* Open the project files in your favorite text editor and preview them in the browser

Pro Tip: Text Editors

* + If you have questions about selecting a good text editor or popular editor extensions, you can locate multiple recommendations and helpful links by reaching out to your Slack team.
* Create an array of objects to store the data for your quotes
  + In your js/script.js file, start by creating a variable named quotes and setting it equal to an empty array.
  + Add a minimum of five empty objects to the quotes array.

JavaScript Reminders:

* + Arrays are defined with brackets: []
  + Objects are defined with curly braces: {}
  + The array of quote objects should be accessible in "the global scope", which means it's written outside of any functions rather than being contained in one.
* Add data to your quote objects
  + Add the following properties to each quote object:
    - quote - a string containing the text of a quote.
    - source - a string containing the origin or source of the quote. For example: "Mark Twain", "Traditional Irish proverb”, "Anonymous".
  + Add the following properties to at least one quote object:
    - citation - a string identifying the venue of the quote. For example, a speech, publication, movie.
    - year - a number identifying the year of the quote.

JavaScript Reminders:

* + An object’s key/value pairs are separated by a comma.
  + An object's key/value pairs are defined like so:

quote: "There's no place like home."

* Testing your array of quote objects
  + Below the array of quote objects, add console.log(quotes);.
  + Shortcut to open the Chrome console: Ctrl + Shift + J (on Windows) or Ctrl + Option + J (on Mac)
  + Check out your objects in the console by clicking on them in the console to open them up and inspect their properties.
  + Consider deleting your log statements as you no longer need them because they can quickly accumulate and clutter up your code, making it harder to manage.
  + Project Warm Up: Printing and logging values can help you better understand your code, the state of your app, and ensure that the code you've written is what you think it is. For some helpful practice, check out the project Warm Up [Printing and Logging Values](https://teamtreehouse.com/library/fsjs-project-warm-up-printing-and-logging-values).
* Create the getRandomQuote function
  + Create a function named getRandomQuote.
  + In the function body:
    - Create a variable to store a random number between zero and the length of the quotes array. See the Project Resources section for a link to the related course video.
    - Use the quotes array, box notation and the random number variable to return a random quote.
  + Use console.log(getRandomQuote()); immediately after the getRandomQuote function definition to test and display the random quote in the console.
  + Project Warm Up: Dealing with functions, arrays and random numbers can be confusing at first. For some helpful practice, check out the project Warm Up [Random Array Index](https://teamtreehouse.com/library/fsjs-project-warm-up-random-array-index).

JavaScript Reminders:

* + Array indexes start at zero.
  + Box notation uses brackets to target an array index like so, array[0].
* Create the printQuote function
  + Create a function named printQuote.
  + In the function body:
    - Create a variable and set it equal to getRandomQuote(). This variable will hold the random quote object from the quotes array.
    - Avoid targeting the quote, source, citation and year sections of the quote HTML, and updating them separately. Instead, assemble a string of all the HTML and quote sections, and print the entire quote all at once.
    - Create a variable and set it equal to an empty string, ''. This variable will eventually hold the entire string of HTML and quote properties that will get printed to the screen. Be sure to use single quotes for your strings to avoid quote mark confusion, since HTML always uses double quotes for element attributes like class names.
    - Using the random quote variable, and the example HTML template below, concatenate the following strings to the empty string variable.
* <p class="quote"> [quote here] </p>
* <p class="source"> [source here]
* <span class="citation"> [citation here] </span>
* <span class="year"> [year here] </span>
* </p>
  + Start with a string that contains:
    - An opening p tag with a class equal to “quotes”.
    - The quote property of the quote object in the random quote variable.
    - A closing p tag.
  + Next, add a string that contains:
    - An opening p tag with a class equal to “source”.
    - The source property of the quote object in the random quote variable.
    - Do not add a closing p tag for this one yet.
  + Use an if statement to test if the quote object has a citation property, and if so, concatenate a string that contains:
    - An opening span tag with a class equal to “citation”.
    - The citation property of the quote object in the random quote variable.
    - A closing span tag.
  + Use an if statement to test if the quote object has a year property, and if so, concatenate a string that contains:
    - An opening span tag with a class equal to “year”.
    - The year property of the quote object in the random quote variable.
    - A closing span tag.
  + Now concatenate a closing p tag to the HTML string.
  + A console.log(); with the HTML string variable passed in can be used here to test and display the HTML string in the console. But the “Show Another Quote” button will need to be clicked to call this function.
  + Lastly, target the div with the id of quote-box, and set its innerHTML equal to the HTML string that’s been assembled.
  + Now when you click the “Show Another Quote” button, a new quote should be displayed on the page, thanks to the event listener that’s been provided for you at the end of the js/script.js file.
  + Project Warm Up: Using conditionals to test if objects or elements exist before trying to do something with them is a common developer task. For some helpful practice, check out the project Warm Up [Conditional String](https://teamtreehouse.com/library/fsjs-project-warm-up-conditional-string).

JavaScript Reminders:

* + Dot notation is used to target a specific property of an object, like so, object.property.
  + To check if an object property exists, the condition needs only be, (object.property).
  + Template literals can be used for the HTMl string if you’re comfortable with that approach.
* 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, without having to review every line of your code.
* Review the "How you'll be graded" section
  + Check out the "How you'll be graded" section, located above, next to the instructions tab, just below the project title. This section lists specifically what will be considered and checked when your project is being reviewed, and your project grade is being determined.
* Cross-Browser consistency:
  + Google Chrome has become the default development browser for most developers. With such a selection of browsers for users to choose from, it's a good idea to get in the habit of testing your projects in all modern browsers.
* Quality Assurance and the Student Project Submission Checklist
  + Take the time for thorough quality assurance testing. Before you submit your project for review, please check off all of the items on the [Student Project Submission Checklist](http://treehouse-techdegree.s3.amazonaws.com/Student-Project-Submission-Checklist.pdf).
  + Share your project in Slack in the #review-my-project channel to get a preliminary review and catch anything you might have missed.

NOTE: What you submit is what will get reviewed.

* + When you submit your project, a snapshot is taken of your repository, and that is what the reviewer will see. Consequently, the reviewer won't see any changes you make to your repo after you submit the project.

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 to conditionally add to the HTML 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. A timing method like setInterval() is helpful here. Avoid using the setTimeout function for this step. See the links in the “Additional Resources” section.