

Assignment 3: Concept of Programming

Introduction

This assignment is designed to test your understanding of conditional and looping statements, recursion, and functions in JavaScript. You will have to write code snippets to solve various problems using these concepts. You will also have to explain your logic and output for each code snippet.

Instructions

- Write your code in a text editor like Notepad or Visual Studio Code.
- Save your file with a .js extension (for example, assignment1.js).
- You can use a web browser such as Chrome or Firefox to run your code. Open the browser's developer tools (Ctrl+Shift+I) and go to the console tab. Then drag and drop your file into the browser window or use the File > Open File option to select your file. You should see the output of your code in the console.
- Alternatively, you can use an online tool such as <https://replit.com/languages/javascript> to write and run your code.
- You have to complete this assignment individually.
- You have to submit your assignment as a single PDF file that contains your code snippets, explanations, and outputs.
- You have to use proper indentation, comments, and naming conventions for your code snippets.
- You have to cite any sources that you use for reference or inspiration.
- You have to follow the marking scheme given below for each question.

Questions

Question 1: Calculate the average of numbers using Array

Write a function that takes an array of numbers as a parameter and returns the average of those numbers.

For example:

```
function average(numbers) {  
    // write your code here  
}  
  
console.log(average([1, 2, 3])); // 2  
console.log(average([10, 20, 30])); // 20  
console.log(average([5])); // 5
```

Question 2: Reverse an array

Write a function that takes an array as a parameter and returns a new array with the elements in reverse order.

For example:

```
function reverse(array) {  
  // write your code here  
}  
  
console.log(reverse([1, 2, 3])); // [3, 2, 1]  
console.log(reverse(['a', 'b', 'c'])); // ['c', 'b', 'a']  
console.log(reverse([])); // []
```

Question 3: Sort an array in ascending order

Write a function that takes an array of numbers as a parameter and returns a new array with the elements sorted in ascending order.

For example:

```
function sort(array) {  
  // write your code here  
}  
  
console.log(sort([5, 2, 4, -2])); // [-2, 2, 4, 5]
```

Question 4: Convert char Array to String

Write a function that takes an array of characters as a parameter and returns a string that contains those characters.

For example:

```
function toString(array) {  
  // write your code here  
}  
  
console.log(toString(['h', 'e', 'l', 'l', 'o'])); // "hello"  
console.log(toString(['J', 'a', 'v', 'a', 'S', 'c', 'r', 'i', 'p', 't']));  
// "JavaScript"  
console.log(toString([])); // ""
```

Question 5: Add two Matrix using Multi-dimensional Arrays

Write a function that takes two matrices (represented as two-dimensional arrays) as parameters and returns a new matrix that is the result of adding them.

For example:

```
function addMatrix(matrix1, matrix2) {  
  // write your code here  
}  
  
console.log(addMatrix([[1, 2], [3, 4]], [[5, 6], [7, 8]])); // [[6, 8], [10,
```

```
12]]  
console.log(addMatrix([[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]));  
// [[8, 10, 12], [14, 16, 18]]  
console.log(addMatrix([[1]], [[2]])); // [[3]]
```

Question 6: Sort strings in alphabetical order

Write a function that takes an array of strings as a parameter and returns a new array with the elements sorted in alphabetical order.

For example:

```
function sortStrings(array) {  
  // write your code here  
}  
  
console.log(sortStrings(['apple', 'banana', 'cherry'])); // ['apple',  
  'banana', 'cherry']  
console.log(sortStrings(['Zebra', 'elephant', 'Lion'])); // ['Lion',  
  'Zebra', 'elephant']  
console.log(sortStrings(['Hello', 'world'])); // ['Hello', 'world']
```

Question 7: Find out the highest and second highest numbers in an array

Write a function that takes an array of numbers as a parameter and returns an array with two elements: the highest and the second highest numbers in the array.

For example:

```
function findHighest(array) {  
  // write your code here  
}  
  
console.log(findHighest([1, 2, 3])); // [3, 2]  
console.log(findHighest([10, 20, 30])); // [30, 20]  
console.log(findHighest([5])); // [5]
```

Question 8: Concatenate two arrays

Write a function that takes two arrays as parameters and returns a new array that is the result of concatenating them.

```
For example:  
function concat(array1, array2) {  
  // write your code here  
}  
  
console.log(concat([1, 2], [3, 4])); // [1, 2, 3, 4]  
console.log(concat(['a', 'b'], ['c', 'd'])); // ['a', 'b', 'c', 'd']  
console.log(concat([], [])); // []
```

End of Assignment

I hope this assignment helps you learn about arrays in JavaScript. If you have any questions or feedback, please let me know. 😊

Appendix

Arrays

Introduction

An array is an object that can store multiple values at once.

For example

```
const words = ['hello', 'world', 'welcome'];
```

Here, words is an array. The array is storing 3 values.

You can create an array using two ways:

1. Using an **array literal**

The easiest way to create an array is by using an array literal [].

For example

```
const words = ['hello', 'world', 'welcome'];
```

2. Using the **new keyword**

You can also create an array using JavaScript's new keyword.

```
const array2 = new Array("eat", "sleep");
```

In both of the above examples, we have created an array having two elements.

Note: It is recommended to use an array literal to create an array.

You can access elements of an array using indices (0, 1, 2 ...).

For example,

```
const myArray = ['h', 'e', 'l', 'l', 'o'];
```

```
// first element
console.log(myArray[0]); // "h"

// second element
console.log(myArray[1]); // "e"
```

Note: Array's index starts with 0, not 1.

You can also add elements or change the elements by accessing the index value. For example:

```
let dailyActivities = [ 'eat', 'sleep'];

// this will add the new element 'exercise' at the 2 index
dailyActivities[2] = 'exercise';

console.log(dailyActivities); // ['eat', 'sleep', 'exercise']
```

You can use the built-in methods `push()` and `unshift()` to add elements to an array.

The `push()` method adds an element at the end of the array. For example:

```
let dailyActivities = ['eat', 'sleep'];

// add an element at the end
dailyActivities.push('exercise');

console.log(dailyActivities); // ['eat', 'sleep', 'exercise']
```

The `unshift()` method adds an element at the beginning of the array. For example:

```
let dailyActivities = ['eat', 'sleep'];

//add an element at the start
dailyActivities.unshift('work');

console.log(dailyActivities); // ['work', 'eat', 'sleep']
```

You can use the `pop()` method to remove the last element from an array.

```
let dailyActivities = ['eat', 'sleep'];

// remove the last element
dailyActivities.pop();
```

```
console.log(dailyActivities); // ['eat']
```

An array can hold values of mixed types. For example,

```
const newData = ['work', 'exercise', 1, true];
```

You can also store arrays, functions and other objects inside an array. For example

```
const newData = [  
  {'task1': 'exercise'},  
  [1, 2, 3],  
  function hello() { console.log ('hello')}  
];
```