Basics of JavaScript 2 - Assignment

Introduction

This week you will be building small programs in JavaScript to assess your logic building skills.

You will be working with the basic concepts like anonymous functions, IIFE, objects, callback functions, JSON etc.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

Your task is as following -

1. **Task 1 (3 points each)**
   - 1.1. Define and invoke an anonymous function which takes a number and returns its square. If input is 2, output should be 4.
   - 1.2. Define an IIFE function which takes a personName as input and displays a greeting message containing the personName. If input is "Harry", output should be "Hello Harry, Welcome to Great Learning!"
   - 1.3. Define a global array variable containing 3 numbers. Define a function which increments the value of each of the elements of this array by 2. Display the array after you have invoked this function.
   - 1.4. Create a "course" object having following information - courseName as "Computer Science", durationInMonths as 24, level as "Beginner". Display object information. Change the value of level as "Intermediate". Display object information again.
   - 1.5. Given an array "students" which is a collection of JavaScript objects where each object consists of information regarding one student. Write a code to iterate through each of these objects and extract the first name and last name of each student.
   - 1.6. Given a function doubleNumber which takes a number as an argument and returns its double value. Write a function which takes 2 arguments- 1st an array of numbers and 2nd the doubleNumber function as a callback function This function should iterate through each of the array numbers and use the doubleNumber function to double it. In the end it should display the updated array. We are not supposed to create a new array. We just need to update the existing array.
   - 1.7. Implement a function called `multiplyBy` that multiplies a number by a specific factor using an IIFE (Immediately Invoked Function Expression). Hence, the IIFE function should return a function which should do the multiplication operation.

1.8. Using the `apply` method, write a function that finds the maximum number in an array. You can use Math class's built-in max() method for this task.

1.9. Declare an object named "car" with an empty object as its initial value. Add the properties "make" and "model" with values "Toyota" and "Camry" respectively.

1.10. Given an array "students" which is a collection of JavaScript objects where each object consists of information regarding one student. Define a function displayByKey() which takes this array object and a keyName(as string) as arguments and displays the value of the key for each of the JavaScript objects.

2. Task 2 (3 points each)

2.1. Define 2 functions - 1st function named as checkEven which will check if the number passed is even or not. 2nd function named filterEvens which will take an array of numbers and the checkEven function as arguments. This filterEvens function will filter out only even numbers using the checkEven function and generate a new array of the even numbers and return it.

2.2. Write an IIFE that calculates the factorial of a given number and immediately logs the result to the console.

2.3. The product1 object (which is already given) consists of title, price and category information of Nike Shoes. The description() function describes the product using its properties. Your task is to create a product2 object which consists of the title, price and category information of Sony TV. Next, use the call() function to invoke the description() method of product1 on product2. This should display the details of product2(Sony TV) on the console.

2.4. Given an array of person objects, define a function to find the oldest person object and return it.

2.5. Create a function that calculates the sum of an array using IIFE function.

2.6. Write a function **printContext** that, when invoked, logs **this** keyword to the console. Then, demonstrate how the context of a function can change when calling it with different objects using the call method.

2.7. Create a function **multiply** that takes two parameters and returns their product. Use the bind method to create a new function "double" that multiplies a single parameter by 2.

2.8. Create an object person with properties name and age. Write a function "introduce" that logs a message introducing the person. Then, use the call method to invoke the introduce function with the person object as the context.

2.9. Write a higher order function **createMultiplier** that takes a factor as an argument and returns another function that multiplies a number by that factor.

2.10. Write a function called "calculate" that adds two numbers and assign a property "description" to it with a string describing what the function does. Then, access and log this property.

Program Organization

- You will be getting a zip folder named **Basics of JS2 Assessment_For Coders** containing two files **Task1.js** and **Task2.js**.
- All the tasks mentioned above for Task 1 should be completed in Task1.js file.
- All the tasks mentioned above for Task 2 should be completed in Task2.js file.

Evaluation Rubric

Total Project Points: 60

- Correctness:
  Correctness of implementation
  - Problem statement - Task 1 (**50%**) : **30 Points**
  - Problem statement - Task 2 (**50%**) : **30 Points**

Program Instructions

- The **Basics of JS2 Assessment_For Coders** folder should have **Task1.js** and **Task2.js** files.
- The **Basics of JS2 Assessment_For Coders** folder should be compressed and submitted as the zip/rar folder.
- Project will not be evaluated if the submitted project is not in the zip/rar format.