

**CST8285 Lab 4: Inspirational JavaScript**

**Read the entire lab instruction before starting**.

This lab is to be completed on BrightSpace any lab worksheets handed in will be discarded. Carefully follow the procedures outlined in this lab worksheet. **If at any time you are unsure or are having problems, consult your lab instructor**.

**Part I: Making the Web Dynamic**

In this lab you will take the provided JavaScript starter code and create a HTML5 compliant HTML page as well as a CSS file to format the page. You will be creating an inspirational quote generator that will use a provided JSON data structure full of inspirational quotes. Parsing JSON and XML is a fairly common task. Usually use will parse data that you have retrieved from a web API. There are APIs for everything from recipes to news feeds to database connectivity. In order to serve such content, you would need to modify your web server, which is beyond the scope of what we would like to cover at this point. In the provided JavaScript starter code, you have a constant array of quotes, this is a JSON object. The powerful thing about JSON is that it is JavaScript Object Notation, meaning that it can but used as native code. When you return such data from an API you need to deserialize it to use it as part of your codebase, we can bypass this step when we use the object already embedded in our code. Feel free to add and remove quotes, but make sure you do not disrupt the specific structure of the JSON data block.

**Part II: Building the Webpage**

1. Create a new file called inspiration.html and add the usual HTML5 boilerplate, including meta data to indicate you are the author of the code. The browser tab for your page should simply say “inspirational” and you should include a header on your page that says : “Inspirational Quotes”.
2. Create a uniquely named div for the quote and inside that div make another uniquely named div for the quote text and under that a third uniquely named div for the author. You should have two uniquely identified divs inside of the first div. This way we can use CSS for format the whole quote and also treat the quote text and author names separately. These two inner divs should be empty as we will target them with our JavaScript to add our text in between their opening and closing tags.
3. Create a centered button beneath these divs and add text such as “Get Quote” to the button. We will use the onclick attribute to manually call a JavaScript function to grab a random quote from the JSON block.
4. Create a new file called inspiration.css, link this file to your HTML page.
5. Target the body element with the desired font stack for your page. You can use a different font for each of the elements, but a good rule of thumb is to use one font for the page and only use a second one if you want to bring attention to a small portion of the page (such as a header). I chose a serif font for the title (header) , quote text, and author. I also chose a sans serif font for the button text. You can choose any fonts you like, but you must use a **full** font stack for at least the body element.
6. In the CSS file add a selector pointing to the outer div that will hold our quotation. Add a border with 15pt rounded corners to this div and give it a pleasant background colour.
7. Format the elements that are on your page so that it looks something like the image below. We will deal with the quote and author soon enough and you will likely want to tweak the CSS as you build the page.

Graphical user interface, application

Description automatically generated

**Part III: Adding the JavaScript**

1. In the HTML file add an onclick= attribute to the button element and give it the value of a JavaScript function that you will create. I called mine setInspiration().
2. Open the provided inspiration.js file and start coding after the JSON block. You are going to create a function with the same name as you used in the onclick of the button element. This function takes no parameters and will use the global constant inspirations as its pool of quotations.
3. Because arrays in JavaScript always start with an offset of 0 we can generate a random number based on the number of quotations in our JSON block. This number will be from 0 to one less than the number of quotations. You will need to know a few things to write the code that dynamically generates this number:
   1. In JavaScript an array is an object that has a member value of length that returns the number of objects in an array.
   2. To generate a random number, you use the random() function that is part of the Math object. This random() function will generate a random number from 0 to 1 that is never equal to 1. We can multiply this result by the total number of choices we want to select from and will generate a number from 0 to one less than our target number.
   3. Because random() gives us floating-point (real) numbers we use the function floor() (also in the Math object) to return the largest integer that is less than or equal to the given floating-point number. It is typical to wrap the results of random() in a floor() function.
4. After you generate a random number you can use this number as an offset to get the quote and author from the JSON block. Put these values into the innerText value of the two divs inside your quotation div using simple DOM traversal.
5. Make sure you link your script file to your HTML document.
6. Now you can test this function using a browser by pressing the button on your page. If your code is correct, then you should get a new random quote with every click of the button.
7. When you are able to generate a quote add embedded code to your HTML document that **calls** the function you created once when the page is loaded. Remember that the function needs to see the targeted elements in the DOM. This call will ensure that you always have a quote in on the page even before you click the new quote button.
8. Format the quotation (CSS) so that the text is nice and large, well centered in the outer div, and that the author’s name is in italics as well as right justified inside the outer div. You can play with the padding and margins to make the quotations look great.
9. Submit your code to an HTML validator (<https://validator.w3.org/>) and take a screen shot of a properly validated HTML page, do the same with a CSS validator (<https://jigsaw.w3.org/css-validator/>).

**Part II: Submitting your Work**

When your code is complete, make sure you zip up all the files (**lab4.zip**) created in this lab including a copy of any images that you have used in your growing website.

* **Your modified version of the starter code.**
* **css** stylesheet
* **html** webpage
* **screenshots** of both HTML and CSS validation showing successful validation of code**.**
* **any images** used