Scripting Language Programming

Assignment 2 - Weather/Display for cities

Overview

There are many online services such as weather prediction, that can be combined with other forms of data (mashups) to provide valuable services.

This assignment will combine weather predictions for various locations around the world.

Marking

The assigment is worth 20% of the overall marks for the subject. The marking will be broken up into

Criteria	Marks
Working to specifications	16%
Quality of code	–2% to 2%
Layout and comments	–2% to 2%
Following license agreements	–2% to 0%

Quality of code refers to the use of appropriate Python features; broken into appropriate subroutines, and ease of reading, choice of variable names etc..

Layout and comments refers to the spacing of your code for readability, and how well your comments help the reader understand your code. This does not mean that every line needs to be commented. Typically you would comment each function/class and any particularly complicated code.

Following license agreements refers to the following the license agreements listed at the websites of the data sources.

As you can see you can *lose* marks if your code doesn't meet at least some basic level of professionalism (quality, layout, comments).

The minimum mark you can get is 0, not –6.
The maximum mark you can get is 22. Any bonus marks will be restricted to the assignment/lab component, and will not be added to the exam component.

Due Date

This assignment is due on ***Sunday 21st May*** at 11:00pm

Submission

You should zip up any of your program's files and submit via Blackboard. Each of your files should have a comment containing your name and student number.

Each stage should be placed in a separate directory, and the final complete assignment zipped and submitted.
Your submission should follow the following directory structure…

```
3123456/
    stage1/
        files
        world_cities/ (an empty folder)
    stage2/
        files
        world_cities/ (an empty folder)
    stage3/
        files
        world_cities/ (an empty folder)
```

If you haven't attempted a stage you don't need to add a directory for it.

Your program should assume that the world cities files will live in a directory named `world_cities`. Do not submit this file! - it will be copied in when the marking is being done.

This directory will look like…

```
world_cities/
    worldcitiespop.txt
```

(you should structure your program development like this so that it will work when it is tested).

Problem specification

You are to implement the following stages of the program.

Stage 1 - 10%

```
Allow the user to specify a latitude/longitude, and a minimum city population...
```

Example usages would be…

```
python stage1.py 37S 140E 1000000 today 5pm
```

```
python stage1.py 37S 140E 1000000 10th April 2017 5pm
```

The general format is…

*python stage1.py {**reading**} {**reading**} {**minimum population**} {**date_specifier**} **time***

reading => **latitude** | **longitude**
latitude => digits N|S
longitude => digits E|W
date_specifier => ***day_of_the_week*** | date | now
day_of_the_week => today | tomorrow | Monday | Tuesday …| Sunday | Mon | Tue | … | Sun
time => hours:minutes[am|pm]

If now is specified, the time is optional.

The values can be in upper or lower case. The only input you can expect to be in quotation marks (i.e. presented as a single command line argument) is the city name. All other values will be individual command line arguments.

See the section “Data Sources” below for more information.

The program should then find the city nearest the specified locations with a population that equals or exceeds the poplutation you entered, and tell you what the weather will be on the date/time you requested.

Stage 2 - 6%

Using the webserver, routes.py file provided, create a locally hosted server on your computer that will create a web page with a list of nearby cities (identified by using a web location api api).
The web requests can be done via a web submit (using GET and POST). The backend must be based on the program from Stage 1.

Stage 3 (Bonus) - 2%

Use Ajax calls to interact with the server to allow a dynamic list of cities based on the current location, and minimum population size selected, and a max list of cities to display.
You should also include an option to sort the list on different criteria (e.g. size, distance).

Data sources

1. You should utilise the forecasting system from developer.forecast.io. This requires setting up a free account, and will provide you with a personalised key that you can use to access the data from their site. You should keep this key in a file named “forecastKey” (no suffix) to allow easy testing by staff.

   Note that you will need to follow the requirement to show a “Powered by Forecast” link on your website to meet the terms of the license. You will also need to ensure you don’t call the API more than 1000 times in one day if you wish to use the service for free.

2. A database of cities is available at free world cities database
3. You should call the web location api to identify your current location based on your IP address.

Once again note the you will need to follow the licensing terms for the data.