

# Weather Data Scraping Exercise

## Programming Exercise: Web Scraping, Data Processing, and CSV Export

---

**Objective:** Scrape data from a public API or website, process the data using `pandas`, and export the processed data to a CSV file.

**Instructions:**

**1. Scrape Public Data**

Use the [Open-Meteo API](#) to retrieve current weather data (temperature, wind speed, humidity) for a list of cities. You will need to:

- Use the `requests` library to call the API.
- Fetch weather data (temperature, wind speed, humidity) for at least 10 cities
  - The API requires to provide latitude and longitude for each city, bonus question is to use external API to get the lat/long from city name but for starting please use the attached cities dict.

**2. Data Processing**

Using the `pandas` library:

- Create a DataFrame from the collected data.
- Calculate additional fields like:
  - Temperature in Fahrenheit if the API returns it in Celsius.
  - Wind speed in miles per hour if the API returns it in meters per second.
- Filter or rank cities based on specific criteria (e.g., highest temperature or lowest humidity).

**3. Export to CSV**

Export the processed DataFrame to a CSV file named `weather_data.csv`.

---

**Requirements:**

- **Use third-party libraries:**
  - `requests` for API calls
  - `pandas` for data manipulation
  - (Optional) `dotenv` to manage API keys securely

- **Program structure:**

The program should:

- Accept user input for cities (a predefined list can also be used for simplicity).
  - Retrieve weather data for these cities.
  - Process and display the data.
  - Save the processed data as a CSV.
- 

**Example Output:**

A CSV file `weather_data.csv` containing:

City	Temperature (C)	Temperature (F)	Humidity (%)	Wind Speed (m/s)	Wind Speed (mph)
------	--------------------	--------------------	--------------	---------------------	---------------------

New York	10	50	80	5	11.2
Tokyo	15	59	60	3	6.7
London	12	53.6	70	4	8.9

---

### Submission:

- **Code:** Submit the Python script.
- **CSV Output:** Include the generated CSV file.
- **Documentation:** Add comments explaining key parts of the code.
- **Optional:** Write a brief README on how to run the script and interpret the output.

---

### Bonus (Optional):

- Visualize the data using `matplotlib` or `seaborn` (e.g., bar chart of temperatures).

### Bonus Plus (Really Optional)

- Serve the Graphs via HTTP Endpoints:
- Using a web framework like `FastAPI` or `Flask`, create endpoints to:
- Serve the visualized graphs as images or plots.
- Provide the CSV file for download.

## Attached City List

```
1 cities = [  
2     {"City": "New York", "Latitude": 40.7128, "Longitude": -74.0060},  
3     {"City": "Tokyo", "Latitude": 35.6895, "Longitude": 139.6917},  
4     {"City": "London", "Latitude": 51.5074, "Longitude": -0.1278},  
5     {"City": "Paris", "Latitude": 48.8566, "Longitude": 2.3522},  
6     {"City": "Berlin", "Latitude": 52.5200, "Longitude": 13.4050},  
7     {"City": "Sydney", "Latitude": -33.8688, "Longitude": 151.2093},  
8     {"City": "Mumbai", "Latitude": 19.0760, "Longitude": 72.8777},  
9     {"City": "Cape Town", "Latitude": -33.9249, "Longitude": 18.4241},  
10    {"City": "Moscow", "Latitude": 55.7558, "Longitude": 37.6173},  
11    {"City": "Rio de Janeiro", "Latitude": -22.9068, "Longitude": -43.1729}  
12 ]
```