UPDATE: Unfortunately, it looks like last week our openweathermap.org instituted a required API key to use. This is good practice since most APIs do require a key, so it is good to know how to do this. To do so, go to: http://openweathermap.org/appid
And click on Sign up. Fill in to create a new account. On success, you will see your API key. Copy it, and then click on the link at the bottom "How to work with your API key". Notice how the call needs to be set up, and add the id and API key to your string when you call urlopen. It should then work!

CSCI 3308 Software Methods & Tools [Fall 2015]
Instructor: Boese
Assignment #4

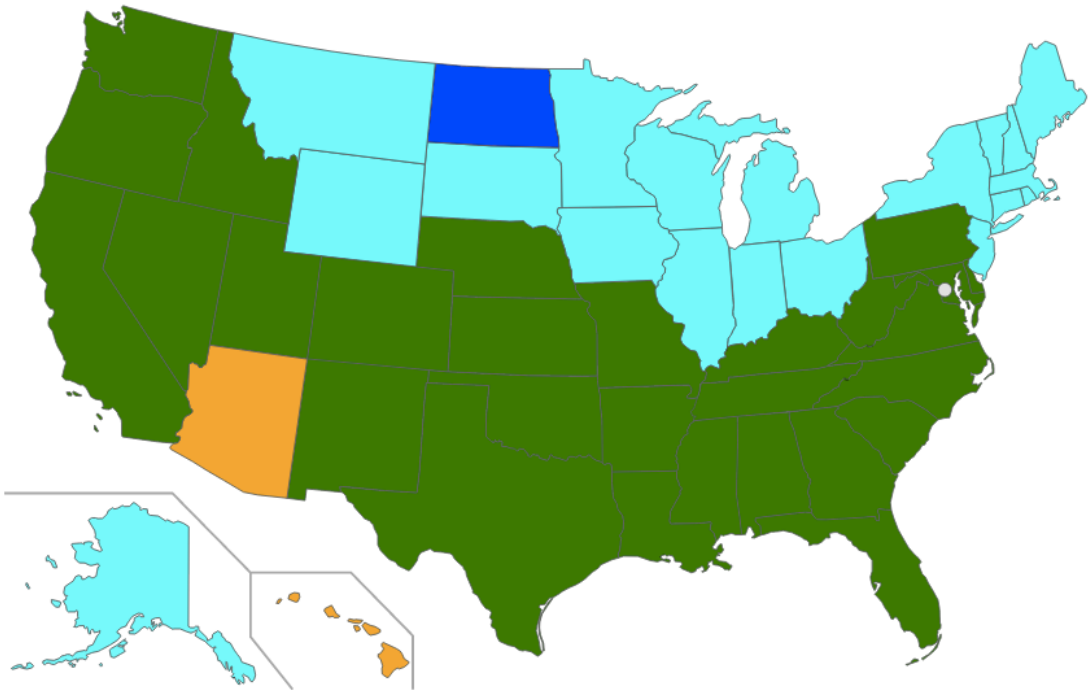
REST

Objectives

- Access information via REST
- Work with JSON formatted data
- Display data with SVG file and JQuery
- Use Google to find answers
- Recommend you work in pairs (If you do so, only ONE person submits in Moodle)

Assignment

In this assignment create a web page showing the weather across the USA.



Requirements/Specifications/Steps

Note: you will need an API key (as done in lab). To do so, go to: http://openweathermap.org/appid And click on Sign up. Fill in to create a new account. On success, you will see your API key. Copy it, and then click on the link at the bottom "How to work with your API key". Notice how the call needs to be set up, and add the id and API key to your string when you call urlopen.

1. First step is that you need a map of the USA. We will use the SVG file format from Wikipedia:
   a. Info: http://en.wikipedia.org/wiki/File:Blank_US_Map.svg
   b. File to download: http://upload.wikimedia.org/wikipedia/commons/3/32/Blank_US_Map.svg
2. Now create a new HTML file named **Firstname_Lastname_HW4.html**

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

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

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

```
:r Blank_US_Map.svg
```

5. Run a python server on your machine (Use the CGI version).
6. Test in your browser – the map of the US should show up.

7. Now convert this to a python CGI script that prints this HTML code.

   a. Copy your html file to a new file named **Firstname_Lastname_HW4.py**
   b. Be sure to put this in the cgi-bin directory like we did in lectures. Check permissions!
   c. The first line in this file should be the shebang stuff.
   d. Then add this:

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

   You DO need the extra print!

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

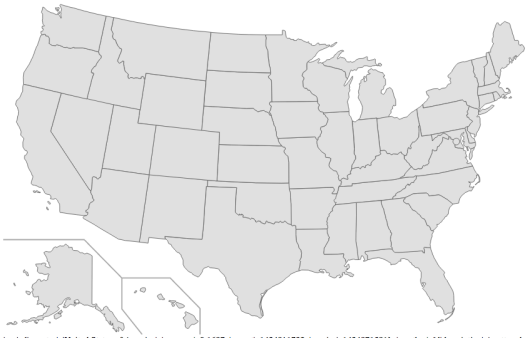
```
contents = '''
 <html>
 <body>
 … etc.
'''
print contents
```

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

8. 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:
http://stackoverflow.com/questions/13921910/python-urllib2-receive-json-response-from-url
Remember, if you use this source then attribute it in your code!

   a. If you print out the results, 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!*

At this point you should now see something like this:



9. 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>
'''
```

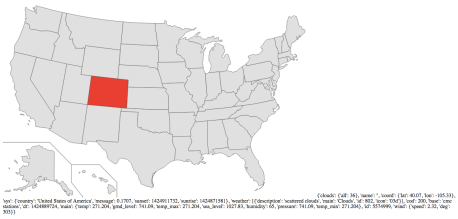
Note: We are dividing up the printing into two separate print statements because we will need to add python code in the middle.
The document ready function is a JavaScript JQuery library thing, so in order to make this work we need to add inside the HTML head tag the following library reference:

```
<script src="//code.jquery.com/jquery-1.11.2.min.js"></script>
```

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

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

Run your page again and you should see Colorado filled in red!

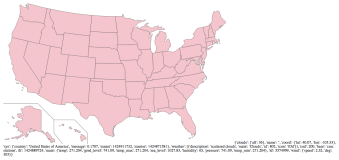


11. Now we certainly don't want to manually type in all the states, and eventually we'll need a city in each state that we can call the weather REST API to find out the temperature so we can color code the state based on the temperature.
    a. We need to find something that can help us do that. We could 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.
    b. This one looks good: https://github.com/tdlm/fun-with-python/blob/master/states_and_capitals.py

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

12. 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.

Your page should now look something like this:

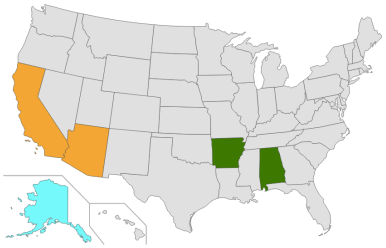


13. Time to call the weather API for each state's capital so we can color-code each state based on the temperature!

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
(vim – go to the line to start and do `:.,+44s/^/#/`
Another option is if you understand the JavaScript, you can load all 50 states information in ONE call to the API. Can you figure it out?

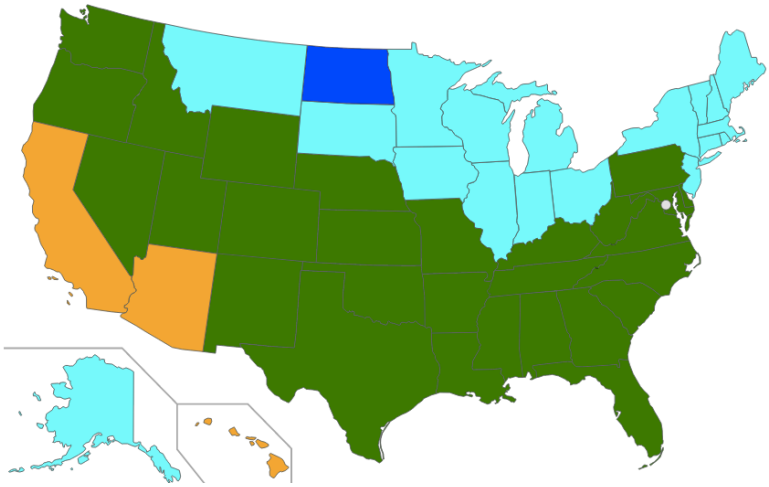
14. In your loop you recently created, add code to get the temperature for each state in the dictionary (which should be 5 right now).
    a. Before you call the API, Arkansas has a problem because the capital "Little Rock" has a space in it. Use python to replace all spaces in the name of the city with a plus sign before you add it to the URL string to call the weather API.
    b. Print out the resulting temperatures.
    You should notice it doesn't appear to be Fahrenheit.
    c. Figure out what unit of measurement is being used, then add a python function to convert it to Fahrenheit.
15. Determine which color should be used for the state. Based on the temperature, use the following color

[default]	Gray
< 10	Blue
10..30	Cyan
30..50	Green
50..80	Orange
> 80	Red



Submission
Submit your **.py** and **.html** files as a zip **Firstname_Lastname_HW4.zip** to Moodle. MAKE SURE ONLY 5 STATES SHOW UP! To make grading easier on the TAs. If you pair programmed name it **Lastname1_Lastname2_HW4.zip**

FOR YOU:
16. Once that is working, uncomment the rest of the states and run it again.



CONGRATS! You just did amazing stuff!