PROMINEO TECH

Intro to JavaScript Week 3 Coding Assignment

Points possible: 75

URL to Your GitHub Repository:

https://github.com/anagalacticRuby/Promineo-Week3

URL to video demonstration of code:

https://youtu.be/xMSUuCS9tvE

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

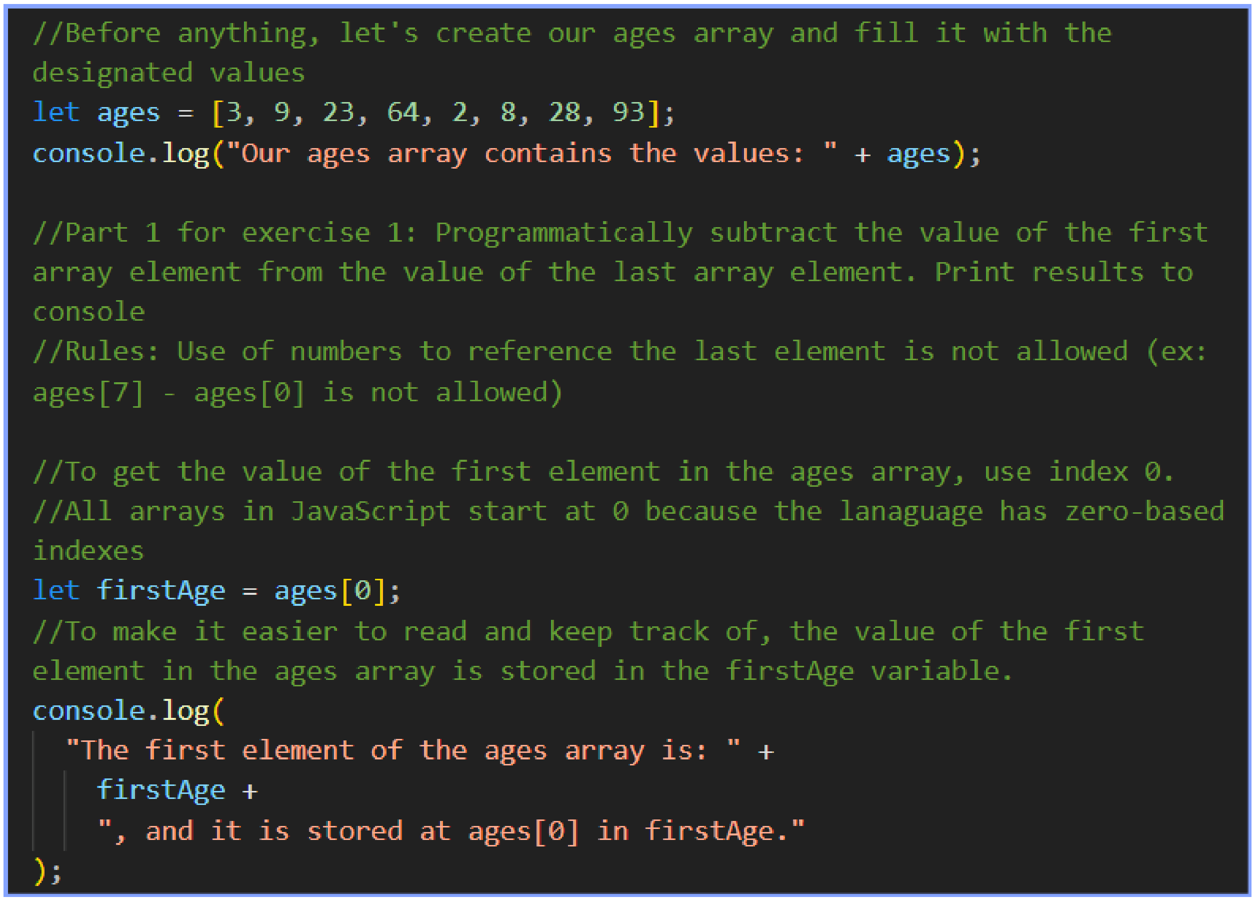
1. Create an array called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
    a. Programmatically subtract the value of the first element in the array from the value in the last element of the array (do not use numbers to reference the last element, find it programmatically, ages[7] – ages[0] is not allowed). Print the result to the console.
    b. Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
    c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.
2. Create an array called names that contains the following values: 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob'.
    a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.
    b. Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and print the result to the console.
3. How do you access the last element of any array?
4. How do you access the first element of any array?
5. Create a new array called nameLengths. Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array.

For example:

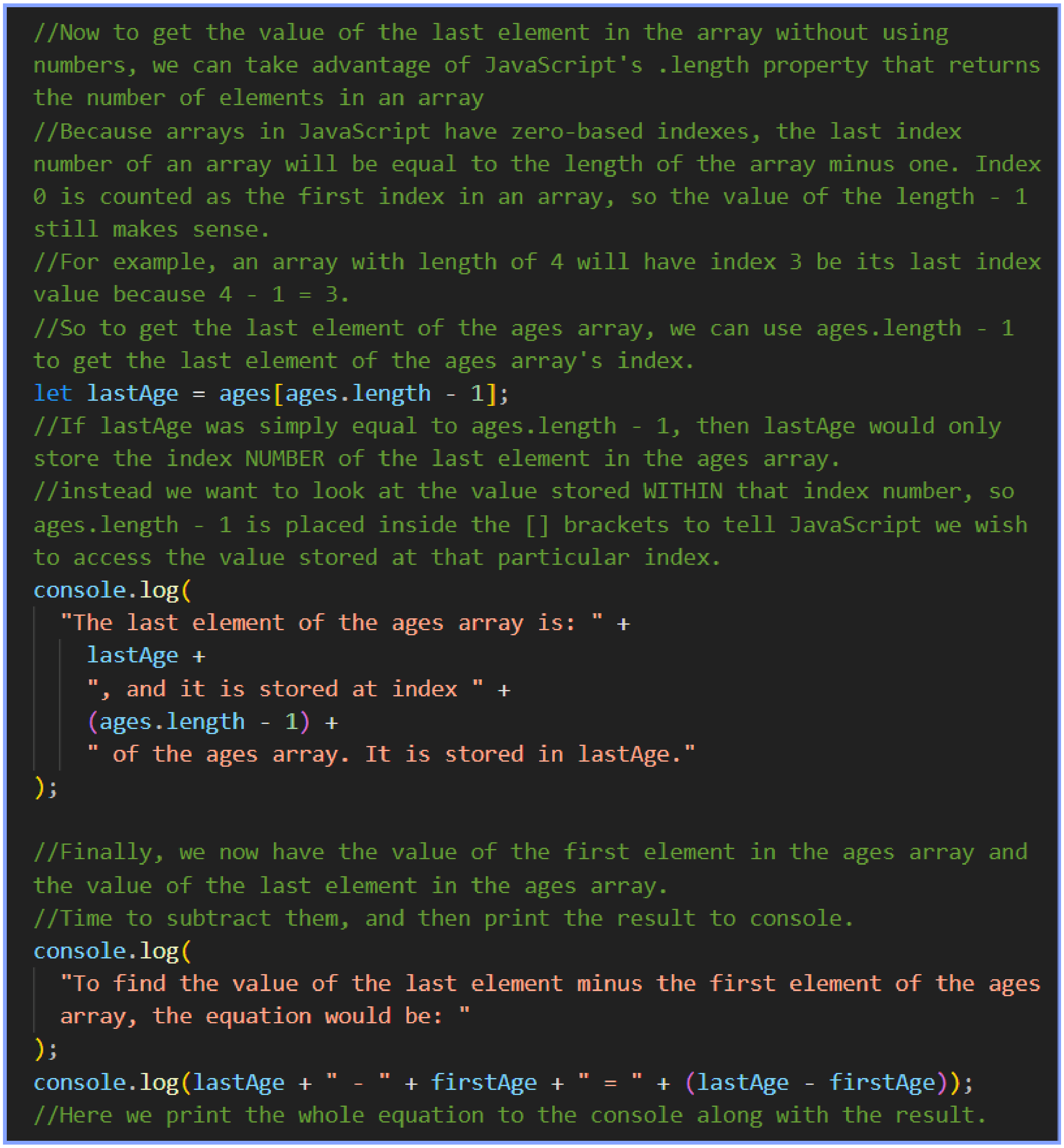
```
namesArray = ["Kelly", "Sam", "Kate"] //given this array
nameLengths = [5, 3, 4] //create this new array
```

6. Write a loop to iterate over the nameLengths array and calculate the sum of all the elements in the array. Print the result to the console.
7. Write a function that takes two parameters, word and n, as arguments and returns the word concatenated to itself n number of times. (i.e. if I pass in ‘Hello’ and 3, I would expect the function to return ‘HelloHelloHello’).
8. Write a function that takes two parameters, firstName and lastName, and returns a full name (the full name should be the first and the last name separated by a space).
9. Write a function that takes an array of numbers and returns true if the sum of all the numbers in the array is greater than 100.
10. Write a function that takes an array of numbers and returns the average of all the elements in the array.
11. Write a function that takes two arrays of numbers and returns true if the average of the elements in the first array is greater than the average of the elements in the second array.
12. Write a function called willBuyDrink that takes a boolean isHotOutside, and a number moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
13. Create a function of your own that solves a problem. In comments, write what the function does and why you created it.

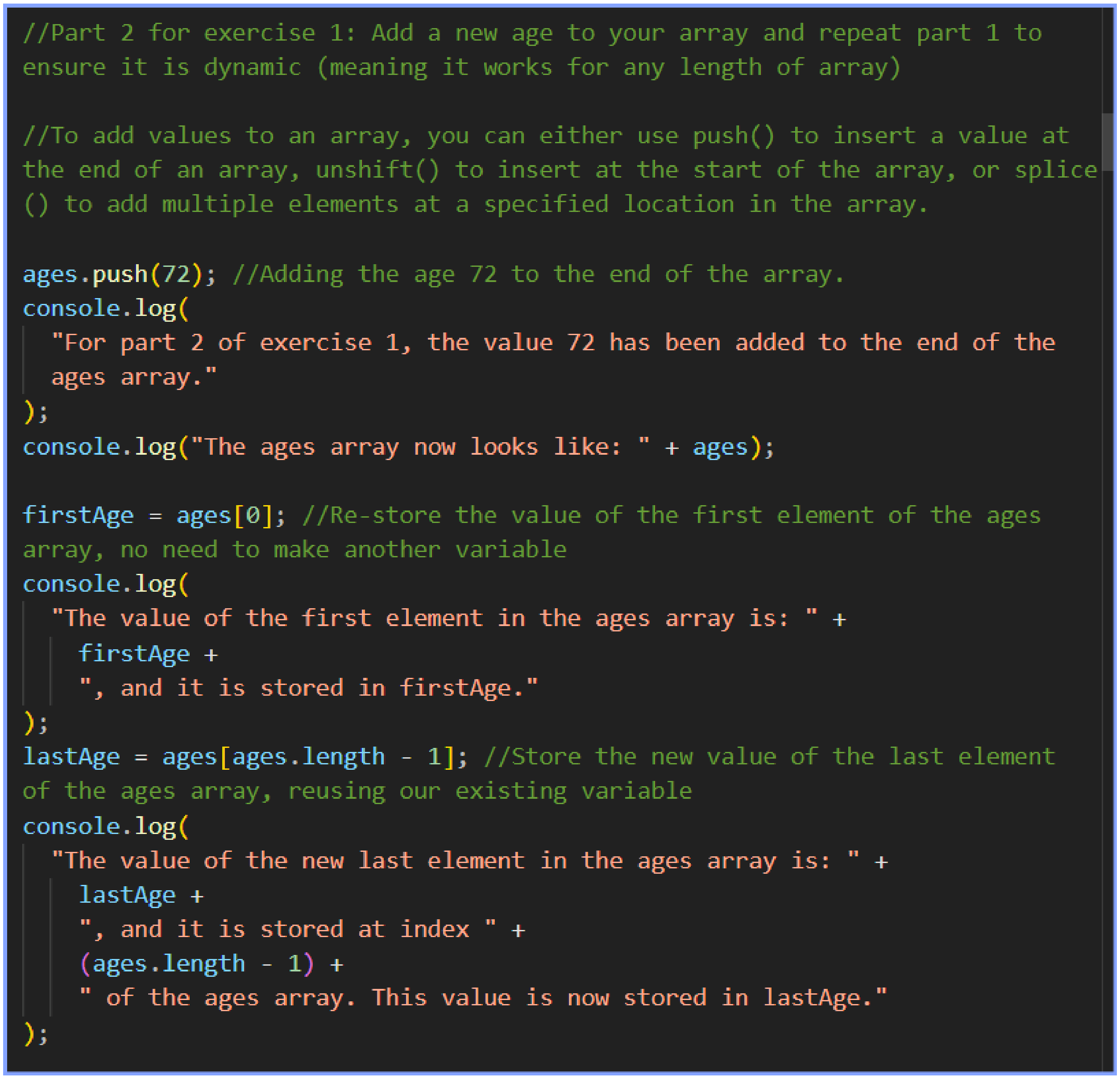
Screenshots of Code:



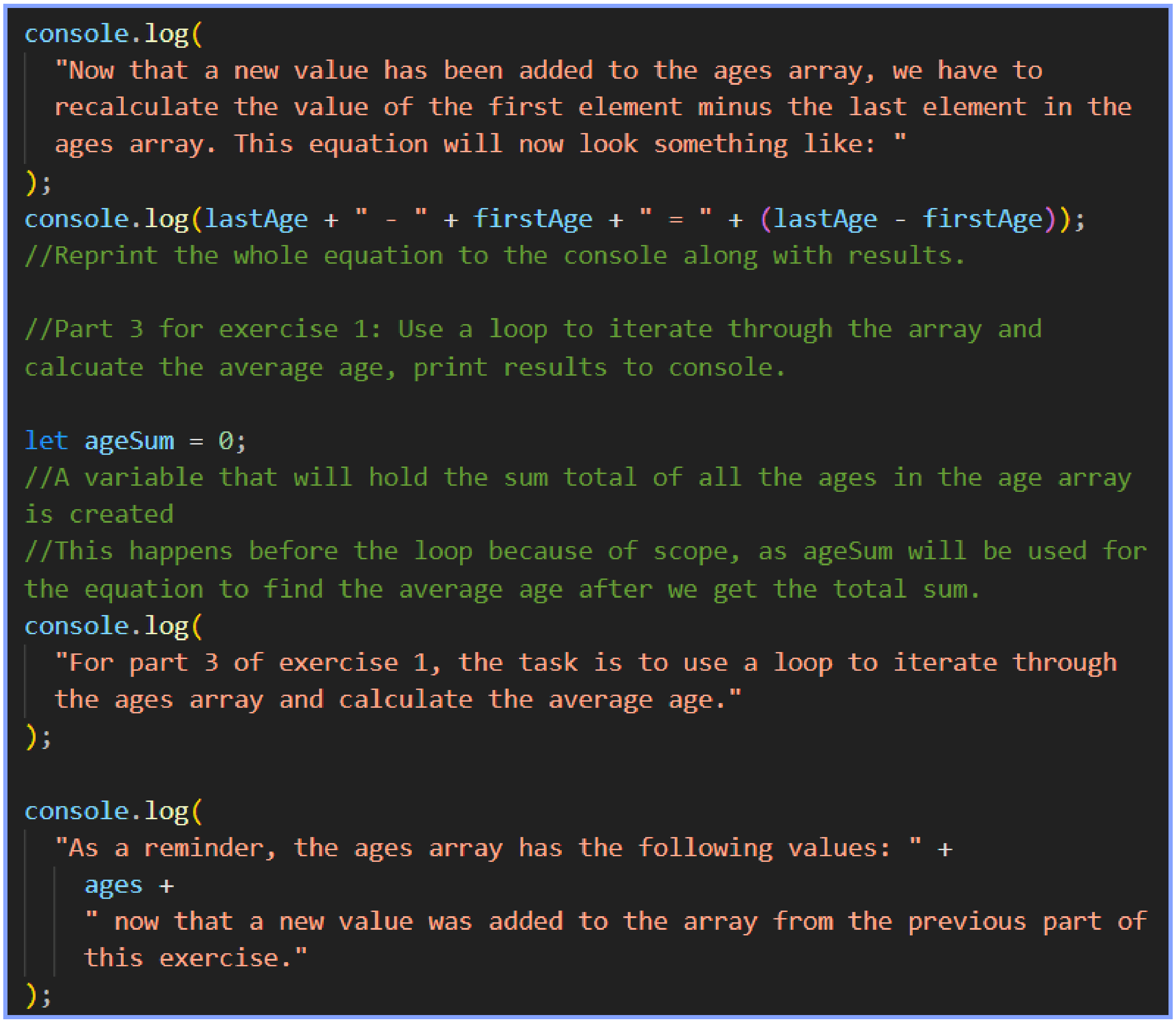
```javascript
//Before anything, let's create our ages array and fill it with the designated values
let ages = [3, 9, 23, 64, 2, 8, 28, 93];
console.log("Our ages array contains the values: " + ages);

//Part 1 for exercise 1: Programmatically subtract the value of the first array element from the value of the last array element. Print results to console
//Rules: Use of numbers to reference the last element is not allowed (ex: ages[7] - ages[0] is not allowed)

//To get the value of the first element in the ages array, use index 0.
//All arrays in JavaScript start at 0 because the lanaguage has zero-based indexes
let firstAge = ages[0];
//To make it easier to read and keep track of, the value of the first element in the ages array is stored in the firstAge variable.
console.log(
  "The first element of the ages array is: " +
    firstAge +
    ", and it is stored at ages[0] in firstAge."
);
```



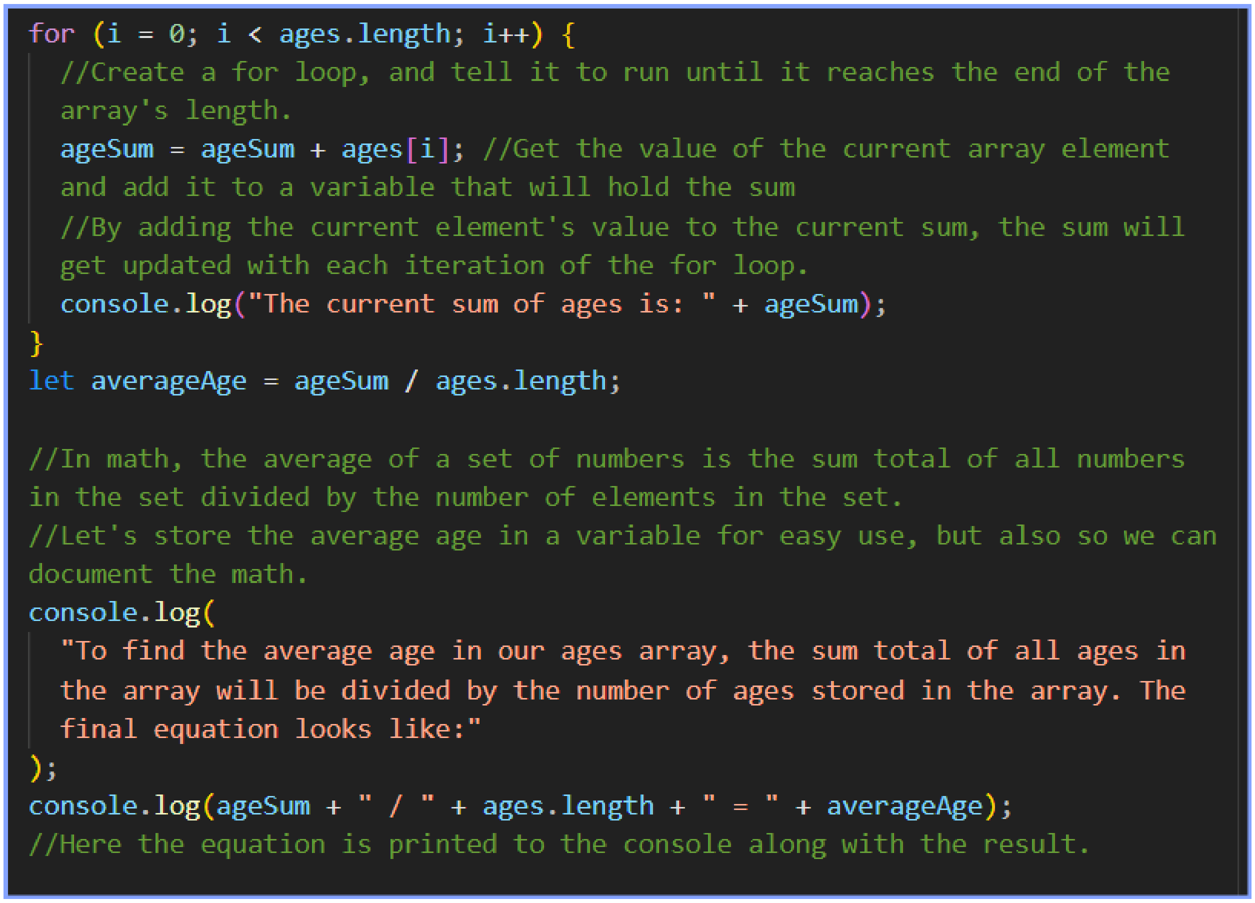
```
//Now to get the value of the last element in the array without using numbers, we can take advantage of JavaScript's .length property that returns the number of elements in an array
//Because arrays in JavaScript have zero-based indexes, the last index number of an array will be equal to the length of the array minus one. Index 0 is counted as the first index in an array, so the value of the length - 1 still makes sense.
//For example, an array with length of 4 will have index 3 be its last index value because 4 - 1 = 3.
//So to get the last element of the ages array, we can use ages.length - 1 to get the last element of the ages array's index.
let lastAge = ages[ages.length - 1];
//If lastAge was simply equal to ages.length - 1, then lastAge would only store the index NUMBER of the last element in the ages array.
//instead we want to look at the value stored WITHIN that index number, so ages.length - 1 is placed inside the [] brackets to tell JavaScript we wish to access the value stored at that particular index.
console.log(
  "The last element of the ages array is: " +
    lastAge +
    ", and it is stored at index " +
    (ages.length - 1) +
    " of the ages array. It is stored in lastAge."
);

//Finally, we now have the value of the first element in the ages array and the value of the last element in the ages array.
//Time to subtract them, and then print the result to console.
console.log(
  "To find the value of the last element minus the first element of the ages array, the equation would be: "
);
console.log(lastAge + " - " + firstAge + " = " + (lastAge - firstAge));
//Here we print the whole equation to the console along with the result.
```



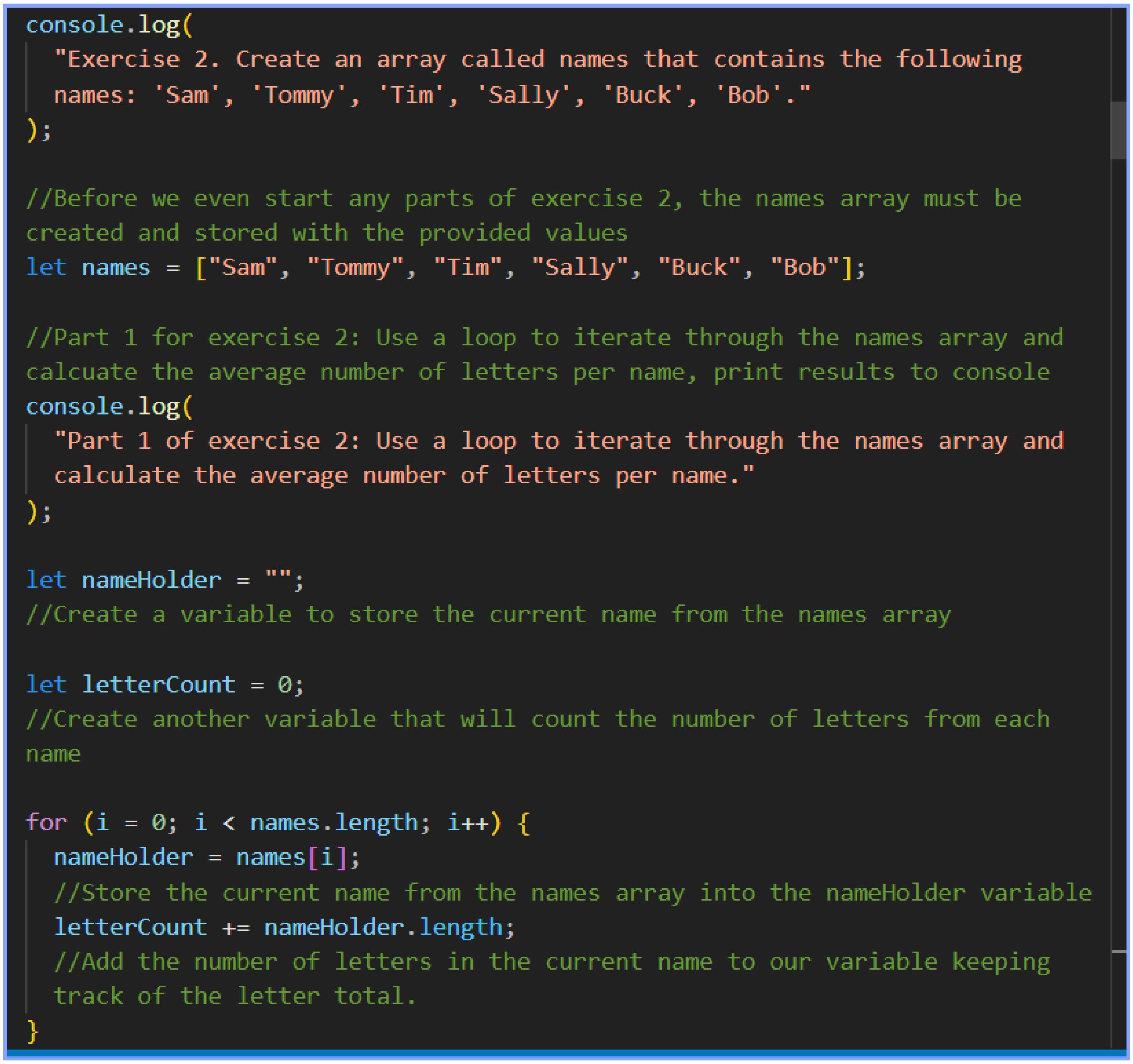
```
//Part 2 for exercise 1: Add a new age to your array and repeat part 1 to
ensure it is dynamic (meaning it works for any length of array)

//To add values to an array, you can either use push() to insert a value at
the end of an array, unshift() to insert at the start of the array, or splice
() to add multiple elements at a specified location in the array.

ages.push(72); //Adding the age 72 to the end of the array.
console.log(
  "For part 2 of exercise 1, the value 72 has been added to the end of the
  ages array."
);
console.log("The ages array now looks like: " + ages);

firstAge = ages[0]; //Re-store the value of the first element of the ages
array, no need to make another variable
console.log(
  "The value of the first element in the ages array is: " +
    firstAge +
    ", and it is stored in firstAge."
);
lastAge = ages[ages.length - 1]; //Store the new value of the last element
of the ages array, reusing our existing variable
console.log(
  "The value of the new last element in the ages array is: " +
    lastAge +
    ", and it is stored at index " +
    (ages.length - 1) +
    " of the ages array. This value is now stored in lastAge."
);
```



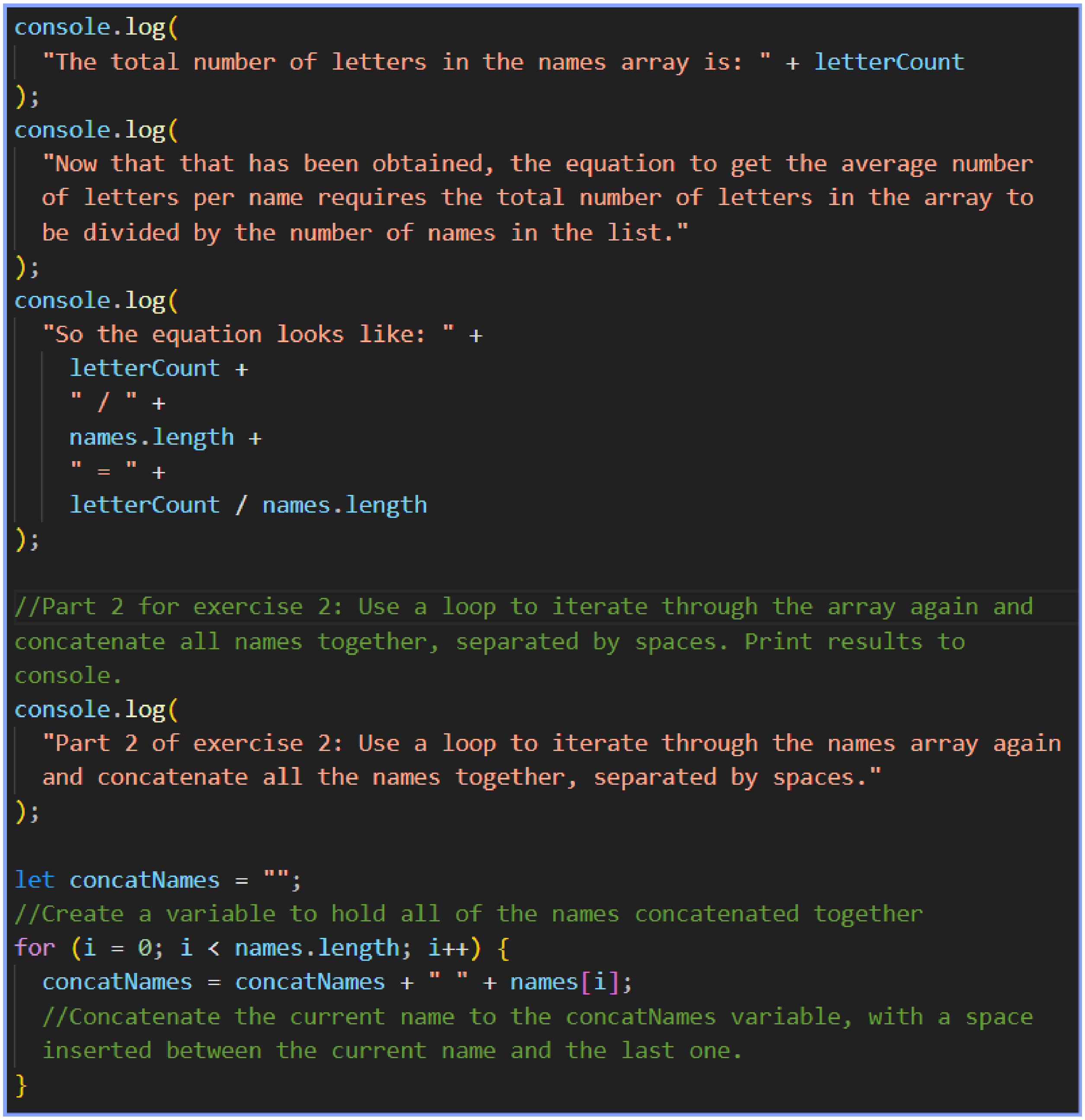
```javascript
console.log(
  "Now that a new value has been added to the ages array, we have to
  recalculate the value of the first element minus the last element in the
  ages array. This equation will now look something like: "
);
console.log(lastAge + " - " + firstAge + " = " + (lastAge - firstAge));
//Reprint the whole equation to the console along with results.

//Part 3 for exercise 1: Use a loop to iterate through the array and
calcuate the average age, print results to console.

let ageSum = 0;
//A variable that will hold the sum total of all the ages in the age array
is created
//This happens before the loop because of scope, as ageSum will be used for
the equation to find the average age after we get the total sum.
console.log(
  "For part 3 of exercise 1, the task is to use a loop to iterate through
  the ages array and calculate the average age."
);

console.log(
  "As a reminder, the ages array has the following values: " +
    ages +
    " now that a new value was added to the array from the previous part of
    this exercise."
);
```



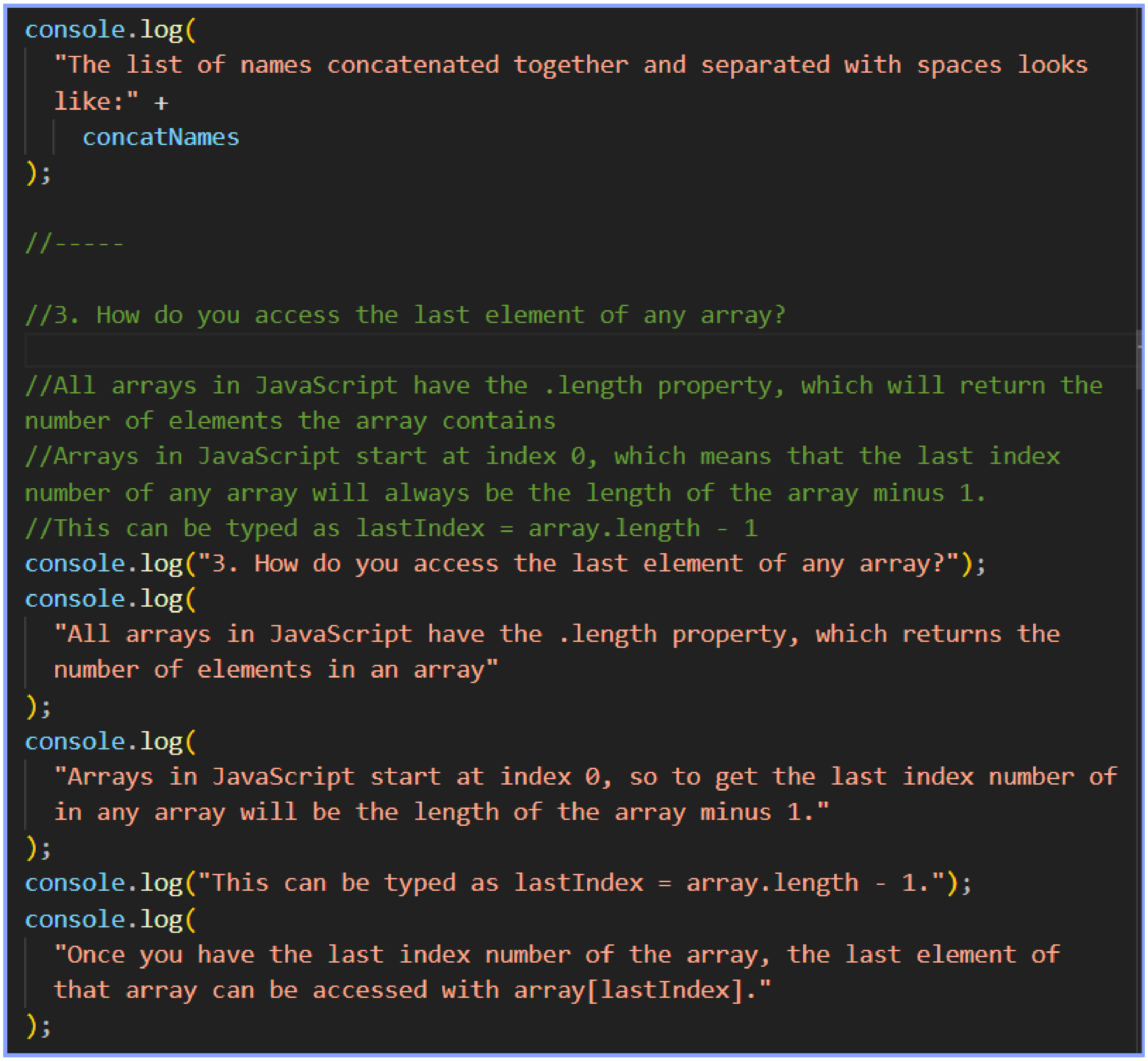
```
for (i = 0; i < ages.length; i++) {
  //Create a for loop, and tell it to run until it reaches the end of the
  array's length.
  ageSum = ageSum + ages[i]; //Get the value of the current array element
  and add it to a variable that will hold the sum
  //By adding the current element's value to the current sum, the sum will
  get updated with each iteration of the for loop.
  console.log("The current sum of ages is: " + ageSum);
}
let averageAge = ageSum / ages.length;

//In math, the average of a set of numbers is the sum total of all numbers
in the set divided by the number of elements in the set.
//Let's store the average age in a variable for easy use, but also so we can
document the math.
console.log(
  "To find the average age in our ages array, the sum total of all ages in
  the array will be divided by the number of ages stored in the array. The
  final equation looks like:"
);
console.log(ageSum + " / " + ages.length + " = " + averageAge);
//Here the equation is printed to the console along with the result.
```



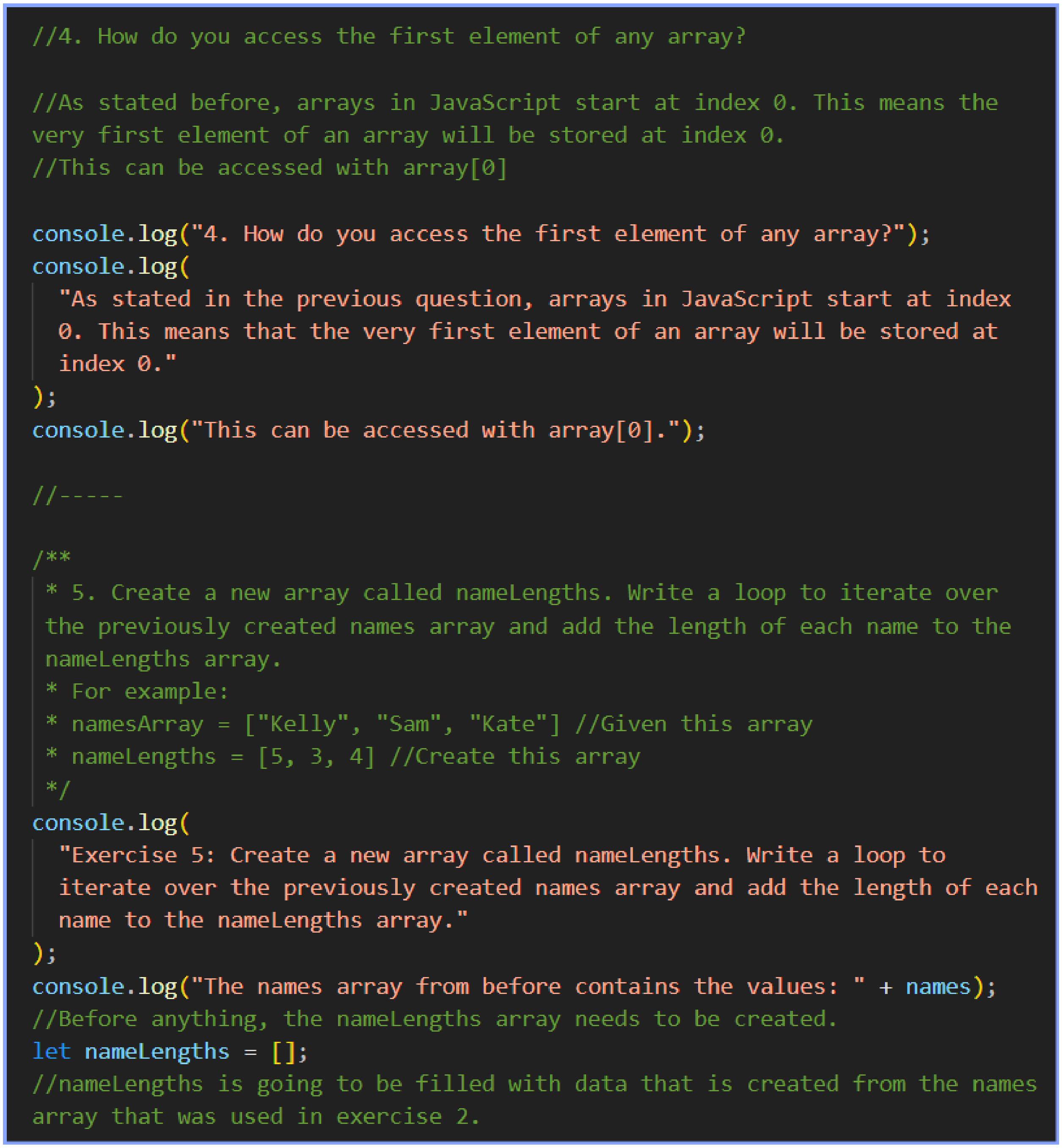
```
console.log(
  "Exercise 2. Create an array called names that contains the following
  names: 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob'."
);

//Before we even start any parts of exercise 2, the names array must be
created and stored with the provided values
let names = ["Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"];

//Part 1 for exercise 2: Use a loop to iterate through the names array and
calcuate the average number of letters per name, print results to console
console.log(
  "Part 1 of exercise 2: Use a loop to iterate through the names array and
  calculate the average number of letters per name."
);

let nameHolder = "";
//Create a variable to store the current name from the names array

let letterCount = 0;
//Create another variable that will count the number of letters from each
name

for (i = 0; i < names.length; i++) {
  nameHolder = names[i];
  //Store the current name from the names array into the nameHolder variable
  letterCount += nameHolder.length;
  //Add the number of letters in the current name to our variable keeping
  track of the letter total.
}
```



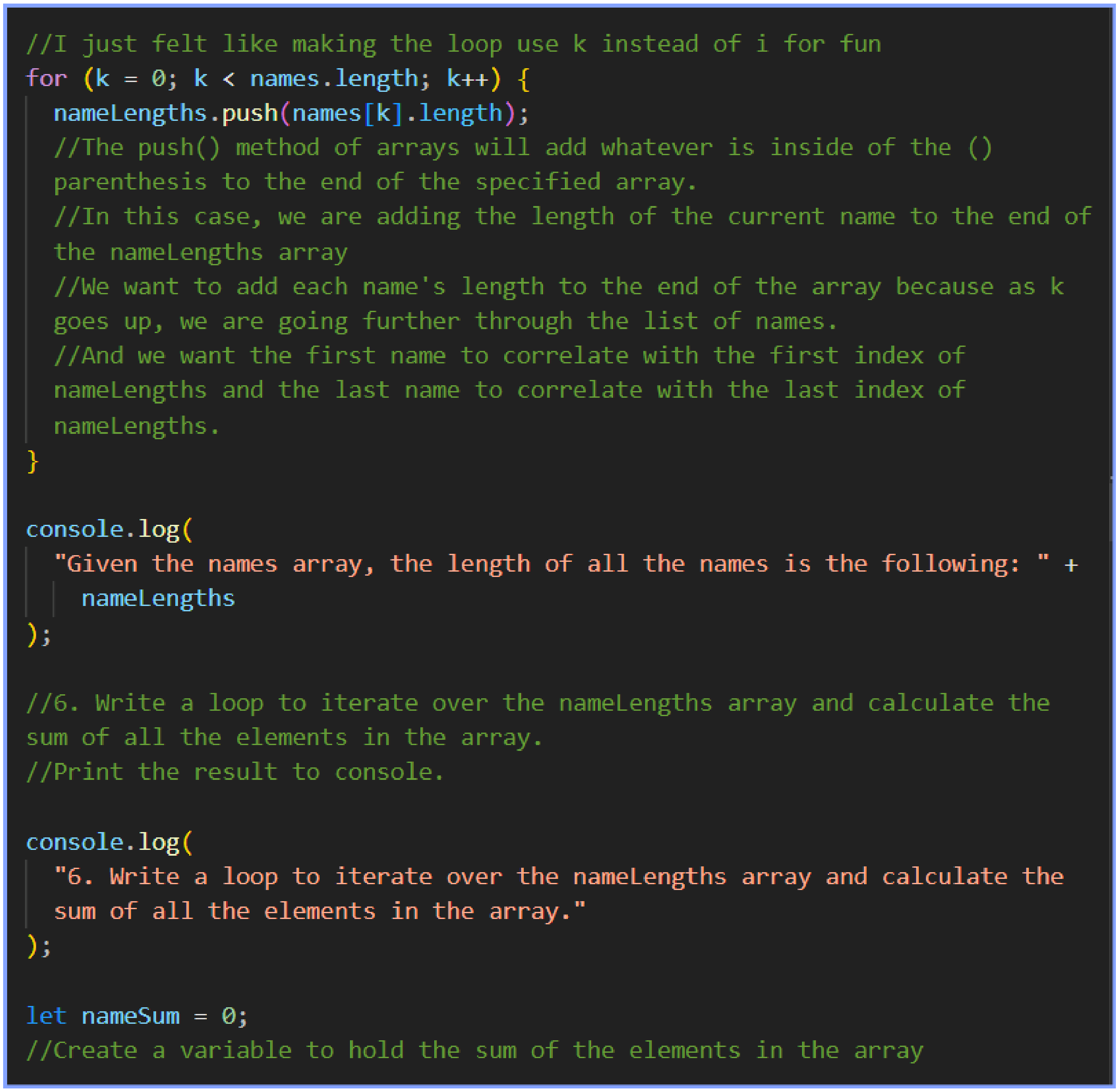
```
console.log(
  "The total number of letters in the names array is: " + letterCount
);
console.log(
  "Now that that has been obtained, the equation to get the average number
  of letters per name requires the total number of letters in the array to
  be divided by the number of names in the list."
);
console.log(
  "So the equation looks like: " +
    letterCount +
    " / " +
    names.length +
    " = " +
    letterCount / names.length
);

//Part 2 for exercise 2: Use a loop to iterate through the array again and
concatenate all names together, separated by spaces. Print results to
console.
console.log(
  "Part 2 of exercise 2: Use a loop to iterate through the names array again
  and concatenate all the names together, separated by spaces."
);

let concatNames = "";
//Create a variable to hold all of the names concatenated together
for (i = 0; i < names.length; i++) {
  concatNames = concatNames + " " + names[i];
  //Concatenate the current name to the concatNames variable, with a space
  inserted between the current name and the last one.
}
```



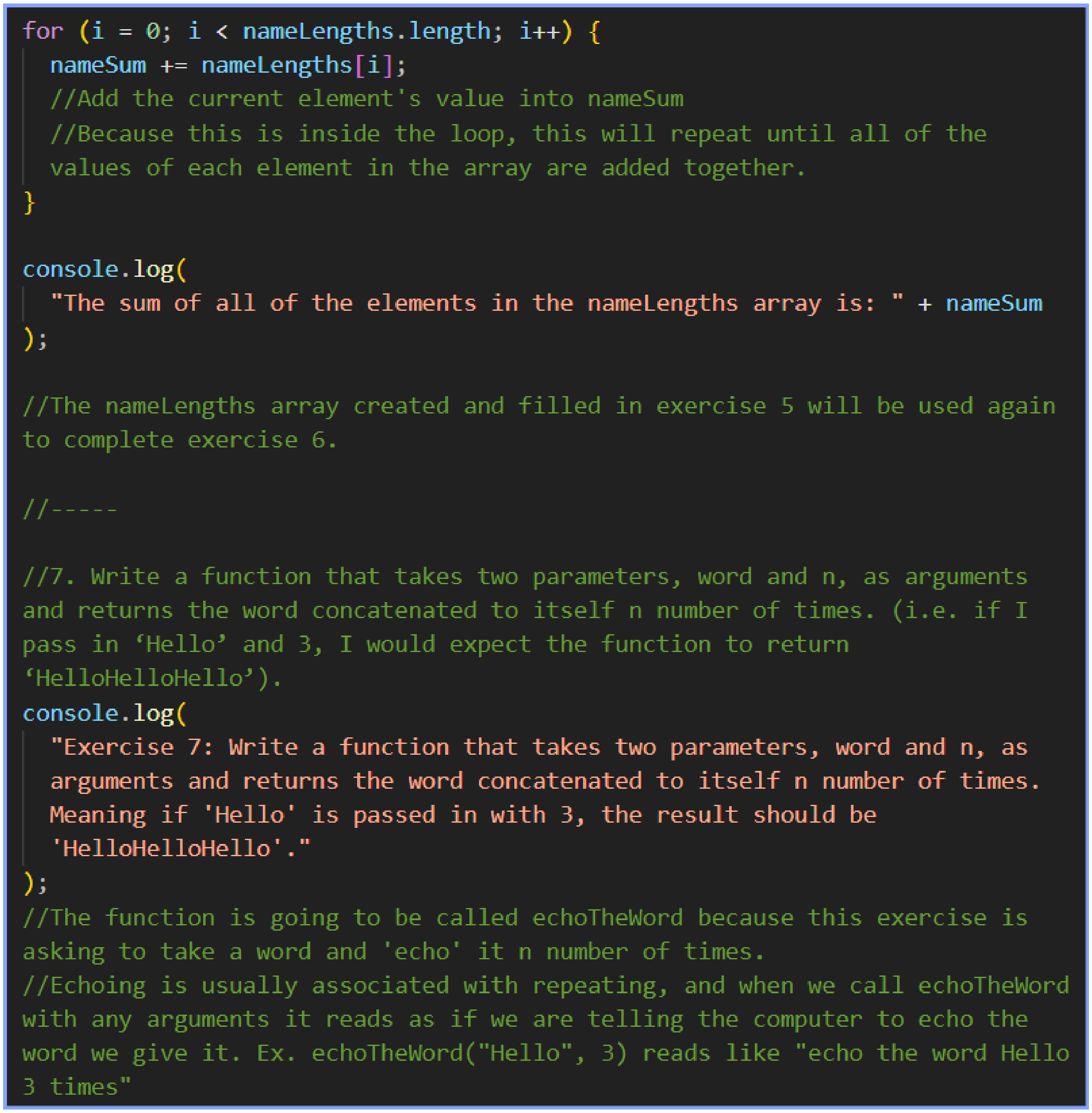
```
console.log(
  "The list of names concatenated together and separated with spaces looks
  like:" +
    concatNames
);

//-----

//3. How do you access the last element of any array?

//All arrays in JavaScript have the .length property, which will return the
number of elements the array contains
//Arrays in JavaScript start at index 0, which means that the last index
number of any array will always be the length of the array minus 1.
//This can be typed as lastIndex = array.length - 1
console.log("3. How do you access the last element of any array?");
console.log(
  "All arrays in JavaScript have the .length property, which returns the
  number of elements in an array"
);
console.log(
  "Arrays in JavaScript start at index 0, so to get the last index number of
  in any array will be the length of the array minus 1."
);
console.log("This can be typed as lastIndex = array.length - 1.");
console.log(
  "Once you have the last index number of the array, the last element of
  that array can be accessed with array[lastIndex]."
);
```



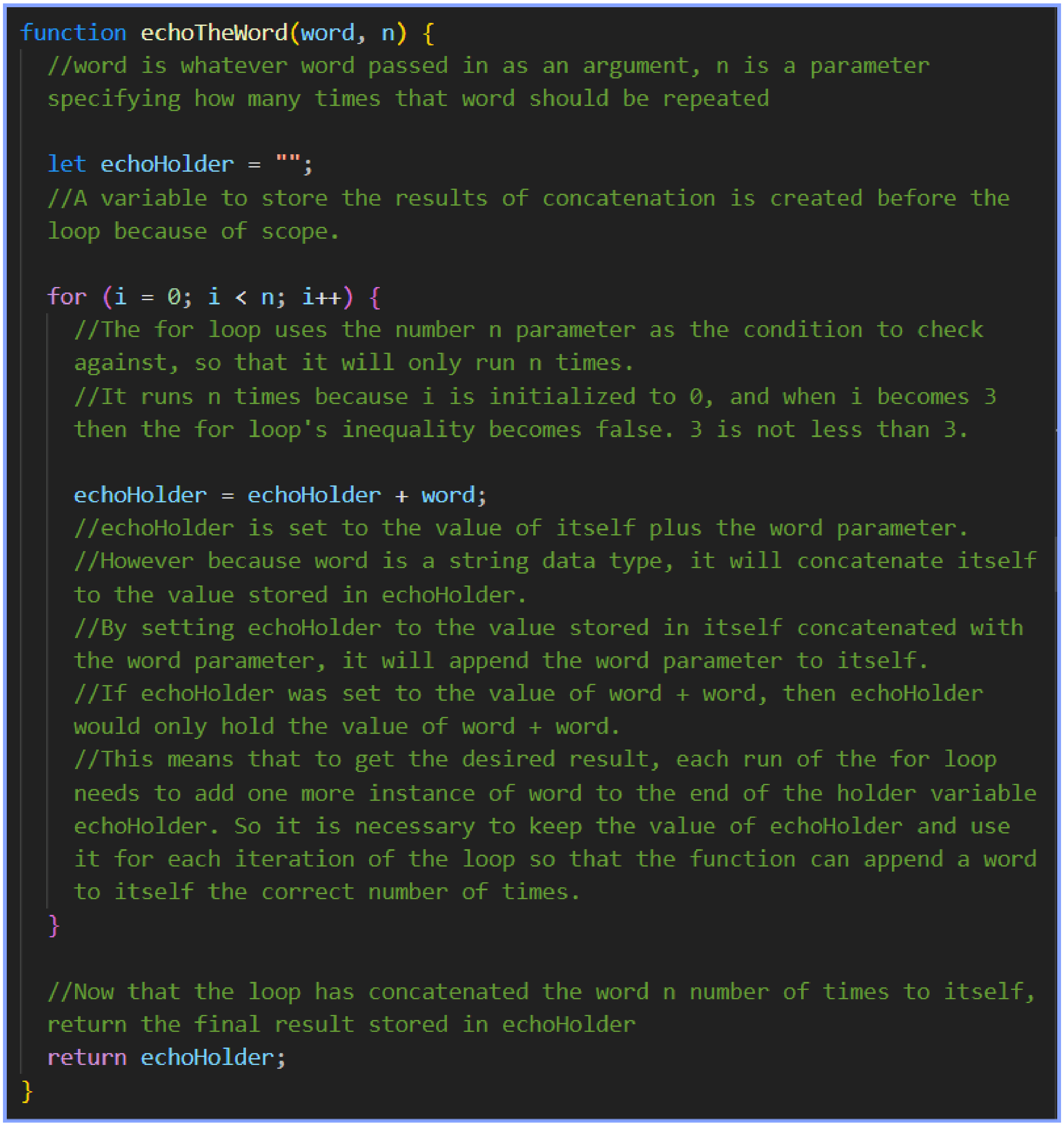
```
//4. How do you access the first element of any array?

//As stated before, arrays in JavaScript start at index 0. This means the very first element of an array will be stored at index 0.
//This can be accessed with array[0]

console.log("4. How do you access the first element of any array?");
console.log(
  "As stated in the previous question, arrays in JavaScript start at index 0. This means that the very first element of an array will be stored at index 0."
);
console.log("This can be accessed with array[0].");

//-----

/**
 * 5. Create a new array called nameLengths. Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array.
 * For example:
 * namesArray = ["Kelly", "Sam", "Kate"] //Given this array
 * nameLengths = [5, 3, 4] //Create this array
 */
console.log(
  "Exercise 5: Create a new array called nameLengths. Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array."
);
console.log("The names array from before contains the values: " + names);
//Before anything, the nameLengths array needs to be created.
let nameLengths = [];
//nameLengths is going to be filled with data that is created from the names array that was used in exercise 2.
```



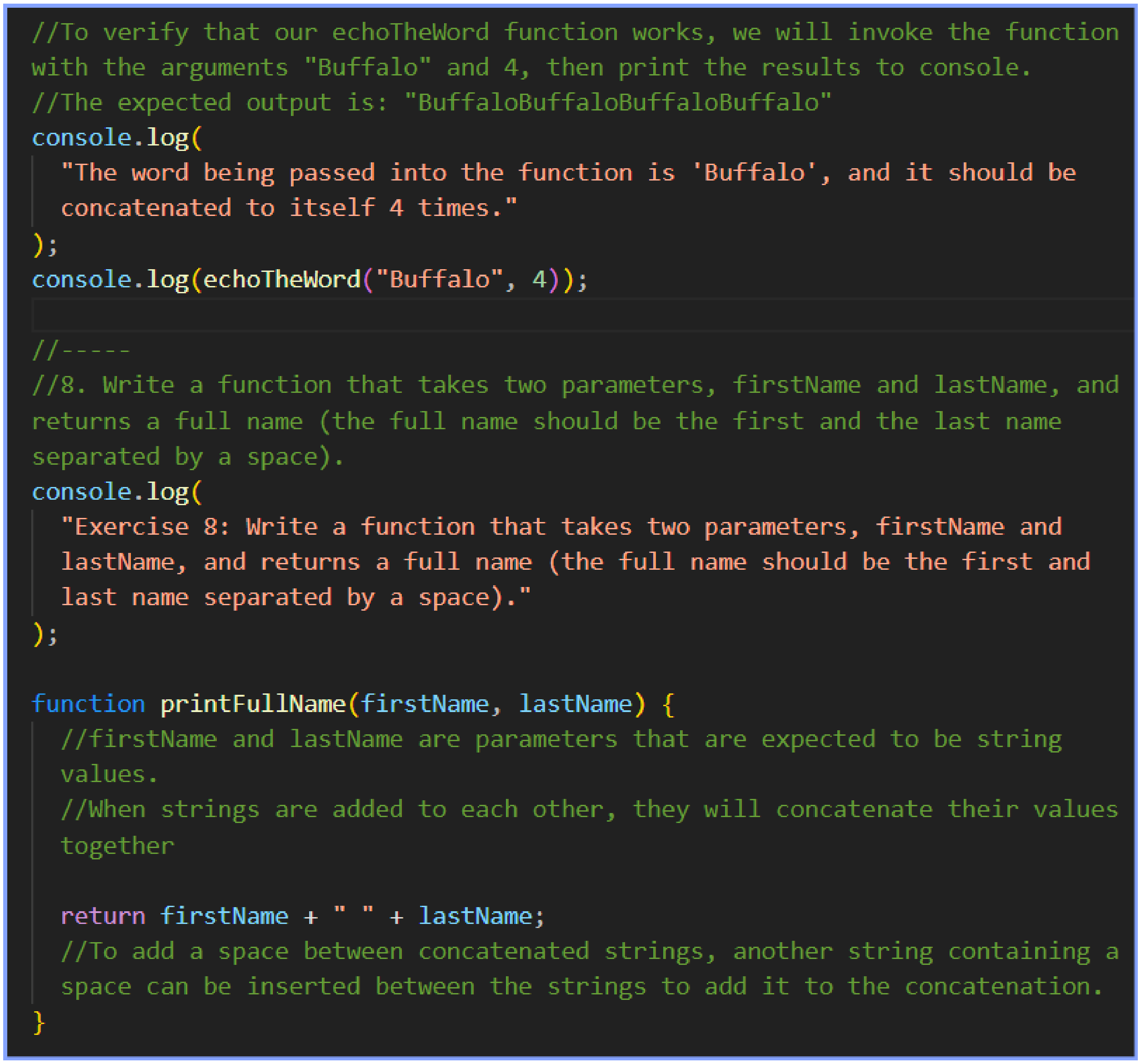
```
//I just felt like making the loop use k instead of i for fun
for (k = 0; k < names.length; k++) {
  nameLengths.push(names[k].length);
  //The push() method of arrays will add whatever is inside of the ()
  parenthesis to the end of the specified array.
  //In this case, we are adding the length of the current name to the end of
  the nameLengths array
  //We want to add each name's length to the end of the array because as k
  goes up, we are going further through the list of names.
  //And we want the first name to correlate with the first index of
  nameLengths and the last name to correlate with the last index of
  nameLengths.
}

console.log(
  "Given the names array, the length of all the names is the following: " +
    nameLengths
);

//6. Write a loop to iterate over the nameLengths array and calculate the
sum of all the elements in the array.
//Print the result to console.

console.log(
  "6. Write a loop to iterate over the nameLengths array and calculate the
  sum of all the elements in the array."
);

let nameSum = 0;
//Create a variable to hold the sum of the elements in the array
```



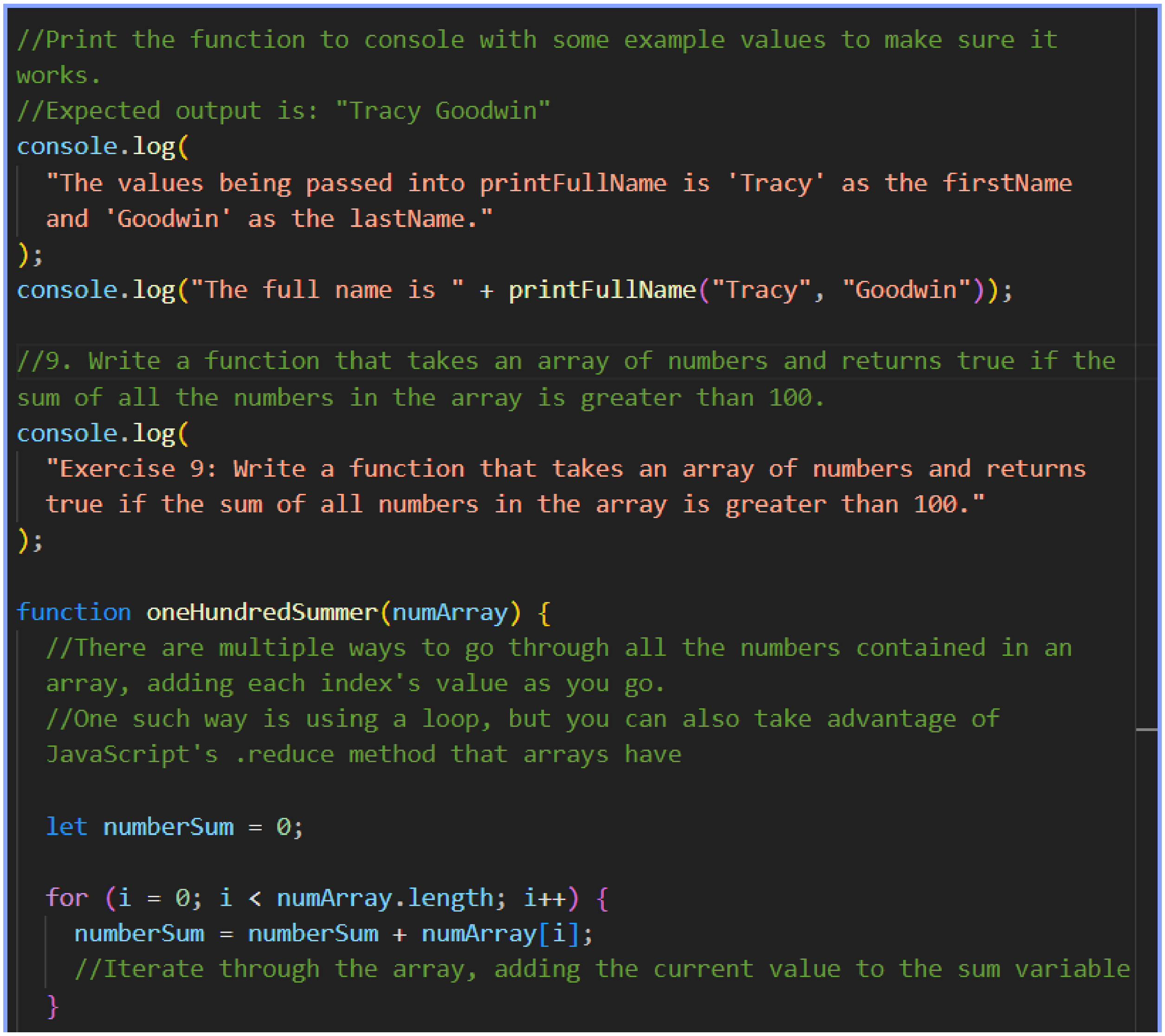
```
for (i = 0; i < nameLengths.length; i++) {
  nameSum += nameLengths[i];
  //Add the current element's value into nameSum
  //Because this is inside the loop, this will repeat until all of the
  values of each element in the array are added together.
}

console.log(
  "The sum of all of the elements in the nameLengths array is: " + nameSum
);

//The nameLengths array created and filled in exercise 5 will be used again
to complete exercise 6.

//-----

//7. Write a function that takes two parameters, word and n, as arguments
and returns the word concatenated to itself n number of times. (i.e. if I
pass in ‘Hello’ and 3, I would expect the function to return
‘HelloHelloHello’).
console.log(
  "Exercise 7: Write a function that takes two parameters, word and n, as
  arguments and returns the word concatenated to itself n number of times.
  Meaning if 'Hello' is passed in with 3, the result should be
  'HelloHelloHello'."
);
//The function is going to be called echoTheWord because this exercise is
asking to take a word and 'echo' it n number of times.
//Echoing is usually associated with repeating, and when we call echoTheWord
with any arguments it reads as if we are telling the computer to echo the
word we give it. Ex. echoTheWord("Hello", 3) reads like "echo the word Hello
3 times"
```



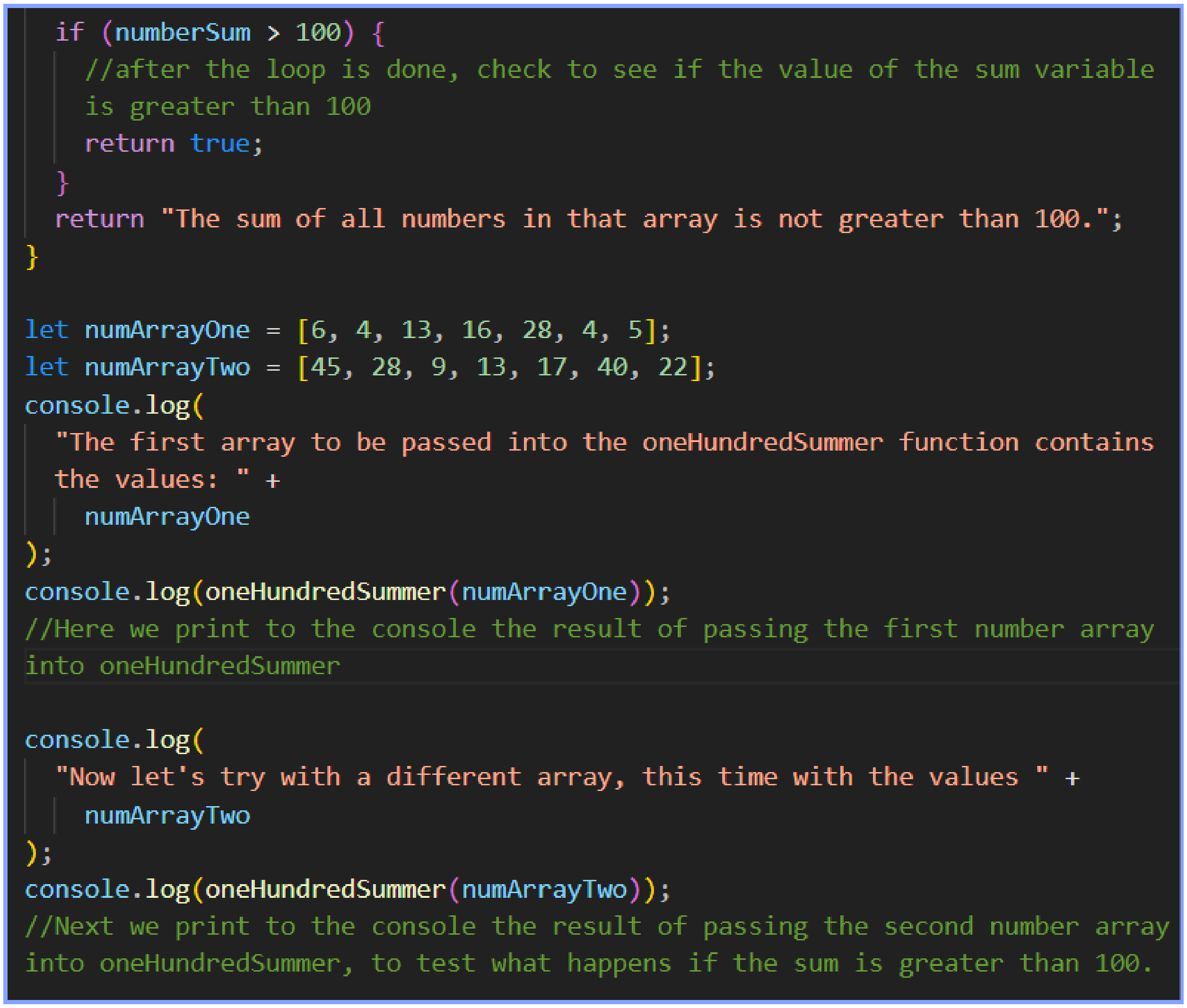
```
function echoTheWord(word, n) {
  //word is whatever word passed in as an argument, n is a parameter specifying how many times that word should be repeated

  let echoHolder = "";
  //A variable to store the results of concatenation is created before the loop because of scope.

  for (i = 0; i < n; i++) {
    //The for loop uses the number n parameter as the condition to check against, so that it will only run n times.
    //It runs n times because i is initialized to 0, and when i becomes 3 then the for loop's inequality becomes false. 3 is not less than 3.

    echoHolder = echoHolder + word;
    //echoHolder is set to the value of itself plus the word parameter.
    //However because word is a string data type, it will concatenate itself to the value stored in echoHolder.
    //By setting echoHolder to the value stored in itself concatenated with the word parameter, it will append the word parameter to itself.
    //If echoHolder was set to the value of word + word, then echoHolder would only hold the value of word + word.
    //This means that to get the desired result, each run of the for loop needs to add one more instance of word to the end of the holder variable echoHolder. So it is necessary to keep the value of echoHolder and use it for each iteration of the loop so that the function can append a word to itself the correct number of times.
  }

  //Now that the loop has concatenated the word n number of times to itself, return the final result stored in echoHolder
  return echoHolder;
}
```



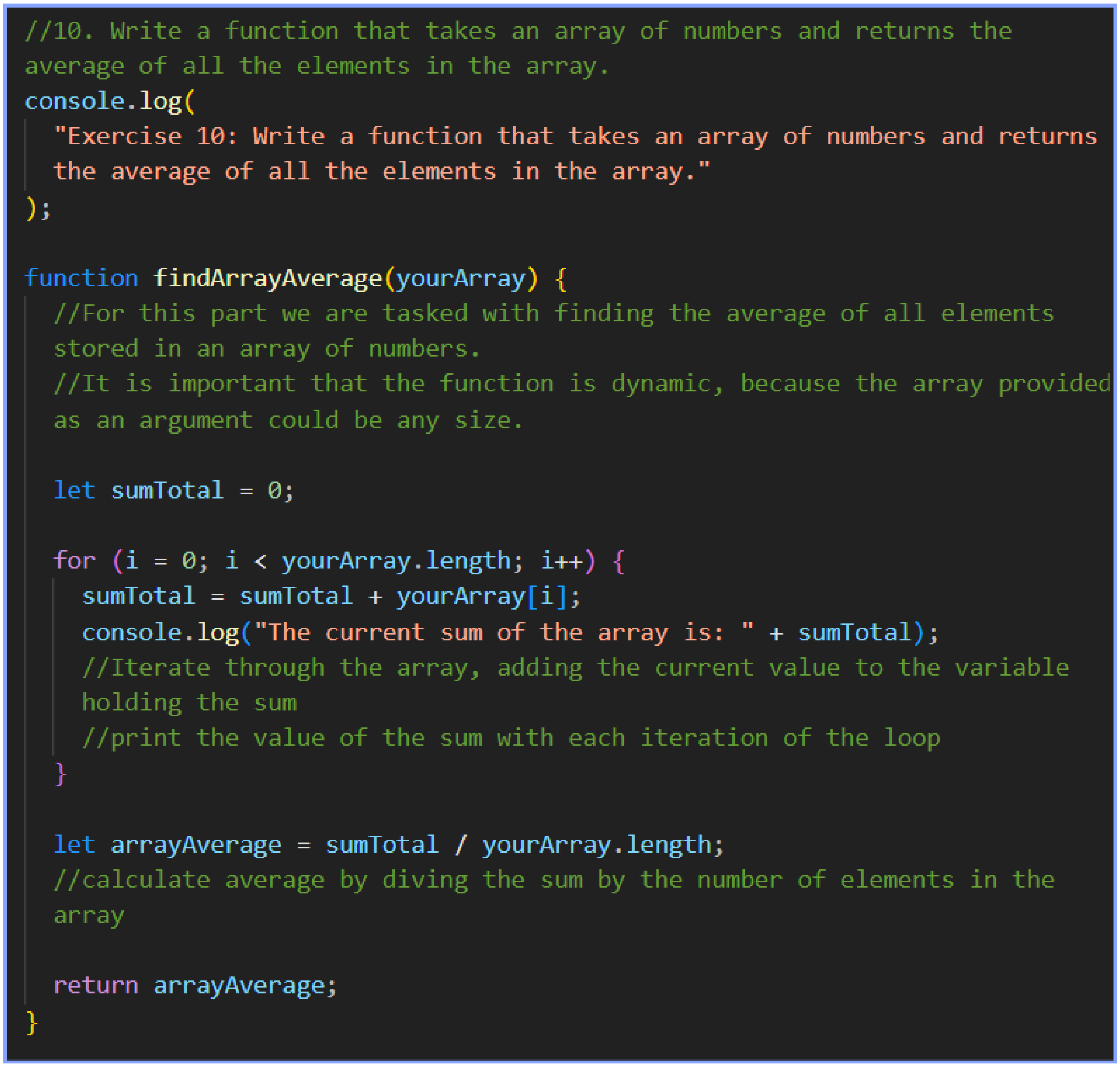
```
//To verify that our echoTheWord function works, we will invoke the function
with the arguments "Buffalo" and 4, then print the results to console.
//The expected output is: "BuffaloBuffaloBuffaloBuffalo"
console.log(
  "The word being passed into the function is 'Buffalo', and it should be
  concatenated to itself 4 times."
);
console.log(echoTheWord("Buffalo", 4));

//-----
//8. Write a function that takes two parameters, firstName and lastName, and
returns a full name (the full name should be the first and the last name
separated by a space).
console.log(
  "Exercise 8: Write a function that takes two parameters, firstName and
  lastName, and returns a full name (the full name should be the first and
  last name separated by a space)."
);

function printFullName(firstName, lastName) {
  //firstName and lastName are parameters that are expected to be string
  values.
  //When strings are added to each other, they will concatenate their values
  together

  return firstName + " " + lastName;
  //To add a space between concatenated strings, another string containing a
  space can be inserted between the strings to add it to the concatenation.
}
```



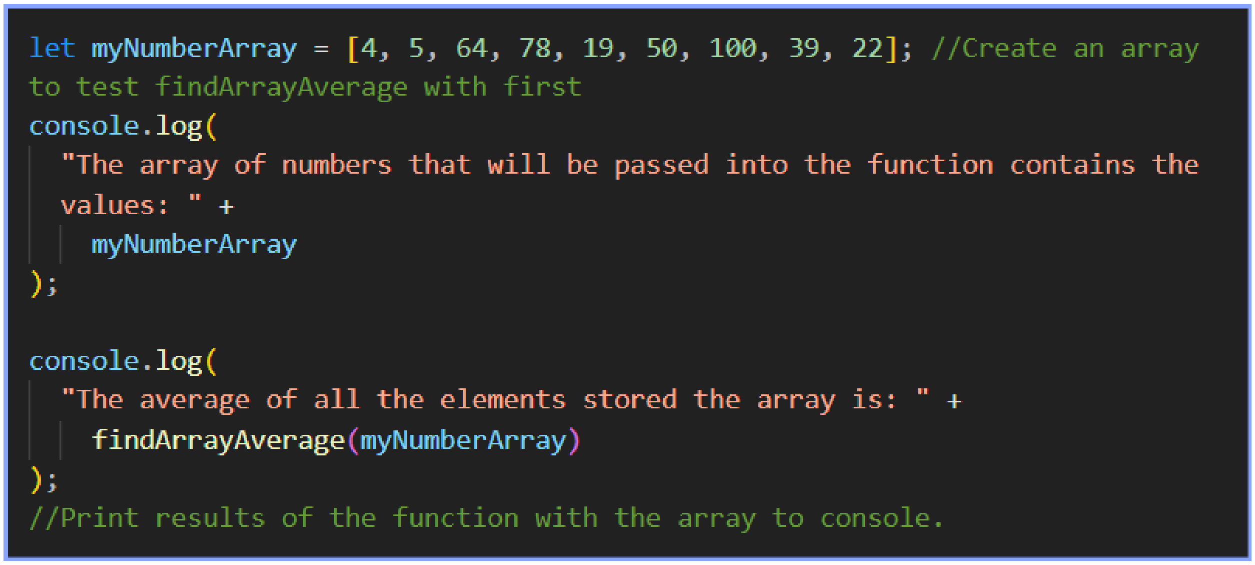
```
//Print the function to console with some example values to make sure it
works.
//Expected output is: "Tracy Goodwin"
console.log(
  "The values being passed into printFullName is 'Tracy' as the firstName
  and 'Goodwin' as the lastName."
);
console.log("The full name is " + printFullName("Tracy", "Goodwin"));

//9. Write a function that takes an array of numbers and returns true if the
sum of all the numbers in the array is greater than 100.
console.log(
  "Exercise 9: Write a function that takes an array of numbers and returns
  true if the sum of all numbers in the array is greater than 100."
);

function oneHundredSummer(numArray) {
  //There are multiple ways to go through all the numbers contained in an
  array, adding each index's value as you go.
  //One such way is using a loop, but you can also take advantage of
  JavaScript's .reduce method that arrays have

  let numberSum = 0;

  for (i = 0; i < numArray.length; i++) {
    numberSum = numberSum + numArray[i];
    //Iterate through the array, adding the current value to the sum variable
  }
```



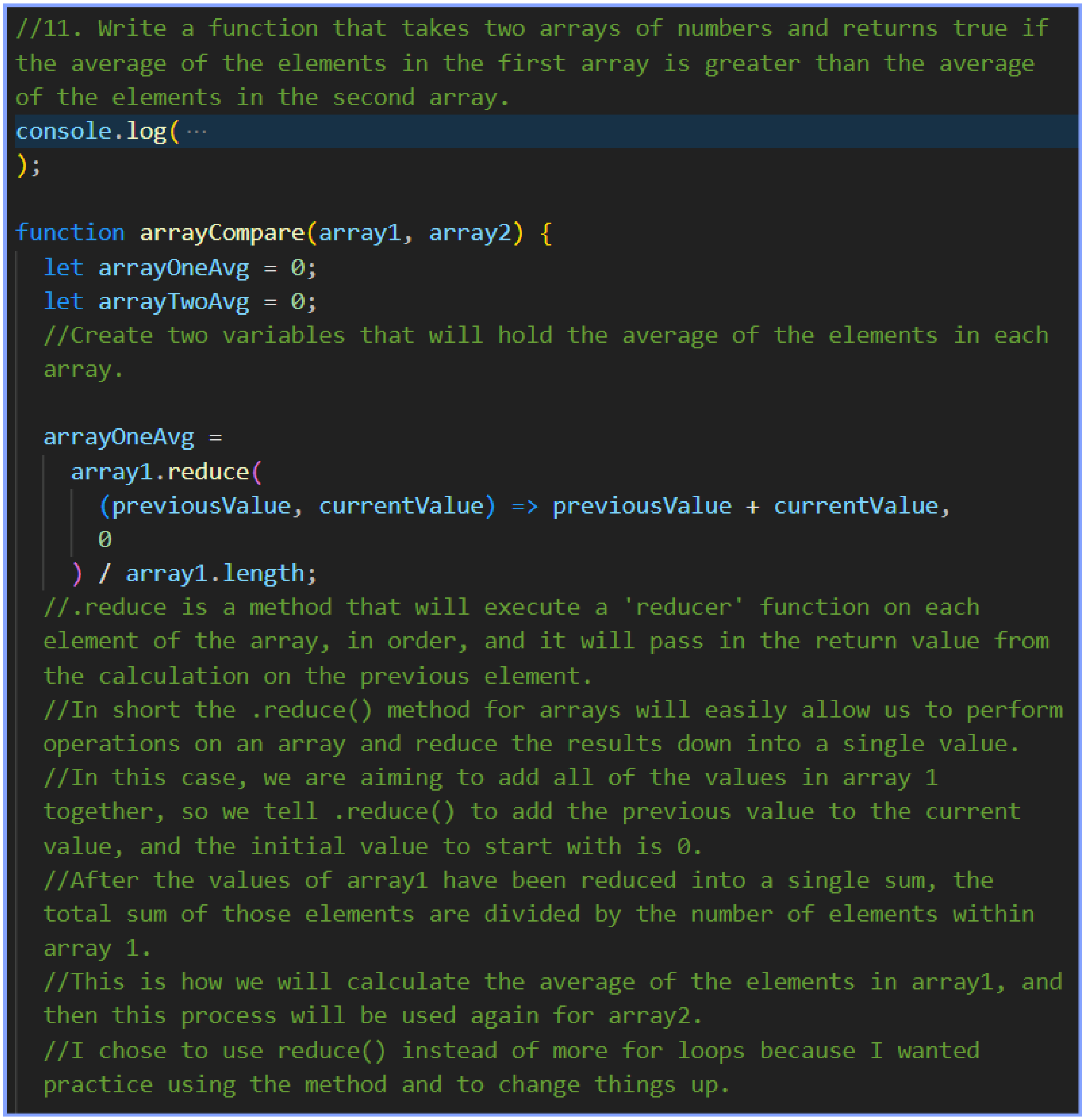
```
  if (numberSum > 100) {
    //after the loop is done, check to see if the value of the sum variable
    is greater than 100
    return true;
  }
  return "The sum of all numbers in that array is not greater than 100.";
}

let numArrayOne = [6, 4, 13, 16, 28, 4, 5];
let numArrayTwo = [45, 28, 9, 13, 17, 40, 22];
console.log(
  "The first array to be passed into the oneHundredSummer function contains
  the values: " +
    numArrayOne
);
console.log(oneHundredSummer(numArrayOne));
//Here we print to the console the result of passing the first number array
into oneHundredSummer

console.log(
  "Now let's try with a different array, this time with the values " +
    numArrayTwo
);
console.log(oneHundredSummer(numArrayTwo));
//Next we print to the console the result of passing the second number array
into oneHundredSummer, to test what happens if the sum is greater than 100.
```



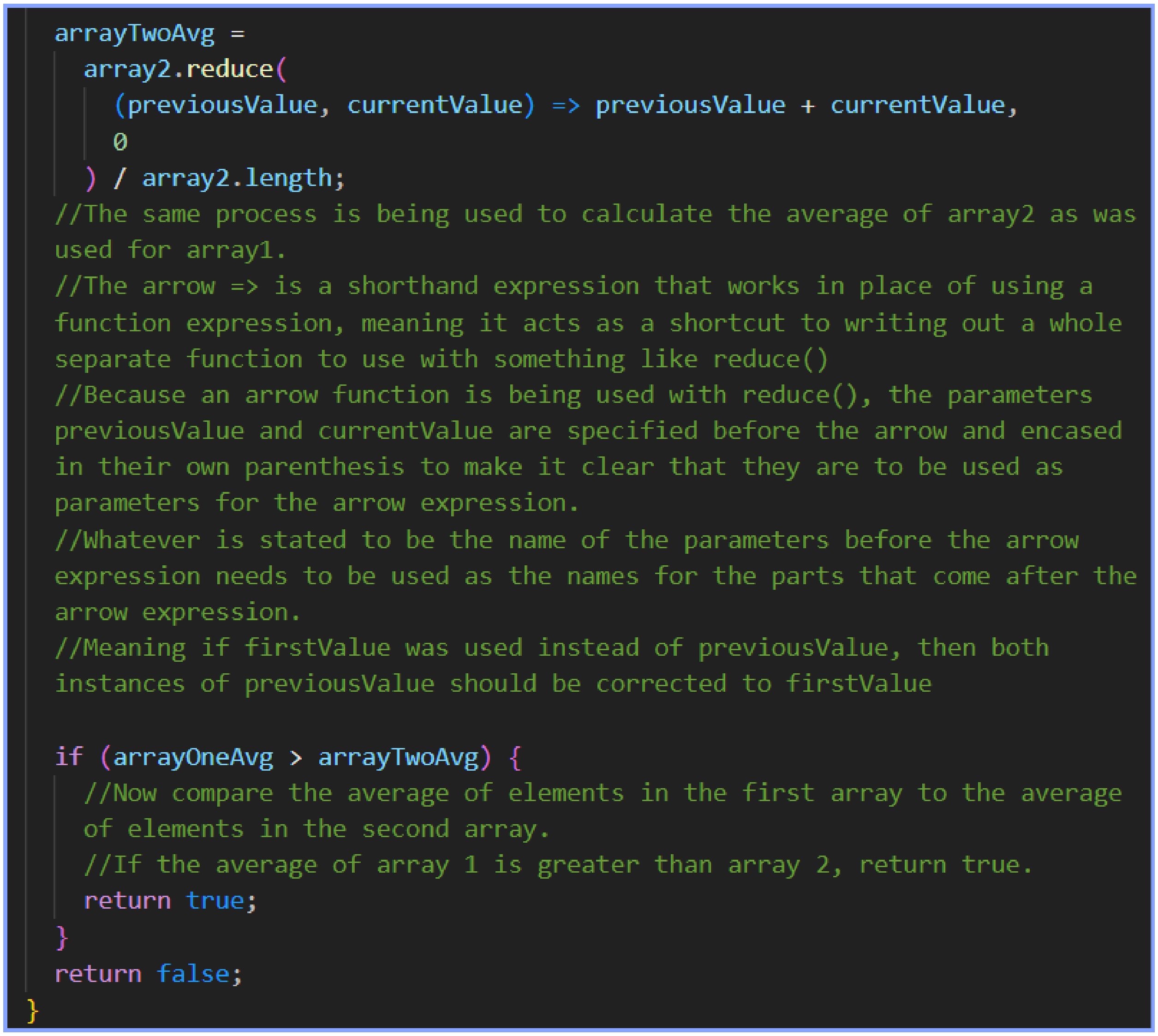
```
//10. Write a function that takes an array of numbers and returns the average of all the elements in the array.
console.log(
  "Exercise 10: Write a function that takes an array of numbers and returns the average of all the elements in the array."
);

function findArrayAverage(yourArray) {
  //For this part we are tasked with finding the average of all elements stored in an array of numbers.
  //It is important that the function is dynamic, because the array provided as an argument could be any size.

  let sumTotal = 0;

  for (i = 0; i < yourArray.length; i++) {
    sumTotal = sumTotal + yourArray[i];
    console.log("The current sum of the array is: " + sumTotal);
    //Iterate through the array, adding the current value to the variable holding the sum
    //print the value of the sum with each iteration of the loop
  }

  let arrayAverage = sumTotal / yourArray.length;
  //calculate average by diving the sum by the number of elements in the array

  return arrayAverage;
}
```



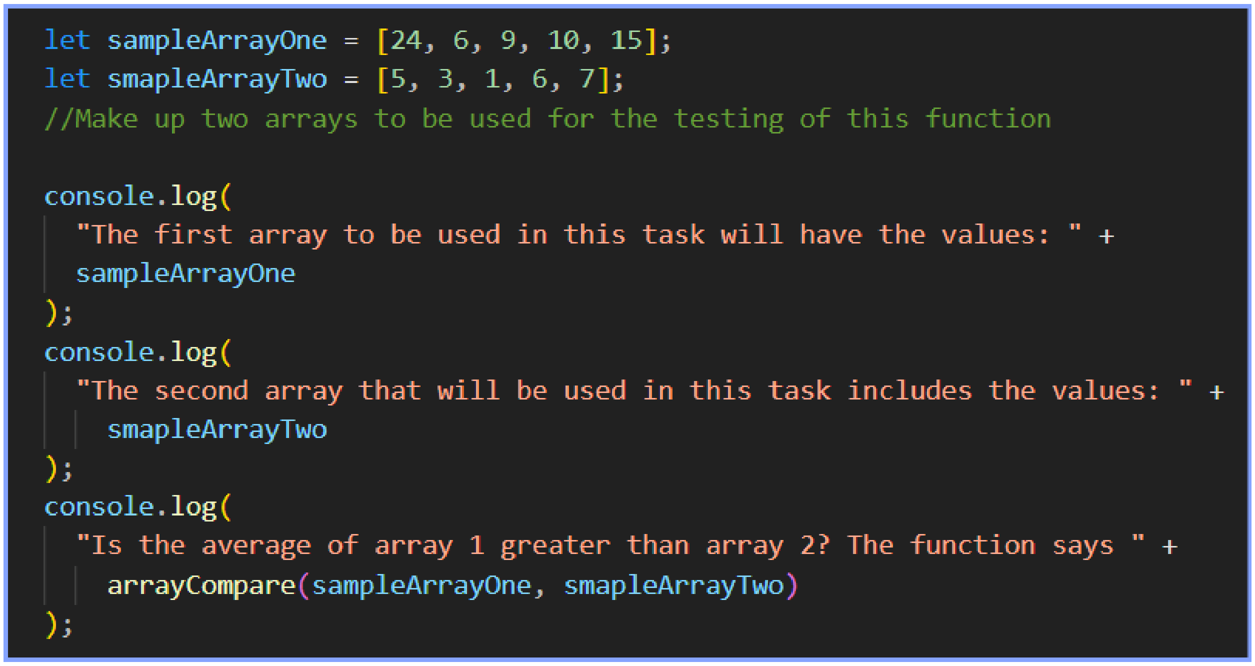
```
let myNumberArray = [4, 5, 64, 78, 19, 50, 100, 39, 22]; //Create an array
to test findArrayAverage with first
console.log(
  "The array of numbers that will be passed into the function contains the
  values: " +
    myNumberArray
);

console.log(
  "The average of all the elements stored the array is: " +
    findArrayAverage(myNumberArray)
);
//Print results of the function with the array to console.
```



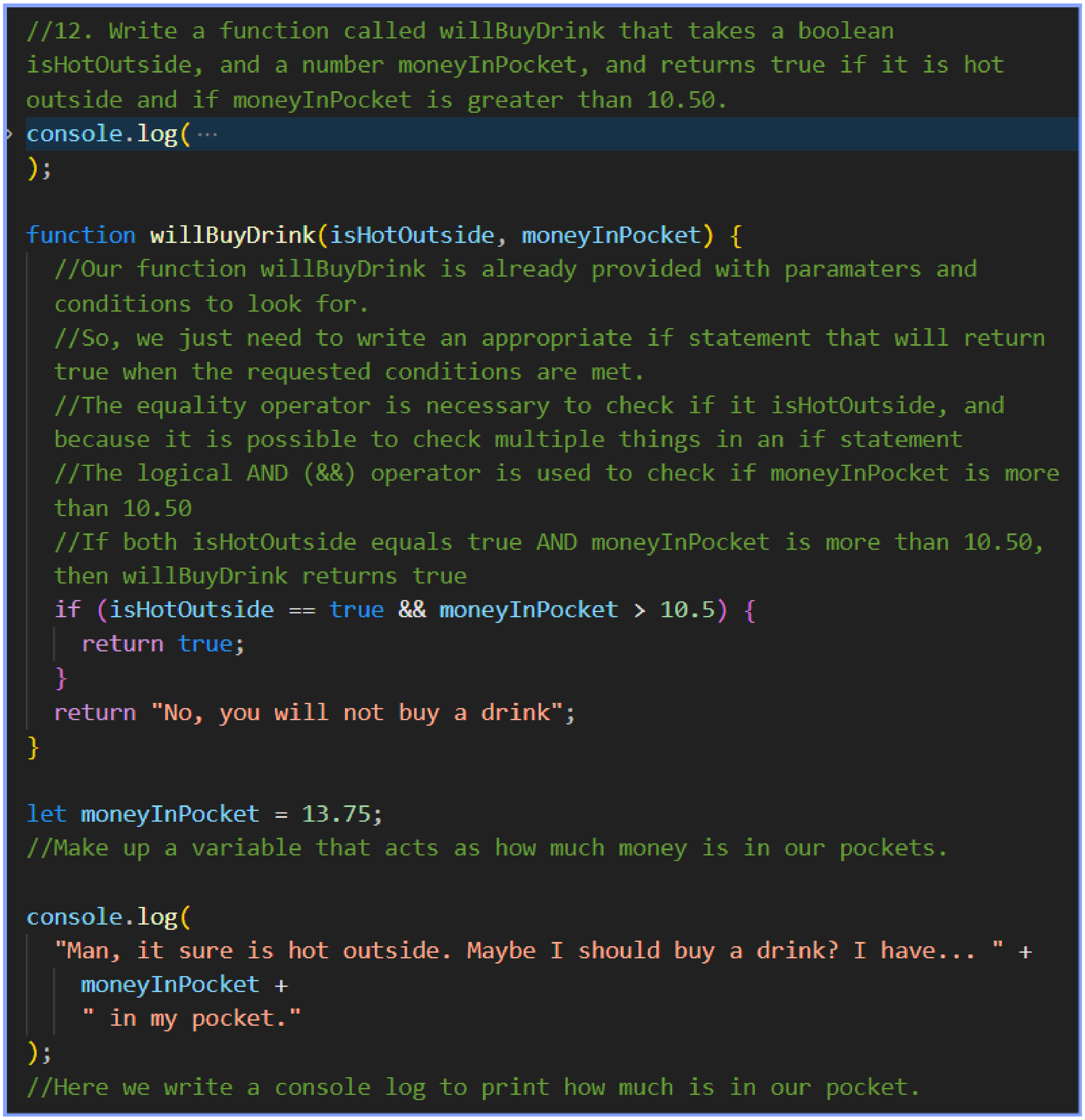
```
//11. Write a function that takes two arrays of numbers and returns true if the average of the elements in the first array is greater than the average of the elements in the second array.
console.log(⋯
);

function arrayCompare(array1, array2) {
  let arrayOneAvg = 0;
  let arrayTwoAvg = 0;
  //Create two variables that will hold the average of the elements in each array.

  arrayOneAvg =
    array1.reduce(
      (previousValue, currentValue) => previousValue + currentValue,
      0
    ) / array1.length;
  //.reduce is a method that will execute a 'reducer' function on each element of the array, in order, and it will pass in the return value from the calculation on the previous element.
  //In short the .reduce() method for arrays will easily allow us to perform operations on an array and reduce the results down into a single value.
  //In this case, we are aiming to add all of the values in array 1 together, so we tell .reduce() to add the previous value to the current value, and the initial value to start with is 0.
  //After the values of array1 have been reduced into a single sum, the total sum of those elements are divided by the number of elements within array 1.
  //This is how we will calculate the average of the elements in array1, and then this process will be used again for array2.
  //I chose to use reduce() instead of more for loops because I wanted practice using the method and to change things up.
```



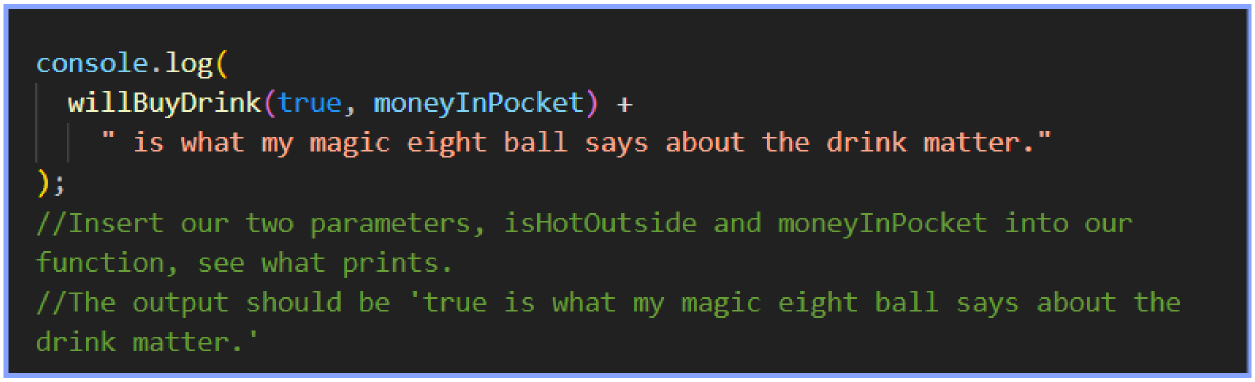
```
  arrayTwoAvg =
    array2.reduce(
      (previousValue, currentValue) => previousValue + currentValue,
      0
    ) / array2.length;
  //The same process is being used to calculate the average of array2 as was
  used for array1.
  //The arrow => is a shorthand expression that works in place of using a
  function expression, meaning it acts as a shortcut to writing out a whole
  separate function to use with something like reduce()
  //Because an arrow function is being used with reduce(), the parameters
  previousValue and currentValue are specified before the arrow and encased
  in their own parenthesis to make it clear that they are to be used as
  parameters for the arrow expression.
  //Whatever is stated to be the name of the parameters before the arrow
  expression needs to be used as the names for the parts that come after the
  arrow expression.
  //Meaning if firstValue was used instead of previousValue, then both
  instances of previousValue should be corrected to firstValue

  if (arrayOneAvg > arrayTwoAvg) {
    //Now compare the average of elements in the first array to the average
    of elements in the second array.
    //If the average of array 1 is greater than array 2, return true.
    return true;
  }
  return false;
}
```



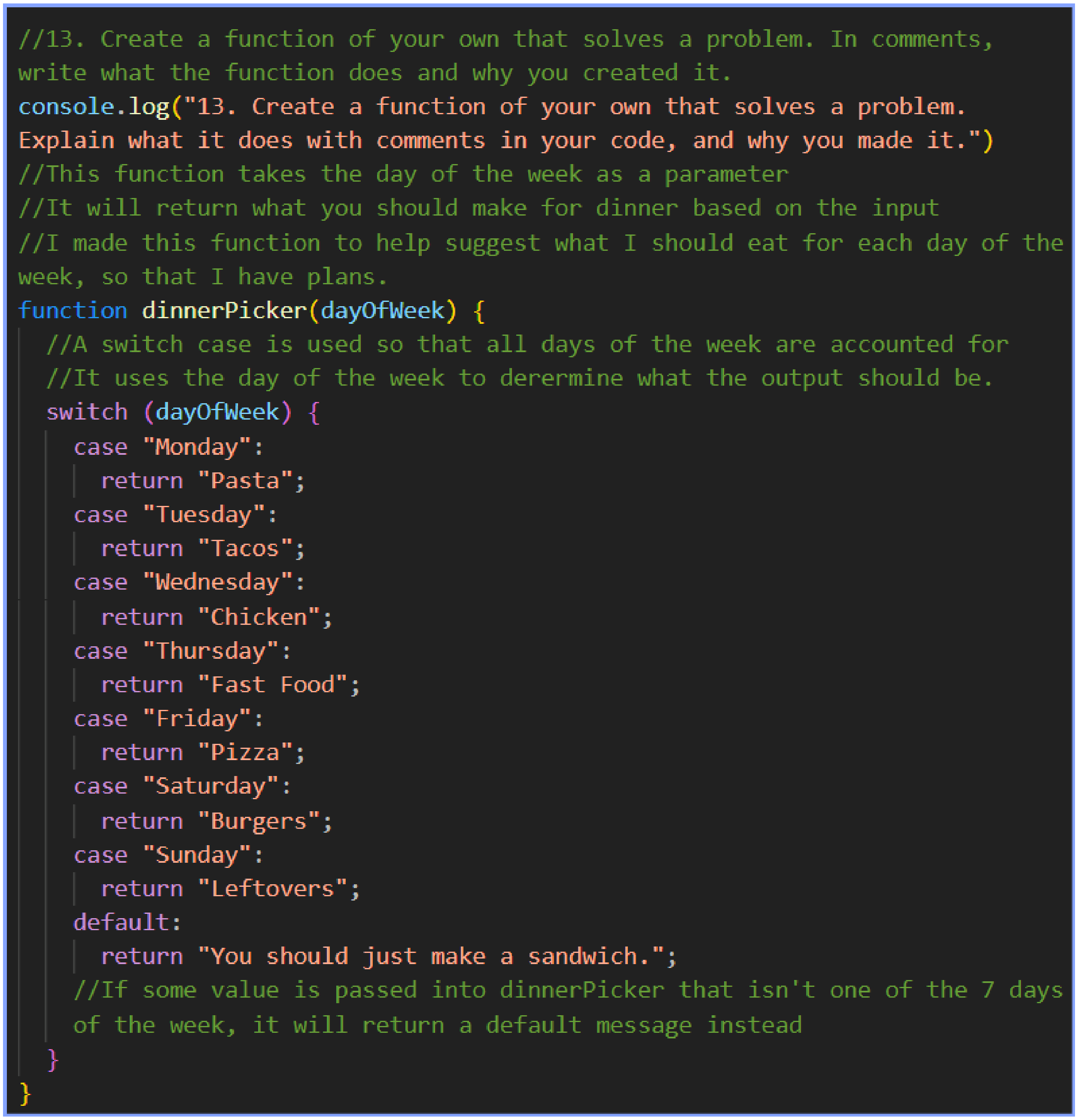
```
let sampleArrayOne = [24, 6, 9, 10, 15];
let smapleArrayTwo = [5, 3, 1, 6, 7];
//Make up two arrays to be used for the testing of this function

console.log(
  "The first array to be used in this task will have the values: " +
  sampleArrayOne
);
console.log(
  "The second array that will be used in this task includes the values: " +
    smapleArrayTwo
);
console.log(
  "Is the average of array 1 greater than array 2? The function says " +
    arrayCompare(sampleArrayOne, smapleArrayTwo)
);
```



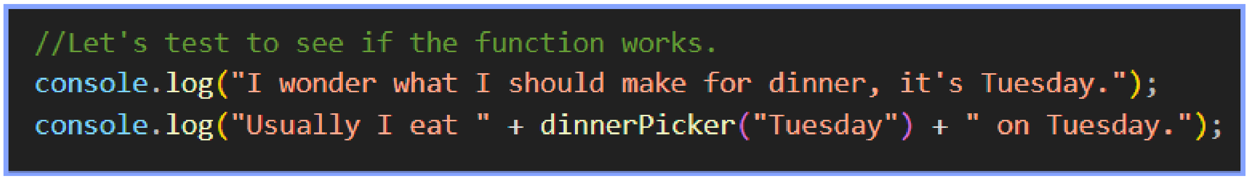
```
//12. Write a function called willBuyDrink that takes a boolean
isHotOutside, and a number moneyInPocket, and returns true if it is hot
outside and if moneyInPocket is greater than 10.50.
console.log( ···
);

function willBuyDrink(isHotOutside, moneyInPocket) {
  //Our function willBuyDrink is already provided with paramaters and
  conditions to look for.
  //So, we just need to write an appropriate if statement that will return
  true when the requested conditions are met.
  //The equality operator is necessary to check if it isHotOutside, and
  because it is possible to check multiple things in an if statement
  //The logical AND (&&) operator is used to check if moneyInPocket is more
  than 10.50
  //If both isHotOutside equals true AND moneyInPocket is more than 10.50,
  then willBuyDrink returns true
  if (isHotOutside == true && moneyInPocket > 10.5) {
    return true;
  }
  return "No, you will not buy a drink";
}

let moneyInPocket = 13.75;
//Make up a variable that acts as how much money is in our pockets.

console.log(
  "Man, it sure is hot outside. Maybe I should buy a drink? I have... " +
    moneyInPocket +
    " in my pocket."
);
//Here we write a console log to print how much is in our pocket.
```



```
console.log(
  willBuyDrink(true, moneyInPocket) +
    " is what my magic eight ball says about the drink matter."
);
//Insert our two parameters, isHotOutside and moneyInPocket into our
function, see what prints.
//The output should be 'true is what my magic eight ball says about the
drink matter.'
```



```
//13. Create a function of your own that solves a problem. In comments, write what the function does and why you created it.
console.log("13. Create a function of your own that solves a problem. Explain what it does with comments in your code, and why you made it.")
//This function takes the day of the week as a parameter
//It will return what you should make for dinner based on the input
//I made this function to help suggest what I should eat for each day of the week, so that I have plans.
function dinnerPicker(dayOfWeek) {
  //A switch case is used so that all days of the week are accounted for
  //It uses the day of the week to derermine what the output should be.
  switch (dayOfWeek) {
    case "Monday":
      return "Pasta";
    case "Tuesday":
      return "Tacos";
    case "Wednesday":
      return "Chicken";
    case "Thursday":
      return "Fast Food";
    case "Friday":
      return "Pizza";
    case "Saturday":
      return "Burgers";
    case "Sunday":
      return "Leftovers";
    default:
      return "You should just make a sandwich.";
    //If some value is passed into dinnerPicker that isn't one of the 7 days of the week, it will return a default message instead
  }
}
```

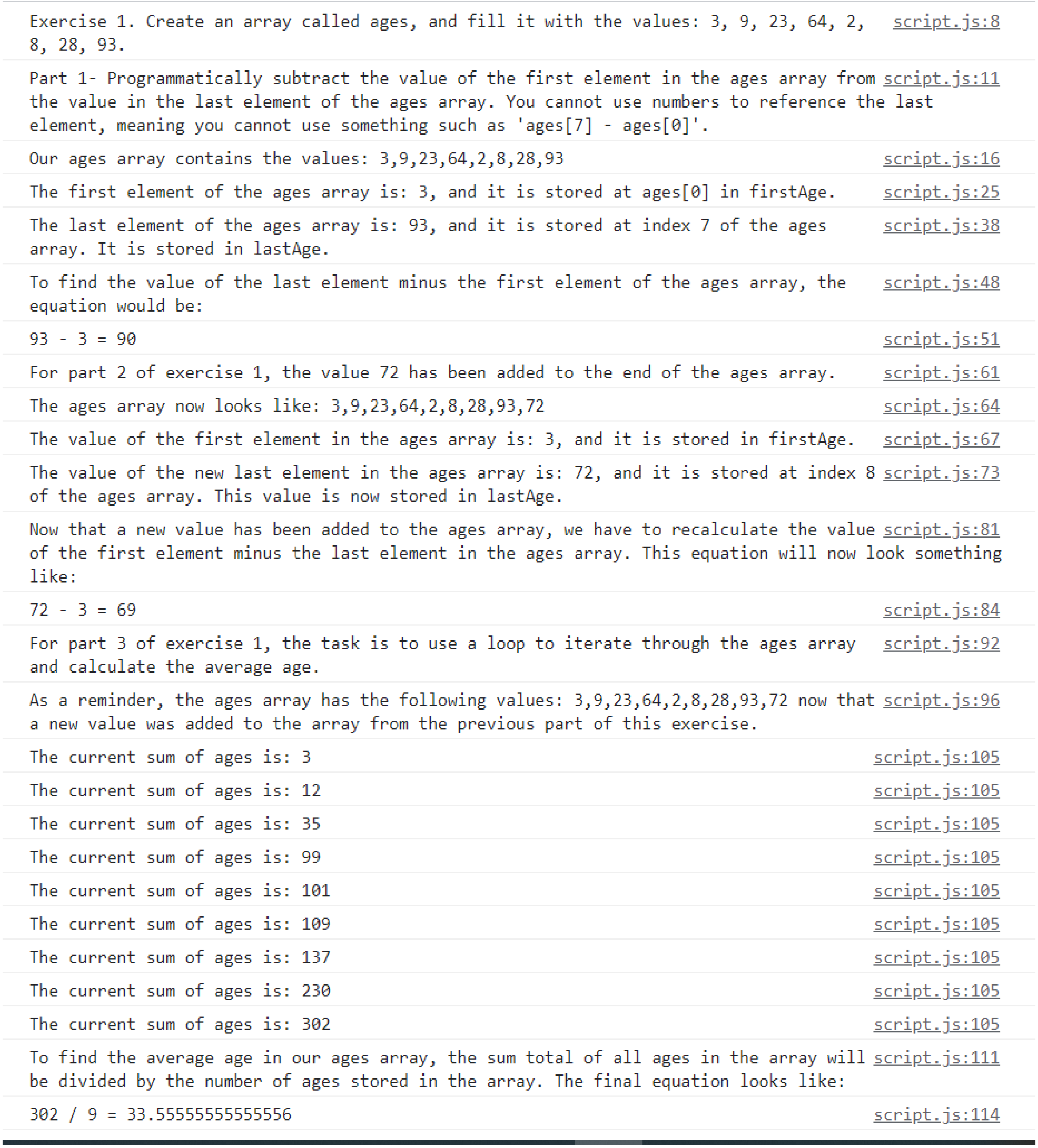


```
//Let's test to see if the function works.
console.log("I wonder what I should make for dinner, it's Tuesday.");
console.log("Usually I eat " + dinnerPicker("Tuesday") + " on Tuesday.");
```

Screenshots of Running Application:

PROMINEO TECH

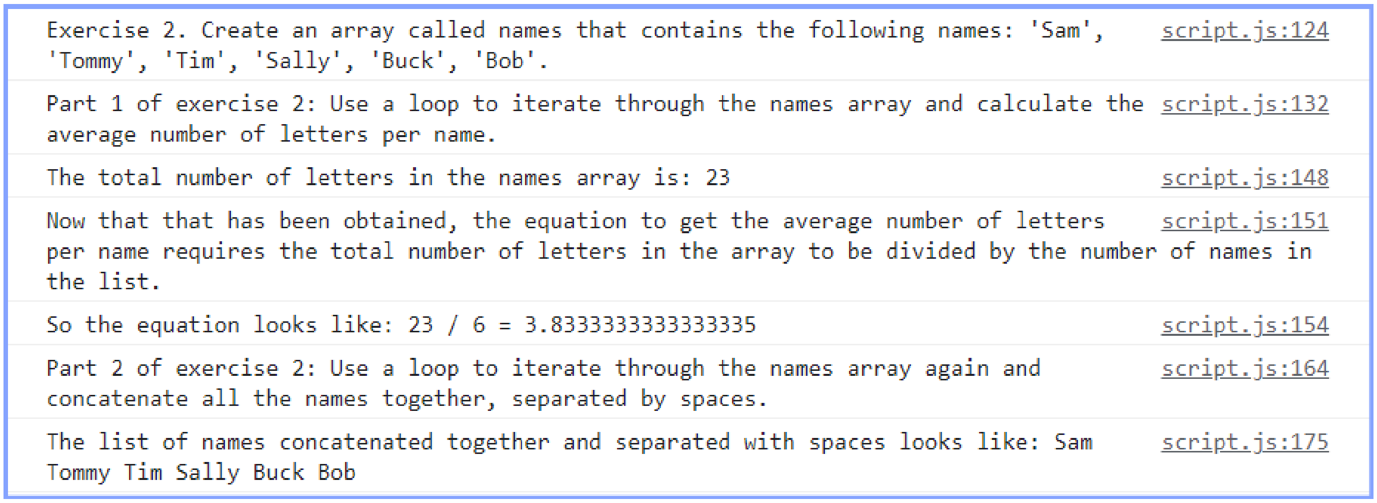
Exercise 1, parts 1-3



```
Exercise 1. Create an array called ages, and fill it with the values: 3, 9, 23, 64, 2,   script.js:8
8, 28, 93.
Part 1- Programmatically subtract the value of the first element in the ages array from script.js:11
the value in the last element of the ages array. You cannot use numbers to reference the last
element, meaning you cannot use something such as 'ages[7] - ages[0]'.
Our ages array contains the values: 3,9,23,64,2,8,28,93                                 script.js:16
The first element of the ages array is: 3, and it is stored at ages[0] in firstAge.     script.js:25
The last element of the ages array is: 93, and it is stored at index 7 of the ages      script.js:38
array. It is stored in lastAge.
To find the value of the last element minus the first element of the ages array, the    script.js:48
equation would be:
93 - 3 = 90                                                                             script.js:51
For part 2 of exercise 1, the value 72 has been added to the end of the ages array.     script.js:61
The ages array now looks like: 3,9,23,64,2,8,28,93,72                                   script.js:64
The value of the first element in the ages array is: 3, and it is stored in firstAge.   script.js:67
The value of the new last element in the ages array is: 72, and it is stored at index 8 script.js:73
of the ages array. This value is now stored in lastAge.
Now that a new value has been added to the ages array, we have to recalculate the value script.js:81
of the first element minus the last element in the ages array. This equation will now look something
like:
72 - 3 = 69                                                                             script.js:84
For part 3 of exercise 1, the task is to use a loop to iterate through the ages array   script.js:92
and calculate the average age.
As a reminder, the ages array has the following values: 3,9,23,64,2,8,28,93,72 now that script.js:96
a new value was added to the array from the previous part of this exercise.
The current sum of ages is: 3                                                          script.js:105
The current sum of ages is: 12                                                         script.js:105
The current sum of ages is: 35                                                         script.js:105
The current sum of ages is: 99                                                         script.js:105
The current sum of ages is: 101                                                        script.js:105
The current sum of ages is: 109                                                        script.js:105
The current sum of ages is: 137                                                        script.js:105
The current sum of ages is: 230                                                        script.js:105
The current sum of ages is: 302                                                        script.js:105
To find the average age in our ages array, the sum total of all ages in the array will script.js:111
be divided by the number of ages stored in the array. The final equation looks like:
302 / 9 = 33.55555555555556                                                            script.js:114
```

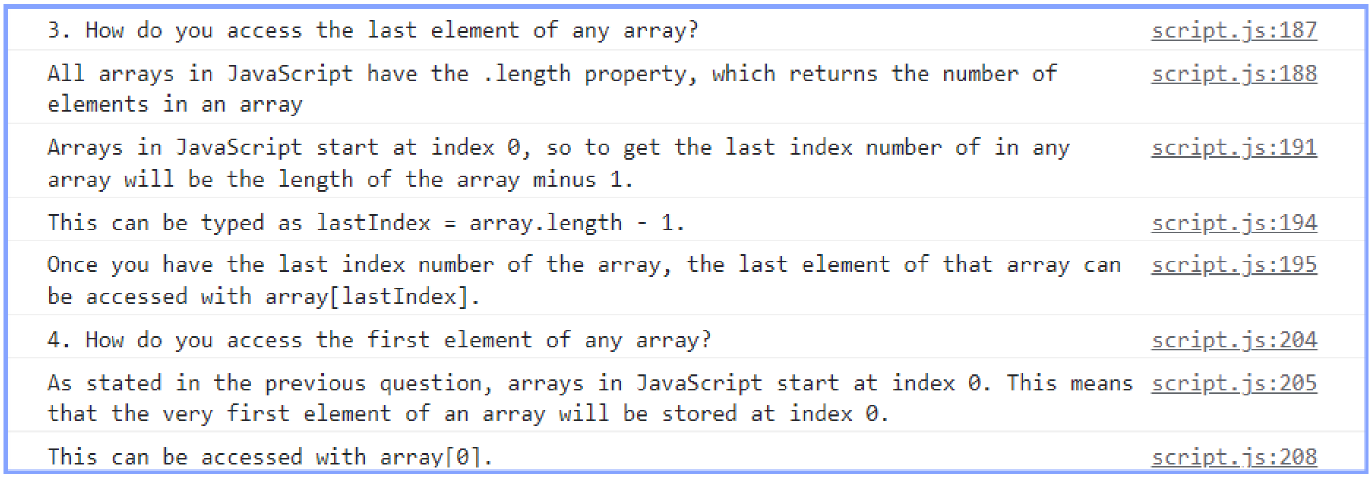
PROMINEO TECH

Exercise 2, parts 1 & 2



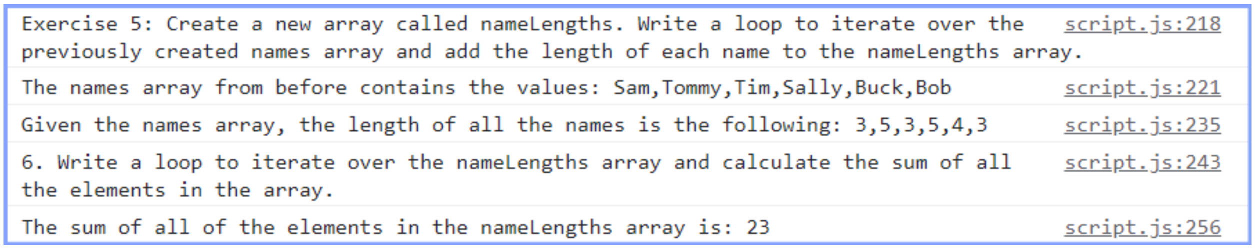
```
Exercise 2. Create an array called names that contains the following names: 'Sam',    script.js:124
'Tommy', 'Tim', 'Sally', 'Buck', 'Bob'.
Part 1 of exercise 2: Use a loop to iterate through the names array and calculate the  script.js:132
average number of letters per name.
The total number of letters in the names array is: 23                                  script.js:148
Now that that has been obtained, the equation to get the average number of letters     script.js:151
per name requires the total number of letters in the array to be divided by the number of names in
the list.
So the equation looks like: 23 / 6 = 3.8333333333333335                                script.js:154
Part 2 of exercise 2: Use a loop to iterate through the names array again and          script.js:164
concatenate all the names together, separated by spaces.
The list of names concatenated together and separated with spaces looks like: Sam      script.js:175
Tommy Tim Sally Buck Bob
```

Exercise 3 & 4



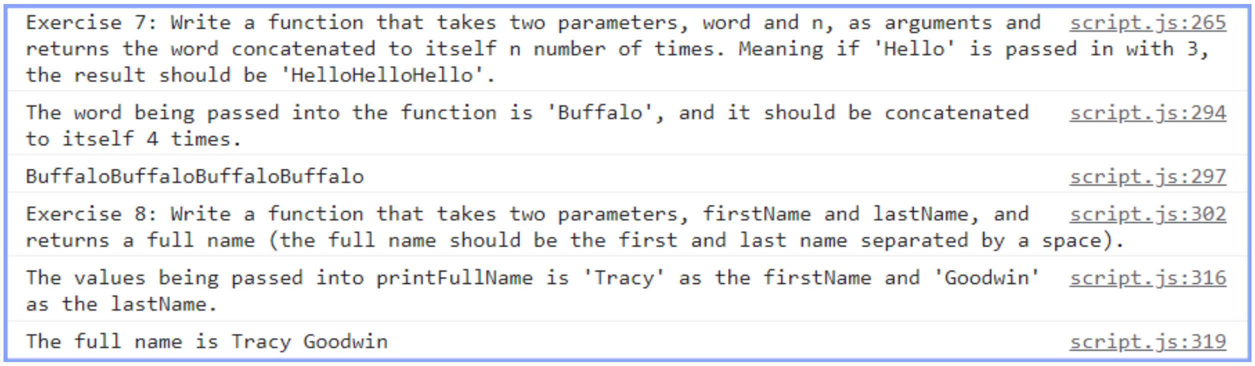
```
3. How do you access the last element of any array?                                    script.js:187
All arrays in JavaScript have the .length property, which returns the number of       script.js:188
elements in an array
Arrays in JavaScript start at index 0, so to get the last index number of in any       script.js:191
array will be the length of the array minus 1.
This can be typed as lastIndex = array.length - 1.                                     script.js:194
Once you have the last index number of the array, the last element of that array can  script.js:195
be accessed with array[lastIndex].
4. How do you access the first element of any array?                                   script.js:204
As stated in the previous question, arrays in JavaScript start at index 0. This means script.js:205
that the very first element of an array will be stored at index 0.
This can be accessed with array[0].                                                    script.js:208
```

Exercises 5 & 6



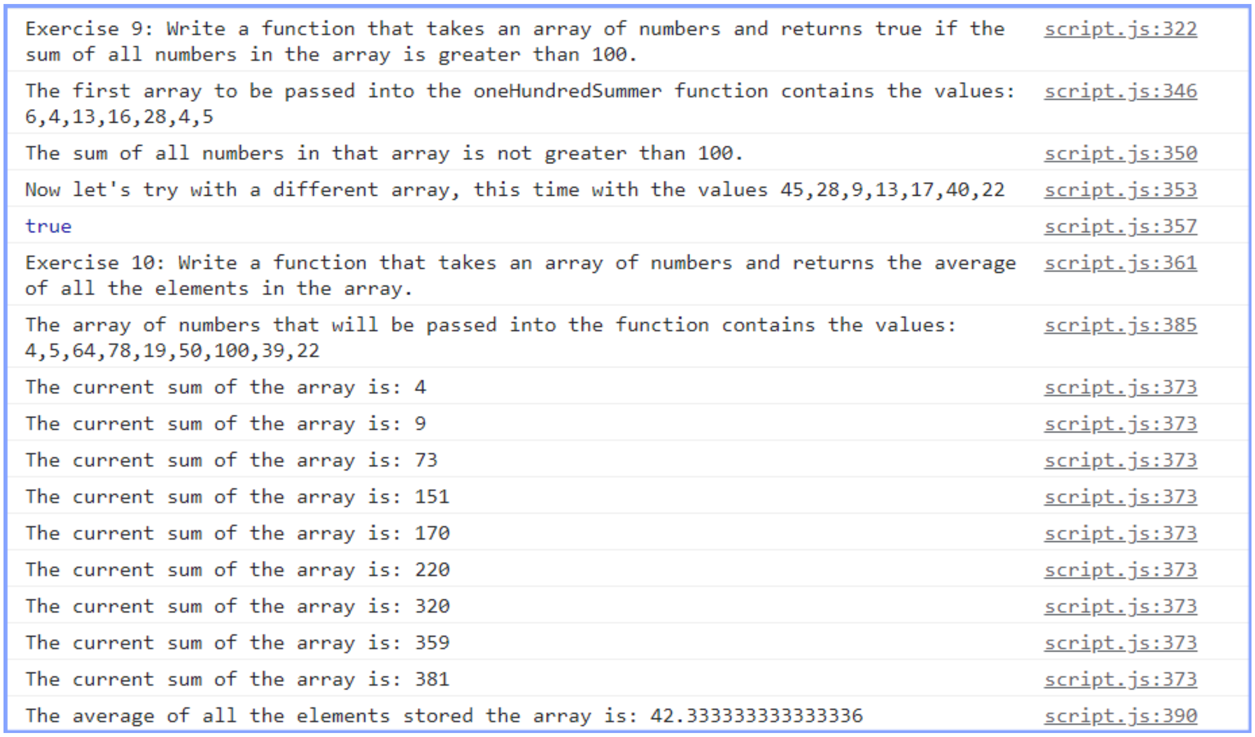
```
Exercise 5: Create a new array called nameLengths. Write a loop to iterate over the    script.js:218
previously created names array and add the length of each name to the nameLengths array.
The names array from before contains the values: Sam,Tommy,Tim,Sally,Buck,Bob          script.js:221
Given the names array, the length of all the names is the following: 3,5,3,5,4,3       script.js:235
6. Write a loop to iterate over the nameLengths array and calculate the sum of all     script.js:243
the elements in the array.
The sum of all of the elements in the nameLengths array is: 23                         script.js:256
```

Exercises 7 & 8



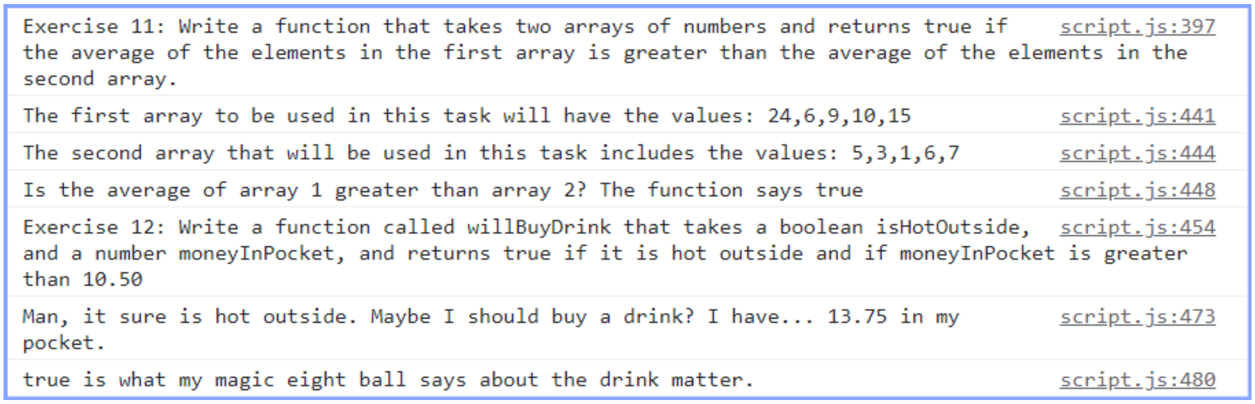
```
Exercise 7: Write a function that takes two parameters, word and n, as arguments and   script.js:265
returns the word concatenated to itself n number of times. Meaning if 'Hello' is passed in with 3,
the result should be 'HelloHelloHello'.
The word being passed into the function is 'Buffalo', and it should be concatenated    script.js:294
to itself 4 times.
BuffaloBuffaloBuffaloBuffalo                                                           script.js:297
Exercise 8: Write a function that takes two parameters, firstName and lastName, and    script.js:302
returns a full name (the full name should be the first and last name separated by a space).
The values being passed into printFullName is 'Tracy' as the firstName and 'Goodwin'   script.js:316
as the lastName.
The full name is Tracy Goodwin                                                         script.js:319
```

Exercises 9 & 10



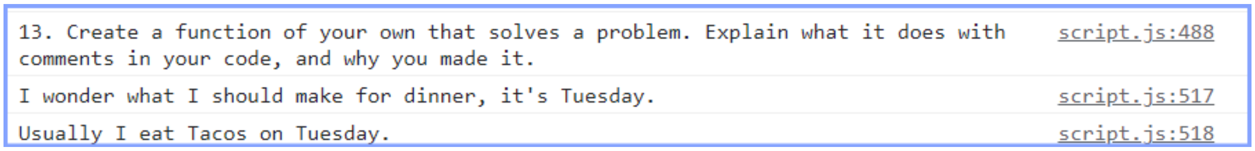
```
Exercise 9: Write a function that takes an array of numbers and returns true if the     script.js:322
sum of all numbers in the array is greater than 100.
The first array to be passed into the oneHundredSummer function contains the values:   script.js:346
6,4,13,16,28,4,5
The sum of all numbers in that array is not greater than 100.                           script.js:350
Now let's try with a different array, this time with the values 45,28,9,13,17,40,22    script.js:353
true                                                                                    script.js:357
Exercise 10: Write a function that takes an array of numbers and returns the average   script.js:361
of all the elements in the array.
The array of numbers that will be passed into the function contains the values:        script.js:385
4,5,64,78,19,50,100,39,22
The current sum of the array is: 4                                                      script.js:373
The current sum of the array is: 9                                                      script.js:373
The current sum of the array is: 73                                                     script.js:373
The current sum of the array is: 151                                                    script.js:373
The current sum of the array is: 170                                                    script.js:373
The current sum of the array is: 220                                                    script.js:373
The current sum of the array is: 320                                                    script.js:373
The current sum of the array is: 359                                                    script.js:373
The current sum of the array is: 381                                                    script.js:373
The average of all the elements stored the array is: 42.333333333333336                 script.js:390
```

Exercises 11 & 12



```
Exercise 11: Write a function that takes two arrays of numbers and returns true if     script.js:397
the average of the elements in the first array is greater than the average of the elements in the
second array.
The first array to be used in this task will have the values: 24,6,9,10,15             script.js:441
The second array that will be used in this task includes the values: 5,3,1,6,7         script.js:444
Is the average of array 1 greater than array 2? The function says true                 script.js:448
Exercise 12: Write a function called willBuyDrink that takes a boolean isHotOutside,   script.js:454
and a number moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater
than 10.50
Man, it sure is hot outside. Maybe I should buy a drink? I have... 13.75 in my         script.js:473
pocket.
true is what my magic eight ball says about the drink matter.                          script.js:480
```

Exercise 13



```
13. Create a function of your own that solves a problem. Explain what it does with     script.js:488
comments in your code, and why you made it.
I wonder what I should make for dinner, it's Tuesday.                                  script.js:517
Usually I eat Tacos on Tuesday.                                                        script.js:518
```