Intro to JavaScript Week 3 Coding Assignment

Points possible: 75

URL to Your GitHub Repository:

https://github.com/Stringerdt/Week3_Coding_Assignment

URL to Your Coding Assignment Video:

https://youtu.be/FYzzO6nWfBk

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.

PROMINEO TECH

Screenshots of Code:

```
// 1. Create an array called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.

let ages = [ 3, 9, 23, 64, 2, 8, 28, 93 ];
console.log(ages);

// 1a. 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.

// I subtract array[array.length - 1], which is the last element of an array, from array[0], the first element.
// ages = [ 3, 9, 23, 64, 2, 8, 28, 93 ] => function => array[array.length - 1 ] - array[0] = 93 - 3 = 90;

const subtractArray = (array) => array[array.length - 1] - array[0];

console.log(subtractArray(ages)); // => 90

// 1b. Add a new age to your array and repeat the step above to ensure it is dynamic
// (works for arrays of different lengths).

// array.push(40) will add 40 to the array, at the array[array.length - 1] index. After push, ages === [ 3, 9, 23, 64, 2, 8, 28, 93, 40 ]
ages.push(40);

console.log( `After push, ${subtractArray(ages)}`); // => 37

// 1c. Use a loop to iterate through the array and calculate the average age.
// Print the result to the console.

// First, i initialize sumAges = 0, then I use a for of loop to iterate through each element, adding each element's value to sumAges.
// I then divide the total sum by the length of the array, to get the average.
// ages = [ 3, 9, 23, 64, 2, 8, 28, 93, 40 ] => loop => ( 3 + 9 + 23 + 64 + 2 + 8 + 28 + 93 + 40) / ages.lenth = 270 / 9 = 30.
let sumAges = 0;
for (age of ages) {
    sumAges += age;
} console.log(`The average age is ${sumAges / ages.length}`);
```

```
// 2. Create an array called names that contains the following values: 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob'.

const names = [ 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob' ];


// 2a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.

// Similar to the previous loop, I initialize sumNames = 0, then loop through each name of the array. The difference is I have to add
// the length of each string element to the sum, rather than a number. I then set the result to 2 decimal places.
// names = [ 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob' ] => loop => ((3 + 5 + 3 + 5 + 4 + 3) / names.length) = (23 / 6) = 3.8333
let sumNames = 0;
for (firstName of names) {
    sumNames += firstName.length
}

console.log(`Each name contains ${(sumNames / names.length)} letters on average.`);
```

```
// 3. How do you access the last element of any array?
//      -  You use the index of the array length - 1. => array[array.length - 1].
//      -  This is because the index starts with 0, while the length begins counting at 1.
//      -  So array.length = 3, array[2] will be the last element
```

```
// 4. How do you access the first element of any array?
//      -  You use index 0. All arrays will begin with 0 index. => array[0] will be the first element.
```

PROMINEO TECH

```
// 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

// I would usually .map() this. However, using a loop, I first set nameLengths to an empty array. I then iterate through each
// element of the names array, pushing the name.length of that element to the nameLenths array.
// namesLengths = []; names = [ 'Sam', 'Tommy', 'Tim', 'Sally', 'Buck', 'Bob' ] => loop => nameLengths = [ 3, 5, 3, 5, 4, 3 ];
const nameLengths = [];
for (firstName of names) {
    nameLengths.push(firstName.length);
}

console.log('nameLengths array =>');
console.log(nameLengths); // => [ 3, 5, 3, 5, 4, 3 ]
```

```
// 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.

// As before, I itialize sumLengths = 0. Then I loop through each element in the array and add the name.length for each to sumLengths. Then, I print sumLenths to the console.
// nameLengths = [ 3, 5, 3, 5, 4, 3 ] => loop => ( 3 + 5 + 3 + 5 + 4 + 3 ) = 23;
let sumLengths = 0;
for (nameLength of nameLengths) {
    sumLengths += nameLength;
}

console.log(`Total numbers of letters in array = ${sumLengths}`);
```

```
// 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').

// first, I create a function that takes in 2 parameters and set the parameters to word and n. Then, I set string to an empty string.
// Next, I loop n number of times. For each loop, I add 1 instance of the word parameter to the string variable.
// Finally, I log the new string. string = ''; word = 'Hello'; n = 3; => loop => string = 'Hello'+'Hello'+'Hello' = 'HelloHelloHello'

const concatenate = (word, n) => {
    let string = ''
    for (let i = 0; i < n; i++) {
        string += word;
    } return string;
};

console.log(concatenate('Hello', 3));

// ...or you can just use the repeat method to get the same result.

console.log(`.repeat() method: ${'Hello'.repeat(3)}`);
```

```
// 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).

// First I create a function that takes 2 parameters, and set them to firstName and lastName.
// I then return a template literal, using `${firstName} ${lastName}`.
// firstName = 'David' lastName = 'Stringer' => function => 'David' 'Stringer'
const firstAndLast = (firstName, lastName) => {
    // logging to check result on console
    console.log(`${firstName} ${lastName}`);
    return (`${firstName} ${lastName}`);
};

firstAndLast('David', 'Stringer');
```

PROMINEO TECH

```
// 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.

// I create a function that takes in an array as the only parameter.
// Then, I reduce through the array, with sum as the previousValue, num as the currentValue, and 0 as the intial value.
// This iterates through each element, adding each element to sum each loop.
// testArray1 = [ 1, 2, 3, 4, 5 ] => function => 1 + 2 + 3 + 4 + 5 = 15 > 100 = false;
// testArray1 = [ 50, 50, 50, 50 ] => function => 50 + 50 + 50 + 50 = 200 > 100 = true;
const testArray1 = [ 1, 2, 3, 4, 5 ];
const testArray2 = [ 50, 50, 50, 50 ];


const isGreaterThan100 = (array) => {
    console.log(`is sum greater than 100? : ${(array.reduce((sum, num) => sum + num, 0)) > 100}`)
    return (array.reduce((sum, num) => sum + num, 0)) > 100;
}

isGreaterThan100(testArray1); // => false
isGreaterThan100(testArray2); // => true
```

```
// 10. Write a function that takes an array of numbers and returns the average of all the elements in the array.

// First I create a function that takes one array, named numbers, as it's only parameter. I then use an the reduce() method to
// sum through each number as it loops, then divide the sum by the length for the average.
// testArray3 = [ 1, 2, 3, 4, 5 ] => function => (1 + 2 + 3 + 4 + 5 ) / testArray3.length = 15 / 5 = 3
// testArray4 = [ 14.40, 7.25, 28.99, 21.23 ] => function => 14.40 + 7.25 + 28.99 + 21.23 / testArray4.length = 71.87 / 4 = 17.97

const testArray3 = [ 1, 2, 3, 4, 5 ];
const testArray4 = [ 14.40, 7.25, 28.99, 21.23 ];

const averageNumber = (array)=> {
    let arrayAverage = array.reduce((sum, num) => (sum + num / array.length), 0);
    return arrayAverage;
};

console.log(`The average is ${averageNumber(testArray3)}`); // => 'The average is 3.00'
console.log(`The average is ${averageNumber(testArray4)}`); // => 'The average is 17.9675'
```

```
// 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.

// ** Note ** It says write "a" function, but I did borrow the previous function for averaging
// an array, to avoid reusing code. If you wanted just 1 function, I would enter the logic of the
// averageNumber function in the conditional, for both array, but it looks messier that way.
// -> if (array1.reduce((sum,num) => (sum + num / array.length), 0) > array2.reduce((sum,num)
// => (sum + num / array.length), 0)).

// first, I create a function that takes 2 arrays as its parameters. Then I make a conditional
// if the averageNumber of array1 > averageNumber of array 2, return true. Return false otherwise.
const testArray5 = [ 1, 2, 3, 4, 5 ];
const testArray6 = [ 2, 3, 4, 5 ];

const isFirstArrayLarger = (array1, array2) => {
    if (averageNumber(array1) > averageNumber(array2)) {
        return true;
    } else {
        return false;
    }
}

console.log(isFirstArrayLarger(testArray5, testArray6)); // => false
console.log(isFirstArrayLarger(testArray6, testArray5)); // => true
```

PROMINEO TECH

```
// 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.

// First, I create a function function with 2 parameters, isHotOutside, and moneyInPocket. I expect a boolean for the first, a number for the second.
const willBuyDrink = (isHotOutside, moneyInPocket) => {

    // if statement that checks if BOTH conditions are true, depending on the input parameters
    // if both meet the conditions, then return true, otherwise, return false.
    if (isHotOutside === true && moneyInPocket > 10.50) {
        console.log(`I will buy a drink`);
        return true;
    } else {
        console.log(`I will not buy a drink`);
        return false;
    }
}

// testing all conditional possibilities
willBuyDrink(true, 11.00); // => true (isHotOutside === true AND more than 10.50 moneyInPocket)
willBuyDrink(true, 10.50); // => false (isHotOutside === true BUT less than 10.50 moneyInPocket)
willBuyDrink(false, 11.00); // => false (isHotOutside !== true, next statement not reached, both cannot be true)
willBuyDrink(false, 10.50); // => false (isHotOutside !== true, next statement not reached, both cannot be true)
```

```
// 13.  Create a function of your own that solves a problem. In comments, write what the function does and why you created it.

// variable for my daily calorie goal
const calorieGoal = 2000;

// variables for each meal of the day; would be dynamic in real use
const breakfast = 550;
const lunch = 400;
const dinner = 800;
const snacks = 300;
const dailyCalorieIntake = breakfast + lunch + dinner + snacks;

// this function compares my actual calorie intake with my calorie goal, and logs a message that corresponds to the result.
// this would be returned rather than logged in a real app, but I am not using any DOM functionality yet.
const isWithinCalorieGoal = (dailyCalorieIntake, calorieGoal) => {

    // initial annoucement of how many calories the user had today.
    console.log(`Today I consumed ${dailyCalorieIntake} calories`);

    // sets the calorieDifference variable = to intake minus goal, to find how far off the goal the user is.
    let calorieDifference = dailyCalorieIntake - calorieGoal;

    // 3 different outcomes, depending 1st on if the user is within 200 calories of goal. if so, function is done there,
    // with a success message.
    if (Math.abs(calorieDifference) <= 200) {
        console.log(`I was within ${Math.abs(calorieDifference)} calories of my goal. Great job!`);

    // if ! within 200 calories, then returns a message based on if the user is over or under their calorie goal.
    } else if (calorieDifference < 0) {
        console.log(`I was ${Math.abs(calorieDifference)} calories under my goal. I will make sure to eat a bit more tomorrow!`);
    } else if (calorieDifference > 0) {
        console.log(`I was ${Math.abs(calorieDifference)} calories over my goal. I will try to eat a little less tomorrow!`);
    }
};

// executing function.
isWithinCalorieGoal(dailyCalorieIntake, calorieGoal); // => ` I was within 50 calories of my goal. Great job!

// I made this function because I am wanting to make a meal planner/calorie counter app for myself. I don't eat many different types of food during
// the week, so I can write functions to calculate all my different meal macros and calories, and automate my meal plan. I will add a
// journal/history section to keep track of how I feel on different food/macro/calorie combos. I can even ballpark my shopping list,
// depending on how accurate I am on my ingredients and portions.. I am sure once I get in there and optimize and refactor,
// the code will improve. This is just a rough draft at comparing results to goals.
```

Screenshots of Running Application:

```
▶ (8) [3, 9, 23, 64, 2, 8, 28, 93]                                  Week3_Coding_Assignment.js:6
90                                                                  Week3_Coding_Assignment.js:18
After push, 37                                                      Week3_Coding_Assignment.js:26
The average age is 30                                               Week3_Coding_Assignment.js:36
Each name contains 3.8333333333333335 letters on average.           Week3_Coding_Assignment.js:53
nameLengths array =>                                                Week3_Coding_Assignment.js:77
▶ (6) [3, 5, 3, 5, 4, 3]                                            Week3_Coding_Assignment.js:78
Total numbers of letters in array = 23                              Week3_Coding_Assignment.js:90
HelloHelloHello                                                     Week3_Coding_Assignment.js:107
.repeat() method: HelloHelloHello                                   Week3_Coding_Assignment.js:111
David Stringer                                                      Week3_Coding_Assignment.js:122
is sum greater than 100? : false                                    Week3_Coding_Assignment.js:140
is sum greater than 100? : true                                     Week3_Coding_Assignment.js:140
The average is 3                                                    Week3_Coding_Assignment.js:162
The average is 17.9675                                              Week3_Coding_Assignment.js:163
false                                                               Week3_Coding_Assignment.js:185
true                                                                Week3_Coding_Assignment.js:186
I will buy a drink                                                  Week3_Coding_Assignment.js:198
I will not buy a drink                                              Week3_Coding_Assignment.js:201
I will not buy a drink                                              Week3_Coding_Assignment.js:201
I will not buy a drink                                              Week3_Coding_Assignment.js:201
Today I consumed 2050 calories                                      Week3_Coding_Assignment.js:230
I was within 50 calories of my goal. Great job!                     Week3_Coding_Assignment.js:238
```