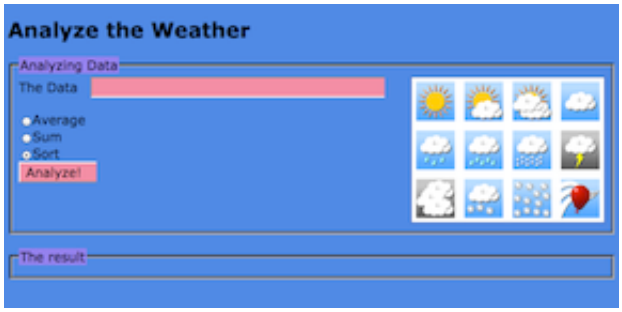
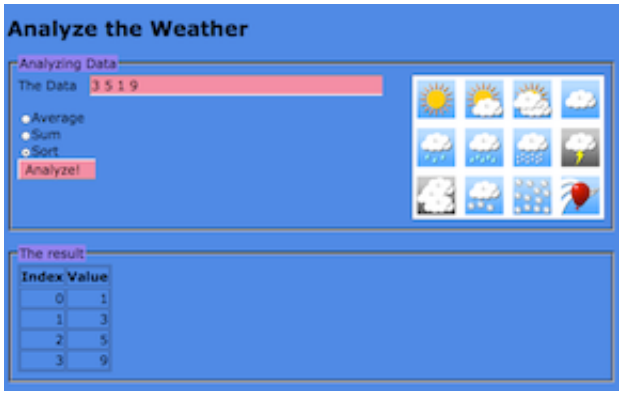
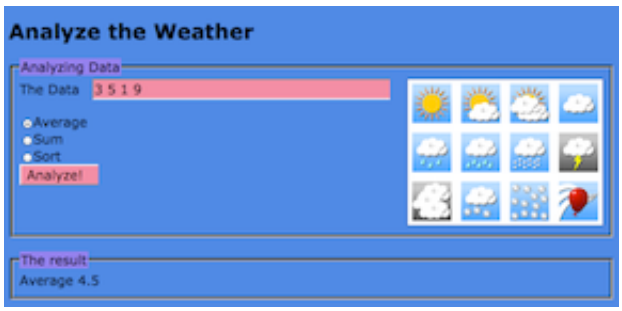
Create **js and html files** that accept a set of data, then either sum, average or sort the data. Put your files in the directory structure we use in class, and upload them to cscilab. Submit just the js and html files to Canvas, and also make sure to add a comment to your submission with the url of your page.

When first opened in Firefox, your page should look something like the image to the below.



After entering a title and some data separated by commas, selecting an operation and pressing the button, the page should show the data average, sum or the data in a table in order. Your page should work with any number of values entered.





To calculate the average:

1. Get the text representing all of the values from the input into a variable
2. Use the split(",") function to separate the text by commas, and put each separate value into an array. You can use indexOf(",") to make sure there is a comma in the string the user

entered.. I also allowed the user to enter data separated by space. You can do this by using these two functions and an if statement.

3. Use a for loop to index through the array and sum all the values.
4. After the loop is complete, divide the sum by the length of the array write the result to the result article (as you have done in previous assignments);
5. Display the average to exactly 1 decimal point.

To sort the data:

1. Repeat steps 1 and 2 above.
2. Sort the array in order from smallest to largest (use the sort() function). This is a little tricky with javaScript, but not too hard. If you call your array my array, as I did below, you can use the following javaScript function to get the desired result. If you just use the sort method, the sort will be alphabetical, so 5 will be greater than 40 **myarray.sort(function(a,b) {return a-b});**
3. Create a table in a string variable , and put the table header information in it. (i.e vary mytable = "<table><tr><th>…")
4. Use a loop to get the values from the array one at a time, adding a row to the table with each iteration.
5. After the loop, finish up the table element (i.e. add </table> to the mytable.
6. Write the table string to the result article.

The sum should be straightforward.

Use an image, and play with formatting the table.