



Intro to JavaScript Week 3 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

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 20.
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:



```

1 // Week3CodingAssignment.js
2
3 // 1. Write a function called sum that takes an array of numbers and returns the sum of all elements.
4 // Example: sum([1, 2, 3, 4, 5]) should return 15.
5
6 // 2. Write a function called max that takes an array of numbers and returns the maximum value.
7 // Example: max([1, 2, 3, 4, 5]) should return 5.
8
9 // 3. Write a function called min that takes an array of numbers and returns the minimum value.
10 // Example: min([1, 2, 3, 4, 5]) should return 1.
11
12 // 4. Write a function called average that takes an array of numbers and returns the average value.
13 // Example: average([1, 2, 3, 4, 5]) should return 3.
14
15 // 5. Write a function called isEven that takes a number and returns a boolean indicating if it is even.
16 // Example: isEven(2) should return true, isEven(3) should return false.
17
18 // 6. Write a function called isOdd that takes a number and returns a boolean indicating if it is odd.
19 // Example: isOdd(2) should return false, isOdd(3) should return true.
20
21 // 7. Write a function called isPrime that takes a number and returns a boolean indicating if it is prime.
22 // Example: isPrime(2) should return true, isPrime(4) should return false.
23
24 // 8. Write a function called isPalindrome that takes a string and returns a boolean indicating if it is a palindrome.
25 // Example: isPalindrome("racecar") should return true, isPalindrome("hello") should return false.
26
27 // 9. Write a function called isAnagram that takes two strings and returns a boolean indicating if they are anagrams.
28 // Example: isAnagram("listen", "silent") should return true, isAnagram("hello", "world") should return false.
29
30 // 10. Write a function called isArmstrong that takes a number and returns a boolean indicating if it is an Armstrong number.
31 // Example: isArmstrong(153) should return true, isArmstrong(123) should return false.
32
33 // 11. Write a function called isHappy that takes a number and returns a boolean indicating if it is a happy number.
34 // Example: isHappy(1) should return true, isHappy(4) should return false.
35
36 // 12. Write a function called isLucky that takes a number and returns a boolean indicating if it is a lucky number.
37 // Example: isLucky(7) should return true, isLucky(4) should return false.
38
39 // 13. Write a function called isBeautiful that takes a number and returns a boolean indicating if it is a beautiful number.
40 // Example: isBeautiful(1) should return true, isBeautiful(2) should return false.
41
42 // 14. Write a function called isUgly that takes a number and returns a boolean indicating if it is a ugly number.
43 // Example: isUgly(6) should return true, isUgly(14) should return false.
44
45 // 15. Write a function called isPowerOfTwo that takes a number and returns a boolean indicating if it is a power of two.
46 // Example: isPowerOfTwo(2) should return true, isPowerOfTwo(3) should return false.
47
48 // 16. Write a function called isPowerOfThree that takes a number and returns a boolean indicating if it is a power of three.
49 // Example: isPowerOfThree(3) should return true, isPowerOfThree(4) should return false.
50
51 // 17. Write a function called isPowerOfFour that takes a number and returns a boolean indicating if it is a power of four.
52 // Example: isPowerOfFour(4) should return true, isPowerOfFour(5) should return false.
53
54 // 18. Write a function called isPowerOfFive that takes a number and returns a boolean indicating if it is a power of five.
55 // Example: isPowerOfFive(5) should return true, isPowerOfFive(6) should return false.
56
57 // 19. Write a function called isPowerOfSix that takes a number and returns a boolean indicating if it is a power of six.
58 // Example: isPowerOfSix(6) should return true, isPowerOfSix(7) should return false.
59
60 // 20. Write a function called isPowerOfSeven that takes a number and returns a boolean indicating if it is a power of seven.
61 // Example: isPowerOfSeven(7) should return true, isPowerOfSeven(8) should return false.
62
63 // 21. Write a function called isPowerOfEight that takes a number and returns a boolean indicating if it is a power of eight.
64 // Example: isPowerOfEight(8) should return true, isPowerOfEight(9) should return false.
65
66 // 22. Write a function called isPowerOfNine that takes a number and returns a boolean indicating if it is a power of nine.
67 // Example: isPowerOfNine(9) should return true, isPowerOfNine(10) should return false.
68
69 // 23. Write a function called isPowerOfTen that takes a number and returns a boolean indicating if it is a power of ten.
70 // Example: isPowerOfTen(10) should return true, isPowerOfTen(11) should return false.
71
72 // 24. Write a function called isPowerOfEleven that takes a number and returns a boolean indicating if it is a power of eleven.
73 // Example: isPowerOfEleven(11) should return true, isPowerOfEleven(12) should return false.
74
75 // 25. Write a function called isPowerOfTwelve that takes a number and returns a boolean indicating if it is a power of twelve.
76 // Example: isPowerOfTwelve(12) should return true, isPowerOfTwelve(13) should return false.
77
78 // 26. Write a function called isPowerOfThirteen that takes a number and returns a boolean indicating if it is a power of thirteen.
79 // Example: isPowerOfThirteen(13) should return true, isPowerOfThirteen(14) should return false.
80
81 // 27. Write a function called isPowerOfFourteen that takes a number and returns a boolean indicating if it is a power of fourteen.
82 // Example: isPowerOfFourteen(14) should return true, isPowerOfFourteen(15) should return false.
83
84 // 28. Write a function called isPowerOfFifteen that takes a number and returns a boolean indicating if it is a power of fifteen.
85 // Example: isPowerOfFifteen(15) should return true, isPowerOfFifteen(16) should return false.
86
87 // 29. Write a function called isPowerOfSixteen that takes a number and returns a boolean indicating if it is a power of sixteen.
88 // Example: isPowerOfSixteen(16) should return true, isPowerOfSixteen(17) should return false.
89
90 // 30. Write a function called isPowerOfSeventeen that takes a number and returns a boolean indicating if it is a power of seventeen.
91 // Example: isPowerOfSeventeen(17) should return true, isPowerOfSeventeen(18) should return false.
92
93 // 31. Write a function called isPowerOfEighteen that takes a number and returns a boolean indicating if it is a power of eighteen.
94 // Example: isPowerOfEighteen(18) should return true, isPowerOfEighteen(19) should return false.
95
96 // 32. Write a function called isPowerOfNineteen that takes a number and returns a boolean indicating if it is a power of nineteen.
97 // Example: isPowerOfNineteen(19) should return true, isPowerOfNineteen(20) should return false.
98
99 // 33. Write a function called isPowerOfTwenty that takes a number and returns a boolean indicating if it is a power of twenty.
100 // Example: isPowerOfTwenty(20) should return true, isPowerOfTwenty(21) should return false.

```

```
1 // 1. Write a function that takes two parameters, word and n, as arguments and returns the word concatenated to itself n number of times.
2
3 function wordRepeat(word, n) {
4   if (n < 1 || n > 10) {
5     return word.repeat(1);
6   }
7   console.log(arguments[0], n);
8
9   // 2. Write a function that takes two parameters, firstWord and lastWord, and returns a full name (the full name should be the first and the last
10  // name separated by a space).
11  function createInvitation(firstWord, lastWord) {
12    return firstWord + ' ' + lastWord;
13  }
14  let fullName = createInvitation('John', 'Doe');
15  console.log(fullName);
16
17 // 3. 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.
18 function isGreaterThan100(arr) {
19   let sum = 0;
20   for (let i = 0; i < arr.length; i++) {
21     sum += arr[i];
22   }
23   console.log('Sum of array after processing the first and last elements used to determine if the sum of all numbers is greater than 100: ' + sum);
24   return sum > 100;
25 }
26 console.log(isGreaterThan100([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]));
27
28 // 4. Write a function that takes an array of numbers and returns the average of all the elements in the array.
29 function arrayAverage(arr) {
30   let sum = 0;
31   for (let i = 0; i < arr.length; i++) {
32     sum += arr[i];
33   }
34   let average = sum / arr.length;
35   return sum / arr.length;
36 }
37 let avg = arrayAverage([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]);
38 console.log('The total average of all elements in the numbers array: ' + avg);
39
40 // 5. 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
41 // average of the elements in the second array.
42 function arrayAverage(arr1, arr2) {
43   let sum1 = 0;
44   for (let i = 0; i < arr1.length; i++) {
45     sum1 += arr1[i];
46   }
47   let total1 = arr1.length;
48   return sum1 / total1;
49 }
50 let avg1 = arrayAverage(arr1, arr2);
51 console.log('Avg array returns average of total: ' + avg1);
52
53 function greaterAvg(arr1, arr2) {
54   if (avg1 > avg2) {
55     return true;
56   }
57 }
58 console.log(greaterAvg(arr1, arr2));
59
60 // 6. Write a function called validateEmail that takes a function isValidEmail, and a number emailCount, and returns true if it has called
61 // validateEmail more than 10 times.
62 function validateEmail(isValidEmail, emailCount) {
63   if (isValidEmail === null || !isValidEmail) {
64     return true;
65   }
66 }
67 console.log(validateEmail('test', 10));
68
69 // 7. Create a function of your own that solves a problem. In comments, write what the function does and why you created it.
70 function solveProblem(arr) {
71   // calculate sum of array
72   let sum = 0;
73   for (let i = 0; i < arr.length; i++) {
74     sum += arr[i];
75   }
76 }
77 console.log(solveProblem([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]));
```



Screenshots of Running Application:

The screenshot shows a Visual Studio Code editor window titled "Promineo Tech Week 3 Coding Assignment". The file path is "C:/Users/jacob/.vscode/Promineo%20Tech%20Week%203%20Coding%20Assignment/index.html". The editor displays a JavaScript file named "Week3CodingAssignment.js" with the following code:

```
const ages = [3, 9, 23, 64, 2, 8, 28, 93];
const firstElement = ages[0];
const lastElement = ages[ages.length - 1];
const subtractionSolution = (93 - 3) * 3;
const arrayAfterAddingNewAge = [...ages, 42];
const subtractionSolutionAfterAddingNewAge = (42 - 3) * 3;
const elementsInAgesArray = ages;
const calculatedAverageOfAgesArray = (ages.reduce((sum, age) => sum + age, 0) / ages.length).toFixed(2);
const names = ["San", "Tommy", "Tim", "Sally", "Buck", "Bob"];
const calculatedAverageNumberOfLettersPerName = (names.reduce((sum, name) => sum + name.length, 0) / names.length).toFixed(2);
const namesTogether = "San Tommy Tim Sally Buck Bob";
const accessedAndRemovedLastNumberInAgesArray = ages.pop();
const accessedAndRemovedLastNameInNamesArray = names.pop();
const accessedAndRemovedFirstNumberInAgesArray = ages.shift();
const accessedAndRemovedFirstNameInNamesArray = names.shift();
const namesLengthArray = names.map(name => name.length);
const calculatedSumOfAllElementsInNamesLengthArray = namesLengthArray.reduce((sum, length) => sum + length, 0);
const helloHelloHello = "HelloHelloHello";
const fullNameFunction = () => { return "Jacob Studer"; };
const totalOfAgesArrayAfterRemovingFirstAndLastElementsUsedToDetermineIfSumOfAllNumbersIsGreaterThan100 = ages.slice(1, -1).reduce((sum, age) => sum + age, 0);
const true = true;
const theTotalAverageOfAllElementsInNamesLengthsArray = (namesLengthArray.reduce((sum, length) => sum + length, 0) / namesLengthArray.length).toFixed(2);
const agesArrayCurrentAverageOfTotal = (ages.reduce((sum, age) => sum + age, 0) / ages.length).toFixed(2);
const true = true;
const true = true;
const setAlarm = () => { return "Set Alarm!"; };
```

The console output shows the following log messages:

```
Ages Array: 3,9,23,64,2,8,28,93
First element of array: 3
Last element of array: 93
Subtraction solution (93 - 3 * 3): 90
Array after adding new age: 3,9,23,64,2,8,28,93,42
Subtraction solution after adding new age (42 - 3 * 3): 39
elements in ages array: 3,9,23,64,2,8,28,93,42
Calculated average of ages array (value of elements added together 272 / 9 number of elements): 30.22222222222222
Names Array: San,Tommy,Tim,Sally,Buck,Bob
Calculated average number of letters per name in the names array (total number of letters 23 / 6 number of names): 3.8333333333333335
Names together: San Tommy Tim Sally Buck Bob
Accessed and removed last number in ages array: 42
Accessed and removed last name in names array: Bob
Accessed and removed first number in ages array: 3
Accessed and removed first name in names array: San
Names Length Array: 3,5,3,5,4,3
Calculated sum of all the elements in the namesLength array: 23
HelloHelloHello
Full name function: Jacob Studer
Total of ages array after removing the first and last elements used to determine if the sum of all numbers is greater than 100: 227
true
The total average of all elements in the namesLengths array: (avg2) 3.8333333333333335
Ages array current average of total: (avg1) 32.42857142857143
true
true
Set Alarm!
```

URL to GitHub Repository: <https://github.com/JacobStuder/Promineo-Tech-Week-3-coding-Assignment.git>