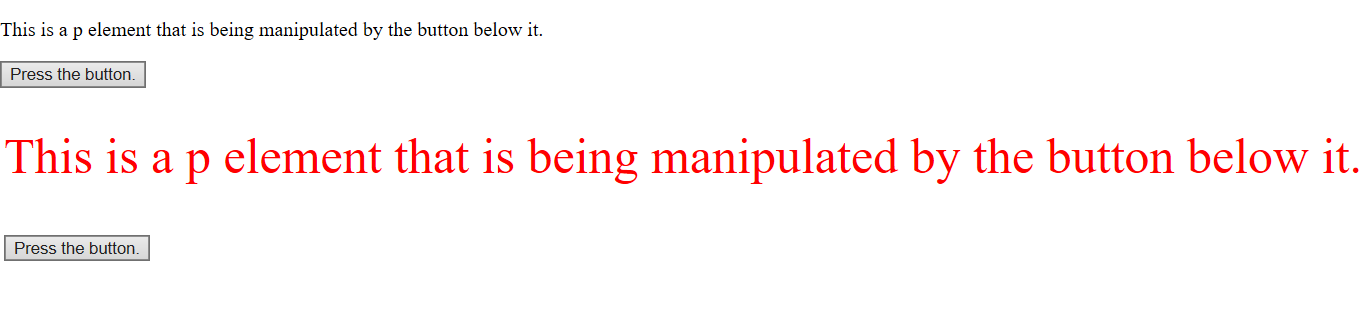
Lab 13!

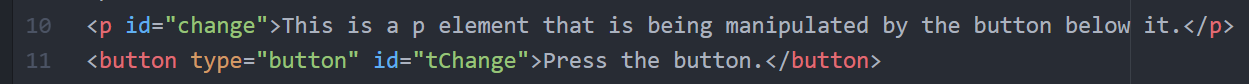
In this lab, students will get a basic breakdown of functions and variables in JavaScript. By the end, they should be able to create their own JS variables and functions.

Demo:

Have students download the index.html file from Webcourses, and have them open it in a browser.



It’s a simple button that makes some straightforward changes to the CSS of the p element. In this case, what’s going on under the hood is infinitely more interesting. Direct students to lines 10 and 11 in the text editor.



Essentially, two things are happening here. There is a p element with the id “change” and a button element with the id “tChange.” These are the HTML pieces that are going to be manipulated with the JS below. So, let’s talk to them about what’s happening down there. Walk them through what is happening here:



First off, we are using a script tag here. In practice, they’d often be using a dedicated .js file, but this one is pretty short, so we can get by with just putting it below the HTML. Start at line 24 (this is a bit counter intuitive, but we’re starting at the end and working back).

This is a function. They are packets of JavaScript that can be deployed all at once. In this case, the function is changing the size, the style, and the color of the font. When the user clicks the button, the p element changes. However, that function needs to be initialized.

From there, let’s jump back to line 17. This function exists to activate the function ‘change’. On line 19, there is a variable being declared. We’re making it equal to the button with the id ‘tChange’. getElementById is a command that targets a specific place in the HTML, and it does it by zooming in on a specific id. Remember that ids are unique identifiers, so this is a precise way to point at your HTML. By making the button ‘tChange’ equal to the variable ‘button’, we can give the variable ‘button’ an onclick in line 21. That is how the function ‘start’ activates the function ‘change’.

From there, jump down to line 31. This window.onload command runs the moment that the window loads. When it does, it activates the function ‘start.’ So, you went through it in reverse order, now go over it with them in the order that it actually happens.

1. Window loads, run start.
2. Start sets the button up.
3. Button is clicked, change activates
4. Change modifies the style of the p id ‘change’

Other things to discuss.

* Camel case: Javascript uses a lot of word strings in its nomenclature. In order for those words to be readable (and still be a single string of characters), it uses a capitalization schema called camel case. getElementById is a great example of this. Yes, this is case sensitive. In support of getting in that habit, encourage them to use the same naming scheme in their variables.
* Semicolons: just like with CSS, ; is your very best friend, and your code will break if you forget it.
* Debugging: if something is not working in JS, there is a very good chance that it is because of a spelling problem (syntax error). Encourage them to use cmd/cntrl f in their text editor and type out the problem area character by character. It worked to find an issue when I was designing this very demo.

Assignment:

Once you have done the demonstration, instruct students that they are to create a single index.html file. It needs to contain a button as well as an HTML element of their choosing. They should use the example as a guide. The final product should have an onload command that runs an initialization function, at least one variable, and a function that changes the CSS of the HTML element when the button is clicked. Once they have this in working order, have them submit it in Lab 13 on Webcourses.