**Step 1**

Your project starts with a basic HTML container and some corresponding CSS. Your first task will be to programmatically generate the cells for your spreadsheet.

The global window object represents the browser window (or tab). It has an onload property which allows you to define behavior when the window has loaded the entire page, including stylesheets and scripts.

Start by setting the onload property of window to an arrow function with no parameters. In the function, declare a container variable and assign it the value of getting the element by the id of container.



Check Your Code (Ctrl + Enter)

**Step 2**

Functions are ideal for reusable logic. When a function itself needs to reuse logic, you can declare a nested function to handle that logic. Here is an example of a nested function:

Example Code

const outer = () => {

const inner = () => {

};

};

Declare a nested createLabel function using arrow syntax. It should take a name parameter.

Currently your range function returns an array with the correct length, but all of the values are the value of start. To fix this, chain the .map() method to your .fill() method.

Pass the .map() method a callback which takes element and index as parameters and returns the sum of those parameters.

Your range function expects numbers, but your start and end values will be strings (specifically, they will be single characters such as A).

Convert your start and end values in your range() call to numbers by using the .charCodeAt() method on them, passing the number 0 as the argument to that method.

# Step 12

range() will return an array of numbers, which you need to convert back into characters. Chain the .map() method to your range() call.

Pass a callback function that takes code as the parameter and implicitly returns the value of passing code to the String.fromCharCode() method.

# Step 13

Now that your helper functions are complete, back in your onload event handler you should declare a letters variable. Assign it the result of calling charRange() with the letters A and J as arguments.

# Step 14

Now call the .forEach() method of your letters array, and pass your createLabel function reference as the callback.

You should see some letters appear across the top of your spreadsheet.

# Step 15

Remember that range() returns an array, so you can chain array methods directly to the function call.

Call range() with 1 and 99 as the arguments, and chain the .forEach() method. Pass the .forEach() method an empty callback which takes number as the parameter.

# Step 16

In your callback, you will need to make two function calls. Start by calling createLabel() and pass number as the argument. You should see some numbers appear in your spreadsheet.

Then call the .forEach() method on your letters array. Pass an empty callback function which takes a letter parameter.

# Step 17

Now in your nested .forEach() call, declare an input variable. Use the .createElement() method of the document object to create an input element. Set the type attribute to text and the id attribute to letter + number.

**Step 18**

In earlier projects you learned about the setAttribute method. Another way to update an attribute in JavaScript is to use the following syntax:

Example Code

el.attribute = value;

The property names for hyphenated HTML attribute values, such as aria-label, follow camel case, becoming ariaLabel.

Example Code

el.ariaLabel = "Aria Label Value";

Set the aria-label attribute for the input element to the same value as the id attribute.

# Step 19Passed

Append the input element to your container element as a child.

You should now be able to see the cells of your spreadsheet.

# Step 20

Most spreadsheet programs include built-in functions for calculation.

Declare a sum function that takes a nums parameter, which will be an array of numbers. It should return the result of calling reduce on the array to sum all of the numbers.

Declare an isEven function, which takes a num parameter and returns true if the number is even, and false otherwise. Use the modulo operator % to determine if a number is even or odd.

# Step 22

Declare an average function which takes an array of numbers as the nums parameter. It should return the average of all the numbers in the array.

The average can be calculated by dividing the sum of all the numbers in the array by the length of the array. Remember that you have a sum function you can use.

Your next function will calculate the median value of an array of numbers. Start by declaring a median arrow function that takes a nums parameter.

In the function, declare a sorted variable and assign it the value of sorting a copy of the nums array.

You should use the slice() method for creating a shallow copy of the array.