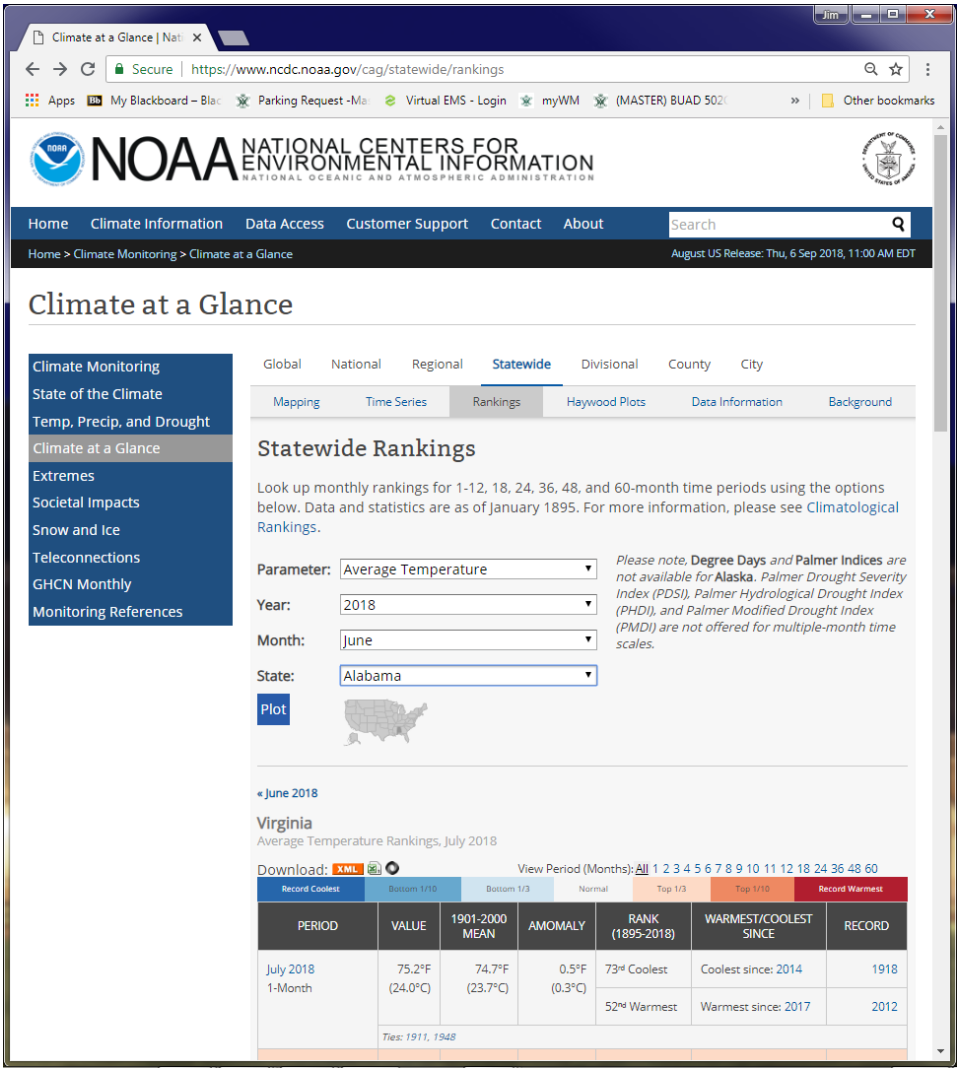
XML Web Scraping Assignment

You will scrape XML data from a NOAA web site in this assignment. This assignment statement provides considerable detail to help you construct your code.

- Use Google Chrome for the initial steps of investigating the NOAA site and determining the URL structure.
- Manually go to the web page noted below and choose the parameters as are, also, noted below before left-clicking on the `Plot` button:
    - https://www.ncdc.noaa.gov/cag/statewide/rankings
    - Parameters:
        - `Parameter`: Average Temperature
        - `Year`: 2018
        - `Month`: July
        - `State`: Virginia



- Left-click on the `Plot` button.
- Right Click on “XML” button and “Open link in New Tab”

- Observe the URL specification in that new tab and how the search parameters are embedded in it. (**Note: At present, there is an error in the web page, which we will discuss in class so you can obtain the correct URL.**)
- Write a Python program named `xml_scrape.py` to access Average Temperature Data in XML format using that URL structure found above by following these instructions/specifications:
    - Note the URL in your web browser that resulted from querying the previous parameter set and how the `Parameter`, `State`, `Month`, and `Year` parameters are indicated in the URL.
    - Build a URL to obtain the XML data for a different time period, as described in the parameter set below. Manually paste that URL into a Chrome web browser to ensure that it works.
        - `Parameter`: Average Temperature
        - `State`: Virginia
        - `Month`: August
        - `Year`: 2016
    - Code a Python program using the template provided as `xml_scrape.py` that incorporates these features and functionality:
        - Create a string variable for each of the parameters above to store string values in the form that the URL requires them.
        - Assign appropriate values to those variables to obtain XML data for the parameter set above.
        - Embed these parameters into a URL using the string substitution method, as described below. Start with a string that represents a "fixed" part of the URL that remains constant regardless of what parameters are of interest. Then use string substitution to embed the parameters
        - Retrieve the XML data with your program (or view it initially in a browser) and notice how you can identify the portion of the XML file that is associated with a 5-month window, April-August 2016
        - Extract these data fields from the XML data that you retrieve programmatically using the `lxml` package and print each of these data items on a separate line for the five-month period April-August, 2016, without any other printed text:
            - Your W&M username (this doesn't come from the web page)
            - `value`
            - `mean`
            - `departure`
            - `lowRank`
            - `highRank`
    - Submit your Python code file using one of these two options depending on your location. The first alternative works only if you are on campus. The second should work anywhere.
        - On Campus: Open a Windows File Explorer Window and paste this location into the address bar while substituting your W&M username for `your_username`:

`\\files.campus.wm.edu\acstore-classes\BUAD5012\student\your_username`

- Off Campus: FTP your file using the directions in the PowerPoint file from Blackboard named “FTP Access to Network Folder.pptx”

- Coding hint… use the string substitution method for URL generation
    - Create a string template with the symbols '`%s`' as placeholders for where you will insert the values for month, year, etc., for example,
        - `template = "My name is %s, %s"`
    - Then you can substitute string values for the '%s' symbols using a statement like this:
        - `last_name = 'Bradley'`
        - `first_name = 'Jim'`
        - `print template % (last_name, first_name)`
    - This results in a printout of '`My name is Bradley, Jim`'
    - You can use the same approach for this assignment by creating variable names for `Parameter`, `Year`, `Month`, and `State`, and substituting those values into a string template for the NOAA web page.