Download the 4 starter files and placed them in a new directory called lab5 in your webpages directory.

Part 1 (25 pts)

You’re given an HTML file with an input field and some paragraphs. Your task is to get the value entered in the input field and use it to update the text 4 different paragraphs. Each paragraph needs to have the input styled a little different, however.

1. Change all the text to uppercase before adding it to the paragraph with id "allUpper"
2. Change all the text to lowercase before adding it to the paragraph with id "allLower"
3. Change the color of the text to "red" and add it to the paragraph with id "redText"
4. Add the text to the paragraph with id "flashyText", and also add the class "flashy" to the paragraph

A couple things to note:

- HTML input elements have an attribute, value, that has the text that is entered in the text box. You can access that string in JS by accessing that attribute.
- We want to use an event on the input box to make sure that each time new text is entered, we update all our paragraphs

Part 2 (25 pts)

Your task here is to add functionality to a simple calculator interface. You are given the HTML for the calculator, two input fields for two operands, and 4 buttons with different operator options.

Similar to part 1, you will be adding the result to the document, this time inside of a span with the id "result". You also want to show what operator was just click by adding the name of the operator to the span with id "operator". All this can be handled in an event handler for each button.

Note, in the result div we also have a *clear* button. This button should not only reset the value of the two input boxes (set it to an empty string), but it should also hide the entire result div, until the next time a result is shown (by selecting enter). We also want to start with the div hidden since no operation has been made at the beginning. During our CSS chapter, we discussed a CSS property that helped hide elements from the document. Here you can combine CSS and JS to toggle the result div from showing. Lastly, for division, please use the Math.floor( ) function to handle cases where the numbers do not divide evenly.

Part 3 (25 pts)

Write a function called getElementsByAttribute that returns all the elements that have a given attribute. The function should take the name of an attribute as a parameter and returns all the elements in the document that have that attribute.

Note: this does not have to be added to the document prototype (it is considered bad practice to modify the default JavaScript prototypes).

Part 4 (25 pts)

Take a look at the code in the script tag. Open the page in the browser and click each of the buttons. Does it show something unexpected? (Yes, there is a bug). Change the code to make it alert the correct message. Hint: Think closures.