Project Instructions

Open the project files in your favorite text editor and preview them in the browser. Also, it's a smart idea to test your project in multiple browsers!

Create an array of objects to hold data for your quotes.

- In your JavaScript file, start by creating the array of quote objects. Be sure to use the name "quotes" when creating the variable that holds the array of quote objects.
- The quotes array should be accessible in what's called "the global scope", which basically just means that it's not contained within a function.

Pro Tip: Remember, arrays are defined with brackets, [ ], and objects are defined with curly braces, { }.

Add Properties to the quote objects.

Each quote object in the quotes array should have the following properties:

- Quote – a string containing the text of the quote that will be displayed on the page.
- Source – a string containing the creator of the quote. For example, "Mark Twain".
- Citation – optional - a string identifying where the quote comes from, like a speech, publication or a movie. If there is no known publication, do not include this property on the object.
- Year – optional - a number identifying the year of the quote. If there is no known year, then do not include this property on the object.

Pro Tip: Remember that an object's properties are defined in key/value pairs, like so:

⇨ Quote: "There's no place like home."

NOTE: To get the most out of this project, it would be best to include at least one year and citation property in your array of quote objects. This will give you the chance to get some good practice working with conditionals when you create the printQuote() function that prints the quotes to the screen.

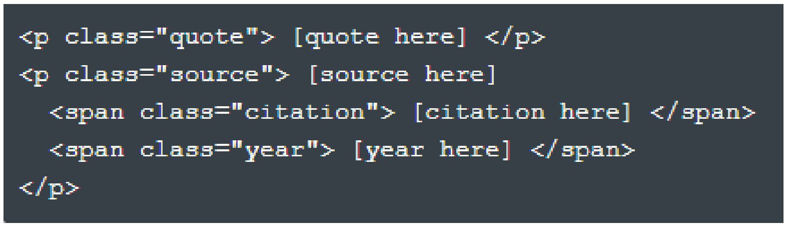
Create the getRandomQuote function.

- Be sure to name the function getRandomQuote.
- The function should take in one parameter, that you can rightly call, array, since the argument that will later get passed into this function when it is invoked will be the array of quotes.
- The body of the function should select and return a random quote object from the quotes array. Remember the method for finding a random number? If not, that's okay. Check the resources list in these instructions or revisit the material in the unit. Alternatively, try using Google to see if you can find what you need on the web. Even professional developers use Google several times a day, so it's good to start developing your Google skills now.

Pro Tip: Remember, we target and select items in an array with "box notation". So, array[0] will select the first item in an array, because, as you'll likely remember, arrays always begin with a first index of zero.

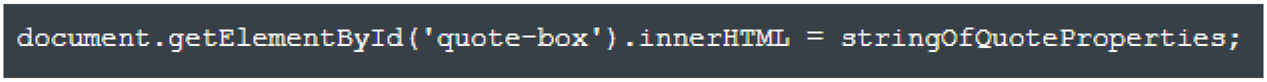
Create the printQuote function.

- Be sure to name the function printQuote.
- In the body of this function:
    - getRandomQuote should be called on the array of quote obejcts, and the result stored in a variable.
    - A new empty string should be created and stored in a variable.
    - The required quote object properties, and the optional quote properties if they exist, should be appropriately concatenated to the new string you created using the following HTML template:



```
<p class="quote"> [quote here] </p>
<p class="source"> [source here]
  <span class="citation"> [citation here] </span>
  <span class="year"> [year here] </span>
</p>
```

- The span elements that hold the citation and year properties should only be created if the current quote object has those properties. This is where your conditionals will come in handy.
- Finally the printQuote function should display the completed HTML string to the page by targeting the necessary DOM element and using the appropriate JavaScript method, like so:



```
document.getElementById('quote-box').innerHTML = stringOfQuoteProperties;
```

Pro Tip: Remember that we target and select object properties with "dot notation". So to reference the citation in a quote object, we would do something like this: quote.citation.

Add code comments.

- Replace the code comments in the file with your own code comments.
- The key to creating good code comments is to keep them brief, but clear, so that your fellow developers can get an idea of what's going on in your code at a glance, and without having to review every single line of code. Generally, good code comments are placed above important code blocks to explain what they do.

NOTE: Sometimes just getting started is the hardest part.

- It's not uncommon to feel overwhelmed and confused when beginning to build a project. If you feel this way, try not to get too focused on seeing the project as a whole. Instead, just take it one small piece at a time. After familiarizing yourself with the instructions, start by downloading the source files, and creating a GitHub repo to store them. That is always a great place to start. Then just start tackling the project one small step at a time. Remember, your first attempt isn't likely to be perfect, and it doesn't have to be. As coders, we focus first on creating something that works. And then we refactor and optimize on later iterations.

- Practice your Google skills by finding different ways to ask the questions you have, paying close attention to the sort of results you get back depending on how your questions are worded.

EXTRA CREDIT

Add more properties to the quote object.

- For example, a tags property could include a list of "tags" like "humor", "business", or "politics" to categorize each quote.
- Use the new properties you've created to conditionally add to the string that gets printed to the screen.

Random background color.

- When the quote changes, randomly change the background color of the page.

Auto-refresh the quote.

- After a set amount of time, like 20 to 30 seconds, print a new quote to the page. Timing methods like setInterval() and setTimeout() can be helpful here. see the links in the “Additional Resources” section.