## Requests Module

**Problem Statement: Implement a Python program that interacts with a public API (e.g., a weather API) to retrieve and display the current weather for a list of cities, also using HTTP methods**

Firstly , the requests library is imported to facilitate making HTTP requests. This library simplifies the process of sending HTTP requests and handling responses from web services.

A class name WeatherAPI is created to do the functionality related to managing a list of cities and fetching weather data.

Moreover, self.cities is the empty list that will hold the names of cities added by the user, add_city checks if the city is already exists else appends the new one.

The `add_city` method:

- Checks if the city is already in the list.
- If not, it appends the city to `self.cities` and prints a confirmation message, indicating the action simulates a POST request.

The `update_city` method:

- Finds the old city and updates it with the new city name.
- If the old city is not found, it informs the user.

The `delete_city` method:

- Removes the specified city from the list if it exists.
- If not found, it alerts the user.

The `get_weather` method:

- Constructs the API URL using the city name and the API key.
- Sends a GET request to fetch current weather data.
- If successful (HTTP status code 200), it parses the JSON response to extract relevant information (city name, temperature, and weather description).
- Returns the weather data as a dictionary; otherwise, it returns `None`.

The `display_weather` method:

- Iterates through the list of cities and calls `get_weather` for each.

Displays the weather information if successfully retrieved; otherwise, indicates failure. The `main` function:

- Initializes the API key and creates an instance of the `WeatherAPI` class.
- Enters an infinite loop displaying a menu for user options.
- Based on the user's choice, it calls the respective methods for adding, updating, deleting cities, or displaying weather information.

API Key is needed to authenticate requests to the OpenWeatherMap API

1. Simulated HTTP Methods:
    - POST: In the `add_city` method, the operation is annotated as a POST request, simulating adding a city.
    - PUT: In the `update_city` method, it's labeled as a PUT request, reflecting the update action.
    - DELETE: The `delete_city` method indicates a DELETE request, which corresponds to removing a city.
2. Consistent Messaging:
    - Each city management action now includes a message indicating the simulated HTTP method that corresponds to the action taken.

Furthermore, the code implements a command-line application that manages a list of cities and retrieves their weather data through a public API(OpenWeatherMap). Thus using HTTP methods(POST,PUT,DELETE) for adding,updating and deleting cities, while using a GET request to fetch weather data.