #import basic libraries
```

```
In [1]: import requests
```

```
In [2]: import pandas as pd
```

```
In [3]: import numpy as np
```

```
In [4]: !pip install bs4
```

```
Requirement already satisfied: bs4 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (0.0.1)  
Requirement already satisfied: beautifulsoup4 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from bs4) (4.11.1)  
Requirement already satisfied: soupsieve>1.2 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from beautifulsoup4->bs4) (2.3.2.post1)
```

```
In [5]: from bs4 import BeautifulSoup
```

```
In [6]: !pip install requests
```

```
Requirement already satisfied: requests in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (2.28.1)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from requests) (2022.12.7)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from requests) (3.4)  
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from requests) (2.0.4)  
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from requests) (1.26.14)
```

```
In [8]: html_page=requests.get("https://www.worldometers.info/coronavirus/")  
html_page
```

```
Out[8]: <Response [200]>
```

PYTHON PROJECT on WEB SCRAPPING

```
In [9]: soup=BeautifulSoup(html_page.content)
        soup
```

```
Out[9]: <!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 - Coronavirus Statistics - Worldometer</title>
<meta content="Daily and weekly updated statistics tracking the number of COVID-19 cases, recovered, and deaths. Historical d
ata with cumulative charts, graphs, and updates." name="description"/>
<!-- Favicon -->
<link href="/favicon/favicon.ico" rel="shortcut icon" type="image/x-icon"/>
<link href="/favicon/apple-icon-57x57.png" rel="apple-touch-icon" sizes="57x57"/>
<link href="/favicon/apple-icon-60x60.png" rel="apple-touch-icon" sizes="60x60"/>
<link href="/favicon/apple-icon-72x72.png" rel="apple-touch-icon" sizes="72x72"/>
<link href="/favicon/apple-icon-76x76.png" rel="apple-touch-icon" sizes="76x76"/>
<link href="/favicon/apple-icon-114x114.png" rel="apple-touch-icon" sizes="114x114"/>
<link href="/favicon/apple-icon-120x120.png" rel="apple-touch-icon" sizes="120x120"/>
<link href="/favicon/apple-icon-144x144.png" rel="apple-touch-icon" sizes="144x144"/>
<link href="/favicon/apple-icon-152x152.png" rel="apple-touch-icon" sizes="152x152"/>
<link href="/favicon/apple-icon-160x160.png" rel="apple-touch-icon" sizes="160x160"/>
<link href="/favicon/apple-icon-167x167.png" rel="apple-touch-icon" sizes="167x167"/>
<link href="/favicon/apple-icon-180x180.png" rel="apple-touch-icon" sizes="180x180"/>
<link href="/favicon/android-icon-36x36.png" rel="android-icon" sizes="36x36"/>
<link href="/favicon/android-icon-48x48.png" rel="android-icon" sizes="48x48"/>
<link href="/favicon/android-icon-72x72.png" rel="android-icon" sizes="72x72"/>
<link href="/favicon/android-icon-96x96.png" rel="android-icon" sizes="96x96"/>
<link href="/favicon/android-icon-144x144.png" rel="android-icon" sizes="144x144"/>
<link href="/favicon/android-icon-192x192.png" rel="android-icon" sizes="192x192"/>
<link href="/favicon/android-icon-512x512.png" rel="android-icon" sizes="512x512"/>
<link href="/favicon/windows-icon-32x32.png" rel="windows-icon" sizes="32x32"/>
<link href="/favicon/windows-icon-48x48.png" rel="windows-icon" sizes="48x48"/>
<link href="/favicon/windows-icon-64x64.png" rel="windows-icon" sizes="64x64"/>
<link href="/favicon/windows-icon-128x128.png" rel="windows-icon" sizes="128x128"/>
<link href="/favicon/windows-icon-256x256.png" rel="windows-icon" sizes="256x256"/>
<link href="/favicon/windows-icon-512x512.png" rel="windows-icon" sizes="512x512"/>
<link href="/favicon/windows-icon-1024x1024.png" rel="windows-icon" sizes="1024x1024"/>
<link href="/favicon/windows-icon-2048x2048.png" rel="windows-icon" sizes="2048x2048"/>
<link href="/favicon/windows-icon-4096x4096.png" rel="windows-icon" sizes="4096x4096"/>
<link href="/favicon/windows-icon-8192x8192.png" rel="windows-icon" sizes="8192x8192"/>
<link href="/favicon/windows-icon-16384x16384.png" rel="windows-icon" sizes="16384x16384"/>
<link href="/favicon/windows-icon-32768x32768.png" rel="windows-icon" sizes="32768x32768"/>
<link href="/favicon/windows-icon-65536x65536.png" rel="windows-icon" sizes="65536x65536"/>
<link href="/favicon/windows-icon-131072x131072.png" rel="windows-icon" sizes="131072x131072"/>
<link href="/favicon/windows-icon-262144x262144.png" rel="windows-icon" sizes="262144x262144"/>
<link href="/favicon/windows-icon-524288x524288.png" rel="windows-icon" sizes="524288x524288"/>
<link href="/favicon/windows-icon-1048576x1048576.png" rel="windows-icon" sizes="1048576x1048576"/>
<link href="/favicon/windows-icon-2097152x2097152.png" rel="windows-icon" sizes="2097152x2097152"/>
<link href="/favicon/windows-icon-4194304x4194304.png" rel="windows-icon" sizes="4194304x4194304"/>
<link href="/favicon/windows-icon-8388608x8388608.png" rel="windows-icon" sizes="8388608x8388608"/>
<link href="/favicon/windows-icon-16777216x16777216.png" rel="windows-icon" sizes="16777216x16777216"/>
<link href="/favicon/windows-icon-33554432x33554432.png" rel="windows-icon" sizes="33554432x33554432"/>
<link href="/favicon/windows-icon-67108864x67108864.png" rel="windows-icon" sizes="67108864x67108864"/>
<link href="/favicon/windows-icon-134217728x134217728.png" rel="windows-icon" sizes="134217728x134217728"/>
<link href="/favicon/windows-icon-268435456x268435456.png" rel="windows-icon" sizes="268435456x268435456"/>
<link href="/favicon/windows-icon-536870912x536870912.png" rel="windows-icon" sizes="536870912x536870912"/>
<link href="/favicon/windows-icon-1073741824x1073741824.png" rel="windows-icon" sizes="1073741824x1073741824"/>
<link href="/favicon/windows-icon-2147483648x2147483648.png" rel="windows-icon" sizes="2147483648x2147483648"/>
<link href="/favicon/windows-icon-4294967296x4294967296.png" rel="windows-icon" sizes="4294967296x4294967296"/>
<link href="/favicon/windows-icon-8589934592x8589934592.png" rel="windows-icon" sizes="8589934592x8589934592"/>
<link href="/favicon/windows-icon-17179869184x17179869184.png" rel="windows-icon" sizes="17179869184x17179869184"/>
<link href="/favicon/windows-icon-34359738368x34359738368.png" rel="windows-icon" sizes="34359738368x34359738368"/>
<link href="/favicon/windows-icon-68719476736x68719476736.png" rel="windows-icon" sizes="68719476736x68719476736"/>
<link href="/favicon/windows-icon-137438953472x137438953472.png" rel="windows-icon" sizes="137438953472x137438953472"/>
<link href="/favicon/windows-icon-274877906944x274877906944.png" rel="windows-icon" sizes="274877906944x274877906944"/>
<link href="/favicon/windows-icon-549755813888x549755813888.png" rel="windows-icon" sizes="549755813888x549755813888"/>
<link href="/favicon/windows-icon-1099511627776x1099511627776.png" rel="windows-icon" sizes="1099511627776x1099511627776"/>
<link href="/favicon/windows-icon-2199023255552x2199023255552.png" rel="windows-icon" sizes="2199023255552x2199023255552"/>
<link href="/favicon/windows-icon-4398046511104x4398046511104.png" rel="windows-icon" sizes="4398046511104x4398046511104"/>
<link href="/favicon/windows-icon-8796093022208x8796093022208.png" rel="windows-icon" sizes="8796093022208x8796093022208"/>
<link href="/favicon/windows-icon-17592186044416x17592186044416.png" rel="windows-icon" sizes="17592186044416x17592186044416"/>
<link href="/favicon/windows-icon-35184372088832x35184372088832.png" rel="windows-icon" sizes="35184372088832x35184372088832"/>
<link href="/favicon/windows-icon-70368744177664x70368744177664.png" rel="windows-icon" sizes="70368744177664x70368744177664"/>
<link href="/favicon/windows-icon-140737488355328x140737488355328.png" rel="windows-icon" sizes="140737488355328x140737488355328"/>
<link href="/favicon/windows-icon-281474976710656x281474976710656.png" rel="windows-icon" sizes="281474976710656x281474976710656"/>
<link href="/favicon/windows-icon-562949953421312x562949953421312.png" rel="windows-icon" sizes="562949953421312x562949953421312"/>
<link href="/favicon/windows-icon-1125899906842624x1125899906842624.png" rel="windows-icon" sizes="1125899906842624x1125899906842624"/>
<link href="/favicon/windows-icon-2251799813685248x2251799813685248.png" rel="windows-icon" sizes="2251799813685248x2251799813685248"/>
<link href="/favicon/windows-icon-4503599627370496x4503599627370496.png" rel="windows-icon" sizes="4503599627370496x4503599627370496"/>
<link href="/favicon/windows-icon-9007199254740992x9007199254740992.png" rel="windows-icon" sizes="9007199254740992x9007199254740992"/>
<link href="/favicon/windows-icon-18014398509481984x18014398509481984.png" rel="windows-icon" sizes="18014398509481984x18014398509481984"/>
<link href="/favicon/windows-icon-36028797018963968x36028797018963968.png" rel="windows-icon" sizes="36028797018963968x36028797018963968"/>
<link href="/favicon/windows-icon-72057594037927936x72057594037927936.png" rel="windows-icon" sizes="72057594037927936x72057594037927936"/>
<link href="/favicon/windows-icon-144115188075855872x144115188075855872.png" rel="windows-icon" sizes="144115188075855872x144115188075855872"/>
<link href="/favicon/windows-icon-288230376151711744x288230376151711744.png" rel="windows-icon" sizes="288230376151711744x288230376151711744"/>
<link href="/favicon/windows-icon-576460752303423488x576460752303423488.png" rel="windows-icon" sizes="576460752303423488x576460752303423488"/>
<link href="/favicon/windows-icon-1152921504606846976x1152921504606846976.png" rel="windows-icon" sizes="1152921504606846976x1152921504606846976"/>
<link href="/favicon/windows-icon-2305843009213693952x2305843009213693952.png" rel="windows-icon" sizes="2305843009213693952x2305843009213693952"/>
<link href="/favicon/windows-icon-4611686018427387904x4611686018427387904.png" rel="windows-icon" sizes="4611686018427387904x4611686018427387904"/>
<link href="/favicon/windows-icon-9223372036854775808x9223372036854775808.png" rel="windows-icon" sizes="9223372036854775808x9223372036854775808"/>
<link href="/favicon/windows-icon-18446744073709551616x18446744073709551616.png" rel="windows-icon" sizes="18446744073709551616x18446744073709551616"/>
<link href="/favicon/windows-icon-36893488147419103232x36893488147419103232.png" rel="windows-icon" sizes="36893488147419103232x36893488147419103232"/>
<link href="/favicon/windows-icon-73786976294838206464x73786976294838206464.png" rel="windows-icon" sizes="73786976294838206464x73786976294838206464"/>
<link href="/favicon/windows-icon-147573952589676412928x147573952589676412928.png" rel="windows-icon" sizes="147573952589676412928x147573952589676412928"/>
<link href="/favicon/windows-icon-295147905179352825856x295147905179352825856.png" rel="windows-icon" sizes="295147905179352825856x295147905179352825856"/>
<link href="/favicon/windows-icon-5902958103587056517056x5902958103587056517056.png" rel="windows-icon" sizes="5902958103587056517056x5902958103587056517056"/>
<link href="/favicon/windows-icon-11805916207174113034112x11805916207174113034112.png" rel="windows-icon" sizes="11805916207174113034112x11805916207174113034112"/>
<link href="/favicon/windows-icon-23611832414348226068224x23611832414348226068224.png" rel="windows-icon" sizes="23611832414348226068224x23611832414348226068224"/>
<link href="/favicon/windows-icon-47223664828696452136448x47223664828696452136448.png" rel="windows-icon" sizes="47223664828696452136448x47223664828696452136448"/>
<link href="/favicon/windows-icon-94447329657392904272896x94447329657392904272896.png" rel="windows-icon" sizes="94447329657392904272896x94447329657392904272896"/>
<link href="/favicon/windows-icon-188894659314785808545792x188894659314785808545792.png" rel="windows-icon" sizes="188894659314785808545792x188894659314785808545792"/>
<link href="/favicon/windows-icon-377789318629571617091584x377789318629571617091584.png" rel="windows-icon" sizes="377789318629571617091584x377789318629571617091584"/>
<link href="/favicon/windows-icon-755578637259143234183168x755578637259143234183168.png" rel="windows-icon" sizes="755578637259143234183168x755578637259143234183168"/>
<link href="/favicon/windows-icon-1511157274518286468366336x1511157274518286468366336.png" rel="windows-icon" sizes="1511157274518286468366336x1511157274518286468366336"/>
<link href="/favicon/windows-icon-3022314549036572936732672x3022314549036572936732672.png" rel="windows-icon" sizes="3022314549036572936732672x3022314549036572936732672"/>
<link href="/favicon/windows-icon-6044629098073145873465344x6044629098073145873465344.png" rel="windows-icon" sizes="6044629098073145873465344x6044629098073145873465344"/>
<link href="/favicon/windows-icon-12089258196146291746930688x12089258196146291746930688.png" rel="windows-icon" sizes="12089258196146291746930688x12089258196146291746930688"/>
<link href="/favicon/windows-icon-24178516392292583493861376x24178516392292583493861376.png" rel="windows-icon" sizes="24178516392292583493861376x24178516392292583493861376"/>
<link href="/favicon/windows-icon-48357032784585166987722752x48357032784585166987722752.png" rel="windows-icon" sizes="48357032784585166987722752x48357032784585166987722752"/>
<link href="/favicon/windows-icon-96714065569170333975445504x96714065569170333975445504.png" rel="windows-icon" sizes="96714065569170333975445504x96714065569170333975445504"/>
<link href="/favicon/windows-icon-193428131138340667950891008x193428131138340667950891008.png" rel="windows-icon" sizes="193428131138340667950891008x193428131138340667950891008"/>
<link href="/favicon/windows-icon-386856262276681335901782016x386856262276681335901782016.png" rel="windows-icon" sizes="386856262276681335901782016x386856262276681335901782016"/>
<link href="/favicon/windows-icon-773712524553362671803564032x773712524553362671803564032.png" rel="windows-icon" sizes="773712524553362671803564032x773712524553362671803564032"/>
<link href="/favicon/windows-icon-1547425049106725343607128064x1547425049106725343607128064.png" rel="windows-icon" sizes="1547425049106725343607128064x1547425049106725343607128064"/>
<link href="/favicon/windows-icon-3094850098213450687214256128x3094850098213450687214256128.png" rel="windows-icon" sizes="3094850098213450687214256128x3094850098213450687214256128"/>
<link href="/favicon/windows-icon-6189700196426901374428512256x6189700196426901374428512256.png" rel="windows-icon" sizes="6189700196426901374428512256x6189700196426901374428512256"/>
<link href="/favicon/windows-icon-12379400392853802748857024512x12379400392853802748857024512.png" rel="windows-icon" sizes="12379400392853802748857024512x12379400392853802748857024512"/>
<link href="/favicon/windows-icon-24758800785707605497714049024x24758800785707605497714049024.png" rel="windows-icon" sizes="24758800785707605497714049024x24758800785707605497714049024"/>
<link href="/favicon/windows-icon-49517601571415210995428098048x49517601571415210995428098048.png" rel="windows-icon" sizes="49517601571415210995428098048x49517601571415210995428098048"/>
<link href="/favicon/windows-icon-99035203142830421990856196096x99035203142830421990856196096.png" rel="windows-icon" sizes="99035203142830421990856196096x99035203142830421990856196096"/>
<link href="/favicon/windows-icon-198070406285660843981712392192x198070406285660843981712392192.png" rel="windows-icon" sizes="198070406285660843981712392192x198070406285660843981712392192"/>
<link href="/favicon/windows-icon-396140812571321687963424784384x396140812571321687963424784384.png" rel="windows-icon" sizes="396140812571321687963424784384x396140812571321687963424784384"/>
<link href="/favicon/windows-icon-792281625142643375926849568768x792281625142643375926849568768.png" rel="windows-icon" sizes="792281625142643375926849568768x792281625142643375926849568768"/>
<link href="/favicon/windows-icon-1584563250285286751853699137536x1584563250285286751853699137536.png" rel="windows-icon" sizes="1584563250285286751853699137536x1584563250285286751853699137536"/>
<link href="/favicon/windows-icon-3169126500570573503707398275072x3169126500570573503707398275072.png" rel="windows-icon" sizes="3169126500570573503707398275072x3169126500570573503707398275072"/>
<link href="/favicon/windows-icon-6338253001141147007414796550144x6338253001141147007414796550144.png" rel="windows-icon" sizes="6338253001141147007414796550144x6338253001141147007414796550144"/>
<link href="/favicon/windows-icon-12676506002282294014829593100288x12676506002282294014829593100288.png" rel="windows-icon" sizes="12676506002282294014829593100288x12676506002282294014829593100288"/>
<link href="/favicon/windows-icon-25353012004564588029659186200576x25353012004564588029659186200576.png" rel="windows-icon" sizes="25353012004564588029659186200576x25353012004564588029659186200576"/>
<link href="/favicon/windows-icon-50706024009129176059318372401152x50706024009129176059318372401152.png" rel="windows-icon" sizes="50706024009129176059318372401152x50706024009129176059318372401152"/>
<link href="/favicon/windows-icon-101412048018258352118636744802304x101412048018258352118636744802304.png" rel="windows-icon" sizes="101412048018258352118636744802304x101412048018258352118636744802304"/>
<link href="/favicon/windows-icon-202824096036516704237273489604608x202824096036516704237273489604608.png" rel="windows-icon" sizes="202824096036516704237273489604608x202824096036516704237273489604608"/>
<link href="/favicon/windows-icon-405648192073033408474546979209216x405648192073033408474546979209216.png" rel="windows-icon" sizes="405648192073033408474546979209216x405648192073033408474546979209216"/>
<link href="/favicon/windows-icon-811296384146066816949093958418432x811296384146066816949093958418432.png" rel="windows-icon" sizes="811296384146066816949093958418432x811296384146066816949093958418432"/>
<link href="/favicon/windows-icon-1622592768292133633898187916836864x1622592768292133633898187916836864.png" rel="windows-icon" sizes="1622592768292133633898187916836864x1622592768292133633898187916836864"/>
<link href="/favicon/windows-icon-3245185536584267267796375833673728x3245185536584267267796375833673728.png" rel="windows-icon" sizes="3245185536584267267796375833673728x3245185536584267267796375833673728"/>
<link href="/favicon/windows-icon-6490371073168534535592751667347456x6490371073168534535592751667347456.png" rel="windows-icon" sizes="6490371073168534535592751667347456x6490371073168534535592751667347456"/>
<link href="/favicon/windows-icon-12980742146337069071185503334694912x12980742146337069071185503334694912.png" rel="windows-icon" sizes="12980742146337069071185503334694912x12980742146337069071185503334694912"/>
<link href="/favicon/windows-icon-25961484292674138142371006669389824x25961484292674138142371006669389824.png" rel="windows-icon" sizes="25961484292674138142371006669389824x25961484292674138142371006669389824"/>
<link href="/favicon/windows-icon-51922968585348276284742013338779648x51922968585348276284742013338779648.png" rel="windows-icon" sizes="51922968585348276284742013338779648x51922968585348276284742013338779648"/>
<link href="/favicon/windows-icon-103845937170696552569484026677559296x103845937170696552569484026677559296.png" rel="windows-icon" sizes="103845937170696552569484026677559296x103845937170696552569484026677559296"/>
<link href="/favicon/windows-icon-207691874341393105138968053355118592x207691874341393105138968053355118592.png" rel="windows-icon" sizes="207691874341393105138968053355118592x207691874341393105138968053355118592"/>
<link href="/favicon/windows-icon-415383748682786210277936106710237184x415383748682786210277936106710237184.png" rel="windows-icon" sizes="415383748682786210277936106710237184x415383748682786210277936106710237184"/>
<link href="/favicon/windows-icon-830767497365572420555872213420474368x830767497365572420555872213420474368.png" rel="windows-icon" sizes="83076749
```

PYTHON PROJECT on WEB SCRAPPING

```
In [11]: country_info = []
         for row in rows:

             columns = row.find_all("td")
             for i in columns:
                 country_info.append(i.text.strip())
             #country_info = [column.text.strip() for column in columns]
         #global_info.append(country_info)

         print("\r")
         print("Number of rows in table: {}".format(len(country_info)))
         print("\r")
```

Number of rows in table: 4884

```
In [62]: country_info
```

```
787 ,
'657,635,742',
'+50,838',
'20,433,359',
'39,703',
'87,867',
'877.2',
'',
'',
'',
'',
'',
'All',
'',
'',
'',
'',
'',
'',
'',
'',
'1',
'USA',
'106,363,426',
...
```

PYTHON PROJECT on WEB SCRAPPING

```
In [12]: new_list=country_info[22:]  
new_list
```

```
Out[12]: ['1',  
          'USA',  
          '106,363,748',  
          '',  
          '1,156,898',  
          '',  
          '104,179,022',  
          '',  
          '1,027,828',  
          '1,810',  
          '317,688',  
          '3,455',  
          '1,175,206,859',  
          '3,510,121',  
          '334,805,269',  
          'North America',  
          '3',  
          '289',  
          '0',  
          '']
```

```
In [13]: srno=new_list[0::22]  
srno
```

```
Out[13]: ['1',  
          '2',  
          '3',  
          '4',  
          '5',  
          '6',  
          '7',  
          '8',  
          '9',  
          '10',  
          '11',  
          '12',  
          '13',  
          '14',  
          '15',  
          '17',  
          '18',  
          '19',  
          '20',  
          '21']
```

PYTHON PROJECT on WEB SCRAPPING

```
In [14]: country = new_list[1::22]
country
```

```
Out[14]: ['USA',
          'India',
          'France',
          'Germany',
          'Brazil',
          'Japan',
          'S. Korea',
          'Italy',
          'UK',
          'Russia',
          'Turkey',
          'Spain',
          'Vietnam',
          'Australia',
          'Taiwan',
          'Netherlands',
          'Iran',
          'Mexico',
          'Indonesia',
          'Belgium']
```

```
In [15]: totalcases = new_list[2::22]
totalcases
```

```
Out[15]: ['106,363,748',
          '44,756,616',
          '39,843,556',
          '38,368,891',
          '37,319,254',
          '33,516,848',
          '30,904,502',
          '25,715,384',
          '24,448,729',
          '22,706,980',
          '17,232,066',
          '13,798,747',
          '11,527,701',
          '11,352,930',
          '10,239,998',
          '8,610,372',
          '7,592,255',
          '7,553,423',
          '6,750,603',
          '6,527,810']
```

PYTHON PROJECT on WEB SCRAPPING

```
In [16]: newcases = new_list[3::22]
          newcases
```

[illegible]

PYTHON PROJECT on WEB SCRAPPING

```
In [18]: totaldeaths = new_list[4::22]  
totaldeaths
```

```
Out[18]: ['1,156,898',  
'530,965',  
'165,857',  
'171,411',  
'700,556',  
'74,081',  
'34,322',  
'189,262',  
'209,396',  
'397,534',  
'102,174',  
'120,426',  
'43,186',  
'19,933',  
'19,005',  
'22,992',  
'145,391',  
'333,594',  
'161,050',  
'119,423',  
'142,690',  
'22,183',  
'36,582',  
'26,450',  
'111,676',  
'64,497',  
'36,982',  
'12,415',  
'34,115',  
'33,938',  
'42,699',
```

PYTHON PROJECT on WEB SCRAPPING

```
In [19]: newdeaths = new_list[5:22]
          newdeaths
```

[illegible]

```
In [20]: totalrecovered = new_list[6::22]
totalrecovered
```

'4,692,636',
'4,588,462',
'4,554,681',
'4,271,787',
'4,378,012',
'4,007,510',
'3,912,506',
'3,291,922',
'3,169,205',
'2,870,130',
'2,675,429',
'2,492,181',
'2,439,497',
'2,149,583',
'2,259,509',
'2,146,693',
'1,998,448',
'1,843,912',
'1,776,548',
'1,731,007'

PYTHON PROJECT on WEB SCRAPPING

```
In [21]: newrecovered = new_list[7::22]
newrecovered
```

```
Out[21]: [' ',
           ' ',
           ' ',
           ' ',
           ' ',
           ' ',
           ' ',
           ' ',
           '+13,077',
           ' ',
           ' ',
           '+3,905',
           ' ',
           ' ',
           'N/A',
           ' ',
           ' ',
           ' ',
           ' ',
           '+8,418',
           '+832',
           ' ',
           '+1,811',
           ' ',
           ' ']
```

```
In [22]: activecases = new_list[8::22]
         activecases
```

```
Out[22]: ['1,027,828',
           '32,814',
           '140,277',
           '80,380',
           '369,537',
           '11,718,041',
           '176,445',
           '132,525',
           '7,659',
           '228,497',
           'N/A',
           '32,372',
           '869,462',
           '41,333',
           '8,656',
           '8,673',
           '104,374',
           '423,050',
           '6,419',
           '1,040,556']
```

PYTHON PROJECT on WEB SCRAPPING

```
In [23]: serious = new_list[9::22]
serious
```

```
Out[23]: ['1,810',
          'N/A',
          '869',
          'N/A',
          'N/A',
          '50',
          '132',
          '85',
          'N/A',
          'N/A',
          '',
          '231',
          '6',
          '45',
          '',
          '53',
          '407',
          'N/A',
          'N/A',
          '66']
```

```
In [24]: tcases1mp = new_list[10::22]
tcases1mp
```

```
'67,479',
'431,796',
'120,721',
'133,439',
'501,398',
'36,295',
'67,073',
'177,330',
'544,814',
'379,459',
'264,479',
'291,639',
'58,474',
'386,754',
'464,327',
'228,993',
'12,140',
'341,708',
'461,986',
'169,597',
```

PYTHON PROJECT on WEB SCRAPPING

```
In [25]: deaths1mp = new_list[11::22]
        deaths1mp
```

```
Out[25]: ['3,455',
          '377',
          '2,529',
          '2,043',
          '3,253',
          '590',
          '669',
          '3,141',
          '3,057',
          '2,726',
          '1,194',
          '2,578',
          '436',
          '765',
          '796',
          '1,336',
          '1,690',
          '2,536',
          '577',
          '3,141']
```

```
In [26]: ttests = new_list[12::22]
        ttests
```

```
Out[26]: ['1,175,206,859',
          '922,613,516',
          '271,490,188',
          '122,332,384',
          '63,776,166',
          '98,681,137',
          '15,804,065',
          '270,400,793',
          '522,526,476',
          '273,400,000',
          '162,743,369',
          '471,036,328',
          '85,826,548',
          '78,835,048',
          '30,742,304',
          '25,984,435',
          '55,353,767',
          '19,825,266',
          '114,158,919',
          '130,673,300']
```

PYTHON PROJECT on WEB SCRAPPING

```
In [27]: ttests1mp = new_list[13::22]
ttests1mp
```

```
10,082,298',
'867,342',
'3,024,116',
'1,286,903',
'1,509,718',
'643,478',
'150,691',
'408,975',
'1,022,088',
'717,327',
'23,302,116',
'9,909,078',
'4,541,442',
'754,855',
'2,552,737',
'2,059,477',
'4,436,346',
'3,166,786',
'246,450',
'5,321,309',
```

```
In [28]: population = new_list[14::22]
population
```

```
Out[28]: ['334,805,269',
'1,406,631,776',
'65,584,518',
'83,883,596',
'215,353,593',
'125,584,838',
'51,329,899',
'60,262,770',
'68,497,907',
'145,805,947',
'85,561,976',
'46,719,142',
'98,953,541',
'26,068,792',
'23,888,595',
'17,211,447',
'86,022,837',
'131,562,772',
'279,134,505',
'37,730,708',
```

PYTHON PROJECT on WEB SCRAPPING

```
In [29]: coronainfo=pd.DataFrame({})

coronainfo["SrNo"] = srno
coronainfo["Country Name"] = country
coronainfo["Total Cases"] = totalcases
coronainfo["New Cases"] = newcases
coronainfo["Total Deaths"] = totaldeaths
coronainfo["New Deaths"] = newdeaths
coronainfo["Total Recovered"] = totalrecovered
coronainfo["New Recovered"] = newrecovered
coronainfo["Active Cases"] = activecases
coronainfo["Serious/Critical"] = serious
coronainfo["Total cases/1M POP"] = tcases1mp
coronainfo["Deaths/1M POP"] = deaths1mp
coronainfo["Total Tests"] = ttests
coronainfo["Total Tests/1M POP"] = ttests1mp
coronainfo["Population"] = population

coronainfo
```

PYTHON PROJECT on WEB SCRAPPING

Out[29]:

	SrNo	Country Name	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious/Critical	Total cases/1M POP	Deaths/1M POP	Total Tests	Total Tests/1M POP
0	1	USA	106,363,748		1,156,898		104,179,022		1,027,828	1,810	317,688	3,455	1,175,206,859	3,510,121
1	2	India	44,756,616		530,965		44,192,837		32,814	N/A	31,818	377	922,613,516	655,903
2	3	France	39,843,556		165,857		39,537,422		140,277	869	607,515	2,529	271,490,188	4,139,547
3	4	Germany	38,368,891		171,411		38,117,100		80,380	N/A	457,406	2,043	122,332,384	1,458,359
4	5	Brazil	37,319,254		700,556		36,249,161		369,537	N/A	173,293	3,253	63,776,166	296,146
...
216	222	Saint Helena	2,166				2		2,164		354,211			
217	224	Montserrat	1,403		8		1,376		19		282,578	1,611	17,762	3,577,442
218	225	Niue	800				797		3		493,218			
219	230	Tokelau	5						5		3,628			
220	231	China	503,302		5,272		379,053		118,977	N/A	347	4	160,000,000	110,461

221 rows × 15 columns

PYTHON PROJECT on WEB SCRAPPING

```
In [30]: coronainfo.to_csv("coronadata")
```

jupyter

coronadata

2 hours ago

Logout

FileEditViewLanguage

Plain Text

1

,SrNo,Country Name>Total Cases,New Cases,Total Deaths,New Deaths,Total Recovered,New Recovered,Active Cases,Serious/Critical,Total cases/1M POP,Deaths/1M POP,Total Tests,Total Tests/1M POP,Population

2

0,1,USA,"106,418,595",,"1,157,462",,"104,260,123",,"1,001,010",,"1,722",,"317,852",,"3,457",,"1,175,622,556",,"3,511,362",,"334,805,269"

3

1,2,India,"44,776,002",,"531,016",,"44,204,771",,"40,215",,"N/A",,"31,832",,"378",,"923,053,282",,"656,215",,"1,406,631,776"

4

2,3,France,"39,867,463",,"165,916",,"39,569,363",,"132,184",,"869",,"607,879",,"2,530",,"271,490,188",,"4,139,547",,"65,584,518"

5

3,4,Germany,"38,377,656",,"171,748",,"38,143,900",,"+5,100",,"62,008",,"N/A",,"457,511",,"2,047",,"122,332,384",,"1,458,359",,"83,883,596"

6

4,5,Brazil,"37,358,092",,"700,811",,"36,249,161",,"408,120",,"N/A",,"173,473",,"3,254",,"63,776,166",,"296,146",,"215,353,593"

7

5,6,Japan,"33,547,551",,"74,164",,"21,725,273",,"11,748,114",,"63",,"267,131",,"591",,"98,881,178",,"787,366",,"125,584,838"

8

6,7,S. Korea,"30,956,026",,"+11,596",,"34,361",,"+5",,"30,734,900",,"+9,347",,"186,765",,"127",,"603,080",,"669",,"15,804,065",,"307,892",,"51,329,899"

9

7,8,Italy,"25,715,384",,"189,262",,"25,393,597",,"132,525",,"85",,"426,721",,"3,141",,"270,400,793",,"4,487,029",,"60,262,770"

10

8,10,Russia,"22,734,824",,"397,681",,"22,116,159",,"220,984",,"N/A",,"155,925",,"2,727",,"273,400,000",,"1,875,095",,"145,805,947"

11

9,11,Turkey,"17,232,066",,"102,174",,"N/A",,"N/A",,"N/A",,"201,399",,"1,194",,"162,743,369",,"1,902,052",,"85,561,976"

12

10,12,Spain,"13,798,747",,"120,426",,"13,645,949",,"32,372",,"231",,"295,355",,"2,578",,"471,036,328",,"10,082,298",,"46,719,142"

13

11,13,Vietnam,"11,528,303",,"43,186",,"10,615,180",,"869,937",,"9",,"116,502",,"436",,"85,826,548",,"867,342",,"98,953,541"

14

12,14,Australia,"11,352,930",,"19,933",,"11,291,664",,"41,333",,"45",,"435,499",,"765",,"78,835,048",,"3,024,116",,"26,068,792"

15

13,15,Taiwan,"10,239,998",,"19,005",,"10,220,950",,"+55,43",,"428,656",,"796",,"30,742,304",,"1,286,903",,"23,888,595"

16

14,17,Netherlands,"8,610,372",,"22,992",,"8,580,796",,"+555",,"6,584",,"53",,"500,270",,"1,336",,"25,984,435",,"1,509,718",,"17,211,447"

17

15,18,Iran,"7,597,982",,"145,571",,"7,342,490",,"109,921",,"407",,"88,325",,"1,692",,"55,353,767",,"643,478",,"86,022,837"

18

16,19,Mexico,"7,556,927",,"333,619",,"6,804,253",,"419,055",,"N/A",,"57,440",,"2,536",,"19,835,770",,"150,770",,"131,562,772"

19

17,20,Indonesia,"6,753,593",,"161,082",,"6,585,054",,"7,457",,"N/A",,"24,195",,"577",,"114,158,919",,"408,975",,"279,134,505"

20

18,21,Poland,"6,505,726",,"119,432",,"5,335,940",,"1,050,354",,"696",,"172,384",,"3,165",,"38,587,511",,"1,022,462",,"37,739,785"

21

19,22,Colombia,"6,363,544",,"142,698",,"6,186,501",,"34,345",,"342",,"123,533",,"2,770",,"36,951,507",,"717,327",,"51,512,762"

22

20,23,Austria,"6,048,674",,"22,217",,"6,012,291",,"14,166",,"55",,"667,130",,"2,450",,"211,273,524",,"23,302,116",,"9,066,710"

23

21,24,Greece,"5,965,643",,"36,582",,"5,921,203",,"+1,964",,"7,858",,"115",,"578,255",,"3,546",,"102,228,365",,"9,909,078",,"10,316,637"

24

22,25,Portugal,"5,577,825",,"26,480",,"5,545,402",,"5,943",,"61",,"550,050",,"2,611",,"46,065,483",,"4,542,692",,"10,140,570"

25

23,26,Ukraine,"5,484,936",,"111,789",,"5,334,190",,"38,957",,"126,989",,"2,588",,"32,603,805",,"754,855",,"43,192,122"

26

24,27,Chile,"5,275,580",,"64,497",,"5,206,417",,"4,666",,"64",,"274,053",,"3,350",,"49,180,601",,"2,554,811",,"19,250,195"

27

25,28,Malaysia,"5,056,911",,"36,994",,"5,006,634",,"13,283",,"10",,"152,403",,"1,115",,"68,342,991",,"2,059,698",,"33,181,072"

28

26,29,Israel,"4,817,269",,"12,416",,"4,798,473",,"6,380",,"21",,"516,542",,"1,331",,"41,373,364",,"4,436,346",,"9,326,000"

29

27,30,Belgium,"4,782,863",,"34,115",,"4,715,583",,"+2,583",,"33,165",,"43",,"409,903",,"2,924",,"36,950,938",,"3,166,786",,"11,668,278"

30

28,32,Thailand,"4,728,967",,"33,940",,"4,692,636",,"2,391",,"N/A",,"67,481",,"484",,"17,270,775",,"246,450",,"70,078,203"

31

29,33,Czechia,"4,636,934",,"42,713",,"4,589,926",,"4,295",,"28",,"431,874",,"3,978",,"57,139,828",,"5,321,876",,"10,736,784"

32

30,34,Canada,"4,634,277",,"52,121",,"4,559,665",,"+1,254",,"22,491",,"99",,"120,721",,"1,358",,"66,343,123",,"1,728,207",,"38,388,419"

33

31,35,Peru,"4,496,200",,"219,944",,"4,272,599",,"3,657",,"33",,"133,481",,"6,530",,"38,392,325",,"1,139,772",,"33,684,208"

34

32,36,Switzerland,"4,399,088",,"14,452",,"4,379,844",,"+1,832",,"4,792",,"39",,"501,398",,"1,647",,"23,833,472",,"2,716,487",,"8,773,637"

35

33,37,Philippines,"4,084,255",,"66,431",,"4,008,470",,"9,354",,"162",,"36,302",,"590",,"34,890,687",,"310,115",,"112,508,994"

36

34,38,South Africa,"4,075,512",,"102,595",,"3,912,506",,"60,411",,"192",,"67,080",,"1,689",,"26,795,090",,"441,027",,"60,756,135"

37

35,39,Romania,"3,380,891",,"67,961",,"3,299,960",,"12,970",,"132",,"177,649",,"3,571",,"27,131,523",,"1,425,624",,"19,031,335"

38

36,40,Denmark,"3,179,265",,"8,441",,"3,169,676",,"1,148",,"19",,"544,866",,"1,447",,"129,249,625",,"22,150,940",,"5,834,950"

39

37,41,Hong Kong,"2,886,027",,"13,488",,"2,870,274",,"2,265",,"28",,"379,526",,"1,774",,"76,127,725",,"10,011,143",,"7,604,299"

40

38,42,Sweden,"2,702,703",,"23,912",,"2,675,854",,"+69",,"2,937",,"8",,"264,479",,"2,340",,"19,500,873",,"1,908,301",,"10,218,971"

41

39,43,Senegal,"2,536,151",,"18,000",,"2,495,687",,"12,464",,"11",,"291,930",,"2,080",,"12,697,546",,"1,467,413",,"8,653,016"

PYTHON PROJECT on WEB SCRAPPING

```
In [31]: !pip install mysql-connector-python
```

Requirement already satisfied: mysql-connector-python in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (8.0.32)
Requirement already satisfied: protobuf<=3.20.3,>=3.11.0 in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (from mysql-connector-python) (3.20.3)

```
In [32]: !pip install pymysql
```

Requirement already satisfied: pymysql in c:\users\kaustubha r gawas\anaconda3\lib\site-packages (1.0.3)

```
In [33]: coronainfo
```

Out[33]:

	SrNo	Country Name	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious/Critical	Total cases/1M POP	Deaths/1M POP	Total Tests	Total Tests/1M POP
0	1	USA	106,418,595		1,157,462		104,260,123		1,001,010	1,722	317,852	3,457	1,175,622,556	3,511,362
1	2	India	44,776,002		531,016		44,204,771		40,215	N/A	31,832	378	923,053,282	656,215
2	3	France	39,867,463		165,916		39,569,363		132,184	869	607,879	2,530	271,490,188	4,139,547
3	4	Germany	38,377,656		171,748		38,143,900	+5,100	62,008	N/A	457,511	2,047	122,332,384	1,458,359
4	5	Brazil	37,358,092		700,811		36,249,161		408,120	N/A	173,473	3,254	63,776,166	296,146
...
215	222	Saint Helena	2,166				2		2,164		354,211			
216	224	Montserrat	1,403		8		1,376		19		282,578	1,611	17,762	3,577,442
217	225	Niue	800				797		3		493,218			
218	230	Tokelau	5						5		3,628			
219	231	China	503,302		5,272		379,053		118,977	N/A	347	4	160,000,000	110,461

220 rows × 15 columns



PYTHON PROJECT on WEB SCRAPPING

```
In [34]: import mysql.connector as mycon
```

```
In [35]: conn=mycon.connect(host="localhost",user="root",database="coronacases")  
print("Connection established successfully")
```

Connection established successfully

```
In [36]: if conn . is_connected():  
        mycursor=conn.cursor()  
        mycursor.execute("create table if not exists covidcases1(SrNo int, Country_Name varchar(50), Total_cases varchar(255), New_ca  
        for i, row in coronainfo.iterrows():  
            sql1="insert into covidcases1 values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"  
            mycursor.execute(sql1,tuple(row))  
            conn.commit()
```

```
In [42]: q="select * from covidcases1"  
mycursor.execute(q)
```

PYTHON PROJECT on WEB SCRAPPING

phpMyAdmin

Server: 127.0.0.1 » Database: coronacases » Table: covidcases1

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Extra options

SrNo	Country_Name	Total_cases	New_cases	Total_deaths	New_deaths	Total_recovered	New_recovered	Active_cases	Serious_critical	Total_cases_per1M_POP	Deaths_per1M_POP	Total_tests	Total_
1	USA	106,418,595		1,157,462		104,260,123		1,001,010	1,722	317,852	3,457	1,175,622,556	3,511,
2	India	44,776,002		531,016		44,204,771		40,215	N/A	31,832	378	923,053,282	656,2
3	France	39,867,463		165,916		39,569,363		132,184	869	607,879	2,530	271,490,188	4,139,
4	Germany	38,377,656		171,748		38,143,900	+5,100	62,008	N/A	457,511	2,047	122,332,384	1,458,
5	Brazil	37,358,092		700,811		36,249,161		408,120	N/A	173,473	3,254	63,776,166	296,1
6	Japan	33,547,551		74,164		21,725,273		11,748,114	63	267,131	591	98,881,178	787,3
7	S. Korea	30,956,026	+11,596	34,361	+5	30,734,900	+9,347	186,765	127	603,080	669	15,804,065	307,8
8	Italy	25,715,384		189,262		25,393,597		132,525	85	426,721	3,141	270,400,793	4,487,
10	Russia	22,734,824		397,681		22,116,159		220,984	N/A	155,925	2,727	273,400,000	1,875,
11	Turkey	17,232,066		102,174		N/A	N/A	N/A		201,399	1,194	162,743,369	1,902,
12	Spain	13,798,747		120,426		13,645,949		32,372	231	295,355	2,578	471,036,328	10,08,
13	Vietnam	11,528,303		43,186		10,615,180		869,937	9	116,502	436	85,826,548	867,3
14	Australia	11,352,930		19,933		11,291,664		41,333	45	435,499	765	78,835,048	3,024,
15	Taiwan	10,239,998		19,005		10,220,950	+55	43		428,656	796	30,742,304	1,286,
17	Netherlands	8,610,372		22,992		8,580,796	+555	6,584	53	500,270	1,336	25,984,435	1,509,
18	Iran	7,597,982		145,571		7,342,490		109,921	407	88,325	1,692	55,353,767	643,4
19	Mexico	7,556,927		333,619		6,804,253		419,055	N/A	57,440	2,536	19,835,770	150,7
20	Indonesia	6,753,593		161,082		6,585,054		7,457	N/A	24,195	577	114,158,919	408,9
21	Poland	6,505,726		119,432		5,335,940		1,050,354	696	172,384	3,165	38,587,511	1,022,
22	Colombia	6,363,544		142,698		6,186,501		34,345	342	123,533	2,770	36,951,507	717,3
23	Austria	6,048,674		22,217		6,012,291		14,166	55	667,130	2,450	211,273,524	23,30
24	Greece	5,965,643		36,582		5,921,203	+1,964	7,858	115	578,255	3,546	102,228,365	9,909,
25	Portugal	5,577,825		26,480		5,545,402		5,943	61	550,050	2,611	46,065,483	4,542,
26	Ukraine	5,484,936		111,789		5,334,190		38,957		126,989	2,588	32,603,805	754,8
		5,275,500		64,407		5,206,447		4,666	64	274,052	2,250	40,100,604	2,554

Console

PYTHON PROJECT on WEB SCRAPPING

```
C:\> XAMPP for Windows - mysql.exe -h localhost -u root
# mysql.exe -h localhost -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 166
Server version: 10.4.27-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| coronacases |
| employee |
| infomatch |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
8 rows in set (0.001 sec)

MariaDB [(none)]> use coronacases;
Database changed
MariaDB [coronacases]> show tables;
+-----+
| Tables_in_coronacases |
+-----+
| covidcases1 |
+-----+
1 row in set (0.001 sec)
```

PYTHON PROJECT on WEB SCRAPPING

XAMPP for Windows - mysql.exe -h localhost -u root

MariaDB [coronacases]> select * from covidcases1;

SrNo	Country_Name	Total_cases	New_cases	Total_deaths	New_deaths	Total_recovered	New_recovered	Active_cases	Serious_critical	Total_cases_per1M_POP	Deaths_per1M_POP	Total_tests	Total_tests_per1M_POP	Population
1	USA	106,418,595		1,157,462		104,260,123		1,001,010	1,722	317,852	3,457	1,175,622,556	3,511,362	334,805,269
2	India	44,776,002		531,016		44,204,771		40,215	N/A	31,832	378	923,053,282	656,215	1,406,631,776
3	France	39,867,463		165,916		39,569,363		132,184	869	607,879	2,530	271,490,188	4,139,547	65,584,518
4	Germany	38,377,656		171,748		38,143,900	+5,100	62,008	N/A	457,511	2,047	122,332,384	1,458,359	83,883,596
5	Brazil	37,358,092		700,811		36,249,161		408,120	N/A	173,473	3,254	63,776,166	296,146	215,353,593
6	Japan	33,547,551		74,164		21,725,273		11,748,114	63	267,131	591	98,881,178	787,366	125,584,838
7	S. Korea	30,956,026	+11,596	34,361	+5	30,734,900	+9,347	186,765	127	603,000	669	15,804,065	307,892	51,329,899
8	Italy	25,715,384		189,262		25,393,597		132,525	85	426,721	3,141	270,400,793	4,487,029	60,262,770
10	Russia	22,734,824		397,681		22,116,159		220,984	N/A	155,925	2,727	273,400,000	1,875,095	145,805,947
11	Turkey	17,232,066		102,174		N/A	N/A	N/A		201,399	1,194	162,743,369	1,902,052	85,561,976
12	Spain	13,798,747		120,426		13,645,949		32,372	231	295,355	2,578	471,036,328	10,082,298	46,719,142
13	Vietnam	11,528,303		43,186		10,615,180		869,937	9	116,502	436	85,826,548	867,342	98,953,541
14	Australia	11,352,930		19,933		11,291,664		41,333	45	435,499	765	78,835,048	3,024,116	26,068,792
15	Taiwan	10,239,998		19,005		10,220,950	+55	43		428,656	796	30,742,304	1,286,903	23,888,595
17	Netherlands	8,610,372		22,992		8,580,796	+555	6,584	53	500,270	1,336	25,984,435	1,509,718	17,211,447
18	Iran	7,597,982		145,571		7,342,490		109,921	407	88,325	1,692	55,353,767	643,478	86,022,837
19	Mexico	7,556,927		333,619		6,804,253		419,055	N/A	57,440	2,536	19,835,770	150,770	131,562,772
20	Indonesia	6,753,593		161,082		6,585,054		7,457	N/A	24,195	577	114,158,919	408,975	279,134,505
21	Poland	6,505,726		119,432		5,335,940		1,050,354	696	172,384	3,165	38,587,511	1,022,462	37,739,785
22	Colombia	6,363,544		142,698		6,186,501		34,345	342	123,533	2,770	36,951,507	717,327	51,512,762
23	Austria	6,048,674		22,217		6,012,291		14,166	55	667,130	2,450	211,273,524	23,302,116	9,066,710
24	Greece	5,965,643		36,582		5,921,203	+1,964	7,858	115	578,255	3,546	102,228,365	9,909,078	10,316,637
25	Portugal	5,577,825		26,480		5,545,402		5,943	61	550,050	2,611	46,065,483	4,542,692	10,140,570
26	Ukraine	5,484,936		111,789		5,334,190		38,957		126,989	2,588	32,603,005	754,855	43,102,122
27	Chile	5,275,580		64,497		5,206,417		4,666	64	274,053	3,350	49,180,601	2,554,811	19,250,195
28	Malaysia	5,056,911		36,994		5,006,634		13,283	10	152,403	1,115	68,342,991	2,059,698	33,181,072
29	Israel	4,817,269		12,416		4,798,473		6,380	21	516,542	1,331	41,373,364	4,436,346	9,326,000
30	Belgium	4,782,863		34,115		4,715,583	+2,583	33,165	43	409,903	2,924	36,950,938	3,166,786	11,668,278
32	Thailand	4,728,967		33,940		4,692,636		2,391	N/A	67,481	484	17,270,775	246,450	70,078,203
33	Czechia	4,636,934		42,713		4,589,926		4,295	28	431,874	3,978	57,139,828	5,321,876	10,736,784
34	Canada	4,634,277		52,121		4,559,665	+1,254	22,491	99	120,721	1,358	66,343,123	1,728,207	38,388,419
35	Peru	4,496,200		219,944		4,272,599		3,657	33	133,481	6,530	38,392,325	1,139,772	33,684,208
36	Switzerland	4,399,088		14,452		4,379,844	+1,832	4,792	39	501,398	1,647	23,833,472	2,716,487	8,773,637
37	Philippines	4,084,255		66,431		4,008,470		9,354	162	36,302	590	34,890,687	310,115	112,508,994
38	South Africa	4,075,512		102,595		3,912,506		60,411	192	67,080	1,689	26,795,090	441,027	60,756,135
39	Romania	3,380,891		67,961		3,299,960		12,970	132	177,649	3,571	27,131,523	1,425,624	19,031,335
40	Denmark	3,179,265		8,441		3,169,676		1,148	19	544,866	1,447	129,249,625	22,150,940	5,834,950
41	Hong Kong	2,886,027		13,488		2,870,274		2,265	28	379,526	1,774	76,127,725	10,011,143	7,604,299
42	Sweden	2,702,703		23,912		2,675,854	+69	2,937	8	264,479	2,340	19,500,873	1,908,301	10,218,971
43	Serbia	2,526,151		18,000		2,495,687		12,464	11	291,939	2,080	12,697,546	1,467,413	8,653,016
44	Iraq	2,465,545		25,375		2,439,497		673	21	58,474	602	19,544,451	463,523	42,164,965
45	Singapore	2,290,689		1,727		2,149,583		147,379	2	386,754	291	24,756,666	4,165,302	5,943,546
46	New Zealand	2,286,481		4,045		2,271,682		10,754		466,800	826	7,768,604	1,586,011	4,898,203
47	Hungary	2,199,764		48,811		2,146,693		4,260	3	228,993	5,081	11,394,556	1,186,160	9,606,259
48	Bangladesh	2,038,091		29,446		1,998,448		10,197	N/A	12,140	175	15,254,399	90,862	167,885,689
49	Slovakia	1,865,894		21,126		1,844,134		634	17	341,727	3,869	7,421,358	1,359,175	5,460,193
50	Georgia	1,833,502		17,019		1,776,548		39,935		461,986	4,288	16,920,079	4,263,340	3,968,738
51	Jordan	1,746,997		14,122		1,731,007		1,868	59	169,597	1,371	17,201,885	1,669,945	10,300,869
52	Ireland	1,708,435		8,782		1,695,720		3,933	19	340,312	1,749	13,083,449	2,606,161	5,020,199
53	Pakistan	1,580,272	+28	30,654		1,538,689		10,929	18	6,886	134	30,570,862	133,213	220,488,994
54	Norway	1,481,980		5,213		1,475,441		1,326	20	268,895	946	11,002,430	1,996,315	5,511,370
55	Finland	1,468,123		9,097		1,455,949	+108	3,077	21	264,290	1,638	12,069,254	2,172,699	5,554,960
56	Kazakhstan	1,410,549		13,848		1,383,020		13,681	24	73,447	721	11,575,012	602,707	19,205,043
57	Slovenia	1,342,241		7,100		1,328,006		7,135	8	645,919	3,417	2,847,701	1,370,382	2,078,034
58	Lithuania	1,316,418		9,649		1,302,623		4,146	7	494,576	3,625	10,425,065	3,916,682	2,661,708
59	Bulgaria	1,301,846		38,285		1,260,854		2,707	40	190,201	5,593	11,148,612	1,628,819	6,844,597
60	Morocco	1,272,679		16,296		1,256,151		232	3	33,693	431	13,001,033	344,191	37,772,756
61	Croatia	1,271,276		18,091		1,252,694		491	4	313,177	4,457	5,560,149	1,369,736	4,059,286
62	Guatemala	1,246,196		20,189		1,225,148		859	5	67,057	1,086	7,142,306	384,325	18,584,039
63	Lebanon	1,235,609		10,883		1,087,587		137,139	74	184,837	1,628	4,795,578	717,380	6,684,849
64	Costa Rica	1,226,248		9,326		860,711		356,211	52	236,620	1,800	4,659,757	899,158	5,182,354