In [1]:

```
# Load libraries
import requests
```

In [2]:

```
def get_city_coordinates():  # takes city and state info to find lat and lon
    base_url = 'http://api.openweathermap.org/geo/1.0/direct?'
    api_key = '02629a596bd71906573eb116ead70f5c'
    city = input('What city would you like to look up?\n').capitalize()
    state_code = input('What state is this city found in? Use the abbreviation. For example, Missouri = MO\n').upper()
    country_code = 'US'
    url = base_url + 'q=' + city + ',' + state_code + ',' + country_code + '&appid=' + api_key

    try: # checks for HTTP errors
        response_test = requests.get(url, timeout=10)
        response_test.raise_for_status()  # raises HTTP errors
        print('Request successful! Retrieving weather data now.')
        try:  # checks for errors in retrieving JSON data
            response = requests.get(url, timeout=10).json()
            lat = response[0]['lat']  # stores lat in variable
            lon = response[0]['lon']  # stores lon in variable
            return str(lat), str(lon), city, state_code
        except IndexError:
            print('Error')
            get_city_coordinates()
    except requests.exceptions.HTTPError as err:  # catches HTTP errors
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.ConnectionError as err:  # catches connection errors
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.Timeout as err:  # catches if request times out
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.RequestException as err:  # catches ambiguous errors not already caught by other statements
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
```

In [3]:

```
def get_zip_coordinates():  # takes zip code to find lat and lon
    base_url = 'http://api.openweathermap.org/geo/1.0/zip?'
    api_key = '02629a596bd71906573eb116ead70f5c'
    zip_code = input('What zip code would you like to look up?\n')
    country_code = 'US'
    url = base_url + 'zip=' + zip_code + ',' + country_code + '&appid=' + api_key

    try:
        response_test = requests.get(url, timeout=10)
        response_test.raise_for_status()  # raises HTTP errors
        print('Request successful! Retrieving weather data now.')
        try:  # accesses JSON data and stores lat, lon, and city as variables
            response = requests.get(url, timeout=10).json()
            lat = response['lat']
            lon = response['lon']
            city = response['name']
            return str(lat), str(lon), city
        except KeyError:
            print('Error, please try again\n')
            get_zip_coordinates()
    except requests.exceptions.HTTPError as err:
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.ConnectionError as err:
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.Timeout as err:
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
    except requests.RequestException as err:
        print('The following error has occurred: ', err)
        print('Please resolve this problem and try running the program again.\n')
```

In [4]:

```
def run_weather_api(lat, lon):
    base_url = 'http://api.openweathermap.org/data/2.5/weather?'
    api_key = '02629a596bd71906573eb116ead70f5c'
    url = base_url + 'lat=' + lat + '&lon=' + lon + '&appid=' + api_key + '&units=imperial'

    try:
        response = requests.get(url, timeout=10).json()
    except requests.exceptions.HTTPError as err:
        print('The following error has occurred: ', err)
        run_weather_api(lat, lon)
    except requests.ConnectionError as err:
        print('The following error has occurred: ', err)
        run_weather_api(lat, lon)
    except requests.Timeout as err:
        print('The following error has occurred: ', err)
        run_weather_api(lat, lon)
    except requests.RequestException as err:
        print('The following error has occurred: ', err)
        run_weather_api(lat, lon)

    print("Today's weather is:")

    temp = response['main']['temp']  # accessing actual temp
    print(f'Temperature: {temp:.2f}°F')  # prints and formats actual temp

    feels_like = response['main']['feels_like']  # accessing feels like temp
    print(f'Feels like: {feels_like:.2f}°F')  # prints and formats feels like temp

    temp_min = response['main']['temp_min']  # accessing min temp
    print(f'Low temperature: {temp_min:.2f}°F')  # prints and formats minimum temp

    temp_max = response['main']['temp_max']  # accessing max temp
    print(f'High temperature: {temp_max:.2f}°F')  # prints and formats maximum temp

    pressure = response['main']['pressure']  # accessing pressure
    print(f'Pressure: {pressure:.2f}inHg')  # prints and formats pressure

    humidity = response['main']['humidity']  # accessing humidity
    print(f'Humidity: {humidity}%')  # prints and formats humidity

    description = response['weather'][0]['description']  # accessing weather description
    print(f'General weather: {description}\n')  # prints and formats general weather description
```

In [5]:

```
def main():
    print("Welcome! Let's find the weather!")
    location_type = input("Would you like to enter a city or zip code?\n").lower()

    if location_type == 'city':
        print('You have chosen to enter a city.')
        lat, lon, city, state_code = get_city_coordinates()  # stores return variables to be used in other function
        print(f'\nWeather data for {city}, {state_code}:')
        run_weather_api(lat, lon)
        continue_running = input("Would you like to find the weather for another location? "
                                 "Type 'y' to continue or any other key to quit.\n")
    if location_type == 'zip code':
        print('You have chosen to enter a zip code.')
        lat, lon, city = get_zip_coordinates()  # stores return variables to be used in other function
        print(f'\nWeather data for {city}:')
        run_weather_api(lat, lon)
    else:
        print("Error, please type either 'city' or 'zip code' to find the weather. \n")  # Catches incorrect user input
        location_type = input("Would you like to enter a city or zip code?\n").lower()
        if location_type == 'city':
            lat, lon, city, state_code = get_city_coordinates()
            print(f'\nWeather data for {city}, {state_code}:')
            run_weather_api(lat, lon)
        if location_type == 'zip code':
            lat, lon, city = get_zip_coordinates()
            print(f'\nWeather data for {city}:')
            run_weather_api(lat, lon)

    # Continues running program until user chooses to quit
    continue_running = input("Would you like to find the weather for another location? "
                             "Type 'y' to continue or any other key to quit.\n")
    if continue_running == 'y':
        main()
    while continue_running != 'y':
        try:
            continue_running = continue_running.lower()  # Checks if user entered Y rather than y
            if continue_running == 'y':
                main()
        finally:
            print('You have exited the program.\n')
            break
```

```
In [ ]: if __name__ == "__main__":
            main()
```