CSCI 3308 Software Development Methods and Tools [Spring 2018]
Instructor: Alan Paradise
Homework 4

REST – WEATHER MAP

Objectives

Access information via REST

Work with JSON formatted data

Display data with SVG file and jQuery

We recommend the you work in pairs!

Assignment

In this assignment, we create a web page showing the weather across the USA. We will use the Dark Sky Forecast API **from Lab 7** as the data source.

Each state will be color-coded per its current temperature.

Part 1 - Static HTML

1. Download a map of the United States in SVG format.

2. Create a new HTML file named ***index.html.***

3. In your HTML file, add the following code to create a web page:

```
<html>
    <head>
        <title>
            HW #4 Your First name(s) and last name(s)
        </title>
    </head>
    <body>
```

4. On the next line in the file, insert the downloaded SVG map. Copy-paste isn't very effective here, but you can do this in vim with the following:

:r Blank_US_Map.svg

{**This is a vim command, please don't type this as plain code in your html file. Open your html file using vim text editor and type the above command to get all the SVG code inserted between your html body tags.**}

5. Close the HTML page with:

</body>
</html>

6. Run a python server on your machine using the following commands: -
   - python –m SimpleHTTPServer 8000 (If your python version is less than 3)
   - python –m http.server 8000 (If your python version is 3 or above)

7. Go to your browser and open up http://localhost:8000/. The map of the US should show up.

8. Now convert this HTML file into a Python CGI script that prints this HTML code: -

   - Create a directory and name it ***cgi-bin.***
   - Create a file inside the ***cgi-bin*** directory named ***index.py***
   - Grant execute permissions to this file. (Hint: chmod +x)
   - The first line of this file should be the shebang line: -

     #!/usr/bin/env python3 ***(if python version is 3 or above)***

     OR

     #!/usr/bin/env python ***(if python version is below 3)***

   - Then add the lines

     print ('Content-type: text/html') ***(if python version is 3 or above)***

     print ('')

     OR

     print 'Content-type: text/html' ***(if python version is below 3)***

     print

   - Then convert the html code we have already to be in a string. The secret to doing this is to use the multi-line quotes, such as:

     contents = """<html>
     <body>
     ... etc. {actual code for world map} """
     print contents

   - Now save the file and navigate to the parent directory of cgi-bin in the terminal.

- Run a CGI server in python

  python3 –m http.server --cgi 8000 ***(If python version is 3 or above)***

  OR

  python –m CGIHTTPServer 8000 ***(if python version is below 3)***

- Now open up your browser and navigate to http://localhost:8000/cgi-bin/index.py. You should now see the map again!

Part 2 - Dynamic weather data

1. Now we want to call the same URL we used in **Lab 7** to get the weather. To start, do this with Boulder, CO. To do this in python, if you Google how to read the JSON from a URL you can find a great answer at StackOverflow.

Remember, if you use this source then attribute it in your code!

2. Our next step is to test manually changing the colors of the states. Add the following code at the end of your file:

```
print response
print '''
<script>
$( document ).ready(function() { '''

print '''
});
</script> '''
```

Note: We are dividing up the printing into two separate print statements because **we will need to add python code in the middle**.

The document ready function is a JavaScript jQuery library function, so to make this work we need to add inside the HTML head tag the following library reference:
<script src="http://code.jquery.com/jquery-2.2.1.js"></script>

3. Inside the ready function, add the following python print statement:

```
print "$('#CO').css('fill', 'red')"
```

Reload the page, and you should see Colorado filled in red!

4. Now we certainly don't want to manually type in all the states, and eventually

we'll need a city in each state that we can call the weather REST API to find out the temperature so we can color code the state based on the temperature.

We need to find something that can help us do that. We could use the capitals of each state as our city for our temperatures. So, we Google for something in python that has each state and capital in a format we can use.
This one looks good.

Copy-paste just the dictionary of ***state_caps*** into your file.

5. Instead of color-coding Colorado only, create a loop through the state_caps dictionary and color-code every state.

Hint: use the python string formatting "{0}".format() to insert the variable for the state! Google for more info.

6. Time to call the weather API for each state's capital so we can color-code each state based on the temperature!

Very important: We don't want to hit the API a billion times while we are testing (You will be denied service to the API if you make too many calls to it in an hour. It will also take a while to fetch weather information for all 50 states). So, comment out the last 45 states.

If you are really interested in coloring the whole map, it is ok if you color all the states. **(15 states are the minimum requirement)**

7. In your loop, you recently created, add code to get the temperature for each state in the dictionary (which should be 5 right now).

Before you call the API, Arkansas has a problem because the capital "Little Rock" has a space in it. Use python to replace all spaces in the name of the city with a plus sign before you add it to the URL string to call the weather API.

Print out the resulting temperatures. You should notice it doesn't appear to be Fahrenheit. (If it appears to be Fahrenheit, ignore the next step)

Figure out what unit of measurement is being used, then add a python function to convert it to Fahrenheit if required.

8. Determine which color should be used for the state. Based on the temperature, use the following color

Temperature	Color
[default]	Gray
<=10	Blue
11..30	Cyan
31..50	Green
51..80	Orange
>80	Red

Credit:

Submit a zip of the **cgi-bin** directory named in the format Firstname_Lastname_HW4.zip (If you are working in pairs the file name would be: LastName1_Lastname2_HW4.zip) to Moodle. Also make sure you have entered both your first name and last name in the title.

Make sure that you have at least 15 states colored!

Note: You can use Python2 or Python3 for this homework. The above write-up assumes that you are using Python 2x version. If you choose to work with Python3, there would be small differences other that the changes mentioned before. For example, print statements, urllib requests etc., (You must look for corresponding code in python3 and make it work).