

Final project

Weather Program

For your class project we will be creating an application to interact with a webservice in order to obtain data. Your program will use all of the information you've learned in the class in order to create a useful application.

Your program must prompt the user for their city or zip code and request weather forecast data from [OpenWeatherMap](#). Your program must display the weather information in a **READABLE** format to the user.

Requirements:

- Create a header for your program just as you have in the past.
- Create a Python Application which asks the user for their zip code or city (Your program must perform both a city and a zip lookup). You must ask the user which they want to perform with each iteration of the program.
- Use the zip code or city name in order to obtain weather forecast data from [OpenWeatherMap](#).
- Display the weather forecast in a readable format to the user. Do not display the weather data in Kelvin, since this is not readable to the average person. You should allow the user to choose between Celsius and Fahrenheit and ideally also Kelvin.
- Use comments within the application where appropriate in order to document what the program is doing. Comments should add value to the program and describe important elements of the program.
- Use functions including a main function and a properly defined call to main. You should have multiple functions.
- Allow the user to run the program multiple times to allow them to look up weather conditions for multiple locations.
- Validate whether the user entered valid data. If valid data isn't presented notify the user. Your program should never crash with bad user input.
- Use the **Requests** library in order to request data from the webservice.
 - Use Try blocks to ensure that your request was successful. If the connection was not successful display a message to the user.
- Use **Python 3**
- Use try blocks when establishing connections to the webservice. You must print a message to the user indicating whether or not the connection was successful.
- You must have proper coding convention including proper variable names (See PEP8).

Deliverables:

PEP8).

Deliverables:

- Final Program in a .py file (Due week 12)

Project Notes:

- Start early. Ask lots of questions when you don't understand something. It's far better to get clarity than to not meet requirements.
- Be creative. This assignment is a real-world program. Use it as an opportunity to improve your knowledge and showcase what you've learned.
- Sign up for API Key <http://openweathermap.org/appid>
- The API key will look something similar to this:
d5751b1a9e2e4b2b8c7983646072da8b
- Make a connection to the API using the Requests library.
- You MUST do a GEO Lookup first then do a weather lookup using the latitude and longitude. This will require you to do 2 API calls. One call will be to obtain the LAT and LON and the other will be to get the weather using the LAT and LON.
- READ all of the OpenWeather GEOCode and Weather Lookup API documentation. Most of the questions you have can be answered by the API documentation.
- Make sure that your try blocks are solid. Don't include huge blocks of code in the try blocks. Don't use generic exceptions. Take a look at the request documentation on the various exceptions you can catch for HTTP connections. You should have specific exceptions with meaningful messages for the end user to make adjustments.
- Make sure you have request specific exceptions. Your code should not throw any unhandled exceptions.
- Make sure that your program allows a user to do a zip code weather lookup and a city/state lookup. You must give them the option and I will test both. For city you should ask the user to enter a state, otherwise there's no way to distinguish between Omaha TX, Omaha AR, and Omaha NE.
- I will put gibberish in your prompts. This is an example of non happy path testing and is a real-world testing scenario. Make sure you have lots of validation to test those scenarios. If you ask the user to enter a number for a menu how will your program respond if the user enters characters for instance.
- Make sure that your program has really good comments. Comments should be meaningful and provide value for other developers looking at your code.
- Make sure you follow PEP8 guidelines. Don't use single variable names etc...
- Make sure to review the requests library
- Level of effort counts. Make sure you're putting your all into this final program

