

CSCI 3308 Software Development Methods and Tools [Fall 2017]

Instructor: Alan Paradise

Homework 4 – Material by Liz Boese & Grace Muzny

Due Date: 5:00 p.m. on Monday, November 13th via Moodle submission link

REST - WEATHER MAP

Objectives

Access information via REST

Work with JSON formatted data

Display data with SVG file and jQuery

We recommend that you work in pairs.

Assignment

In this assignment create a web page showing the weather across the USA. We will use [Dark Sky Forecast API](#) from lab 9 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 Firstname_Lastname_HW4.html. (If you are working in pairs the file name would be: PP_LastName1_Lastname2_HW4.html)

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

```
<html>
    <head>
        <title>HW #4 Your 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 (Use the [CGI version](#)).
7. Test the html file in your browser – the map of the US should show up.
8. Now convert this to a Python CGI script that prints this HTML code.
Copy your html file to a new file named Firstname_Lastname_HW4.py (If you are working in pairs the file name would be: PP_LastName1_Lastname2_HW4.py)

Be sure to place this .py file in the `cgi-bin` directory.

Check permissions! Grant permissions on this .py file.

The first line in this file should be the shebang line:

```
#!/usr/local/bin/python2
```

Then add this:

```
print "Content-type: text/html"  
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
```

You should now see the map again, via the python script.

Part 2 - Dynamic weather data

1. Now we want to call the URL we used in lab 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!

If you print out the results (the response from the URL), you may notice the letter 'u' everywhere. This means it is returning the values in Unicode. Search Google for a solution to convert to strings. Be sure to attribute your source! (If you don't see the letter 'u' everywhere, then you need not do any conversion to strings)

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 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 (I started to get denied service if I did it too much in an hour, and it takes a while to call it for all 50 states). So, comment out the last 45 states. If you are using vim – go to the line to start and do

```
:.,+44s/^/#/
```

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 your .py and .html files as a zip `Firstname_Lastname_HW4.zip` (If you are working in pairs the file name would be: `LastName1_Lastname2_HW4.zip`) to Moodle.

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 for cgi-script, print statements, url lib requests etc., (You must look for corresponding code in python3 and make it work)