

## Lab 2

### Introduction

This lab will have you try to implement some of the JavaScript syntax and basic DOM manipulation similar to what we discussed in class.

### Objective

Create a single HTML5 file that includes some CSS and JavaScript embedded on the page. This single file will have some basic JavaScript functionality where the scripts gather information from form fields and modify the page with basic DOM manipulation.

### Requirements

- All HTML tags and page structure should be **HTML5** compliant
- CSS and JavaScript needs to be embedded in the HTML page. I only want one HTML file turned in, it can be zipped.
- HTML is provided for you. Do not modify the HTML except where noted with comments. You may add CSS in the top of the document where the comment states if you think you need to.
- All JavaScript should go at bottom of the body where indicated with comments.
- Do not put any spaces in your file name. Name files as show in submission guidelines.
- Do not use jQuery for this lab. Pure JavaScript for DOM manipulation.
- Undergrads in 465 do not need to add functionality for the reset button unless you want. But the user should still be able to run the lab again without refreshing.

### Steps

1. Place all JavaScript in the embedded script tag at the bottom of the body section.
2. You need to create at least one function. It should execute when the run lab button is pushed. This function needs to be added to the button as click event handler with the `addEventListener` JavaScript function. This function will contain most of your code.
3. The function should get the input from the two text boxes and the color select menu. If either text box is empty you should use a JavaScript alert prompt to tell the user they must have content and stop executing the function. Use the `parseInt` function or `Number()` object constructor to convert the quantity text box to a number. Highly suggest `Number()` object vs `parse` functions. If the user didn't enter a positive number use a JavaScript alert box to tell them quantity needs to be a number and do not continue executing the function. You will be using multiple `if/else` conditional statements and `return` statements probably.
4. If everything is ok with the text input and quantity box add that quantity of output boxes to the results area. The results should be placed in the div with

id="resultsWrapper". You will need a loop for this. Each output box should consist of a div with a H2 as the headline and a P as the body text nested inside it. The H2 should have "Output" and the number it is in the loop, so like "Output 1". Don't start the numbering at zero. First element should be 1. The div and H2 should be colored based on the choice in the color drop down. You can do this multiple ways. You could add inline styles to the div and H2 or you could add a class to the div and then write css to color each item. The class and css route is probably the better choice. Some CSS for this already exists in the html template. You may add css to the template in the space provided. It can be done with the markup and css in the template only.

5. You will also need to clear the values out of the body text input and quantity input on a successful attempt.
6. If the user runs the lab again it should automatically clear the lab results first. You should be able to rerun the lab without refreshing the browser. **Do not add more boxes to the results, replace them.**

### Graduate Additional Requirements

If you are involved in **any section of 565** you need to complete the additional requirements listed here.

You will need to create the results boxes using the pure DOM manipulation functions. Do not just make a string of html and set it with innerHTML. You will need the createElement(), createTextNode(), and appendChild() methods. You may use innerHTML for other purposes.

Need to add the reset lab button as shown on the mockup. We should see another function that is bound to the click event on that button for this. Reset button should be disabled if there is not content in the results section. Enable it on successful attempt and disable after reset clicked. There is a property for this on buttons.

### README File

Not Required for this Lab

### Due Date / Late Policy

This assignment is due **Tuesday February 6, 2018 6:25 PM Chicago Time**. We will discuss this in class on February 6 so **no assignments will be accepted as of class (6:25pm) that night. No Extensions will be given or late work accepted.**

### Submission Guidelines

You must upload your submission, to the blackboard assignment by the due date. The submission must be in the following format and structure. If you do not submit your assignment exactly as specified, you will receive an immediate 5% deduction.

*Submission Format Specification:*

**YourUsername\_lab2.zip**

**|-----YourUsername\_lab2.html**

## Help

JavaScript Functions:

getElementById()

addEventListener()

createElement()

createTextNode()

appendChild()

removeChild()

alert()

parseInt() vs Number()

isNaN()

value property of a form control

className property of an element and how it relates to the class attribute

disabled attribute for form control and how to set it in JavaScript

<https://developer.mozilla.org/en-US/docs/Web/API/Node>

<https://developer.mozilla.org/en-US/docs/Web/API/Element>

<https://developer.mozilla.org/en-US/docs/Web/API/Document>

<https://developer.mozilla.org/en-US/docs/Web/API/Node/removeChild>

<https://developer.mozilla.org/en-US/docs/Web/API/Node/appendChild>

<https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement>

<https://developer.mozilla.org/en-US/docs/Web/API/Document/createTextNode>

<https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById>

<https://developer.mozilla.org/en-US/docs/Web/API/Window/alert>

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/parseInt](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/parseInt)

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number)

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/NaN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/NaN)