CADT IDT

W4 - PRACTICE

JS – ES6 Functions

At the end of this practice, you should be able to…

- ✓ Use function **default argument**
- ✓ Use **arrow functions**
- ✓ Pass **function as a parameter**
- ✓ Use **Destructuring syntax**
- ✓ Use **spread syntax** to clone objects

How to work?

BEFORE THE PRACTICE

✓ First watch and understand the **following pages and videos**:
Function basics, default parameter, arrow function, destructuring, spread operator Video 1,

✓ Then complete the **following quiz** (*you can re-do it until you have 100% score*)
https://forms.gle/FtUHEsoRnV8eBsTx7

DURING THE PRACTICE

✓ To start the practice. **download the start code** from Google classroom

How to submit?

✓ **Create a repository on GitHub** with the name of this practice:
Ex: `C1-S2-PRACTICE`

✓ **Push your final code** on this GitHub repository (if you are lost, follow this tutorial)

✓ Finally, submit on **Google classroom** your GitHub repository URL
Ex: `https://github.com/thebest/ C2-S1-PRACTICE.git`

UNDERSTAND THE CONCEPTS…

Before starting exercises, explain - *in your own words* – the benefit of some concepts in JS

What is the benefit of	You explanation
Benefit of function **default argument**	*No need to manually check if a parameter is undefined, improve code readability*
Benefit of **arrow functions**	*Allow a shorter syntax for function expressions.*
Benefit of the **destructuring syntax**	*Helps to unpack values from array and objects into distinct variables*
Benefit of the spread operator	*Allows us to create copies of arrays or objects without modifying the originals.*

EXERCISE 1

In the provided code, there's an array named `shoppingCart` representing a user's shopping cart with items containing 'name', 'price', and 'quantity' properties.
The current code calculates the total price of items in the shopping cart without using functions.

THE PROBLEM

This code works only for this specific shopping cart. But we want to be able to compute the total price for any shopping cart

YOUR JOB

Your task is to refactor the code by extracting the logic for calculating the total **price into a reusable function.**

✓ Write a function named `calculateTotalPrice` that takes the shoppingCart array as a parameter and returns the total price.

✓ Check that your code still produces the same result

✓ Check that your code can work with many shopping carts

```
let shoppingCart = [
  { name: "Apples", price: 2.5, quantity: 3 },
  { name: "Bananas", price: 1.5, quantity: 2 },
  { name: "Oranges", price: 3, quantity: 1 },
];

// Calculate total price without using functions
let totalPrice = 0;
for (let item of shoppingCart) {
  totalPrice += item.price * item.quantity;
}
```

EXERCISE 2

The originalArray contains some elements.
The function updateArray takes an array, an index, and a new value as parameters, and updates the value at the specified index in the array.

THE PROBLEM

We want the original array to **remain unchanged**!

YOUR JOB

In the function updateArray, you need to use the spread operator (...) to clone the original array before making modifications.

```
let originalArray = [1, 2, 3, 4, 5];

function updateArray(array, index, newValue) {
  array[index] = newValue;
  return array;
}

let updatedArray = updateArray(originalArray, 2, 10);

console.log("Original array:", originalArray); // original array should not be
modified...
console.log("Updated array:", updatedArray);
```

EXERCISE 3

In the provided code, there's a function named findAverage that calculates the average of numbers in an array.
The findAverage function takes an array of numbers as a parameter.

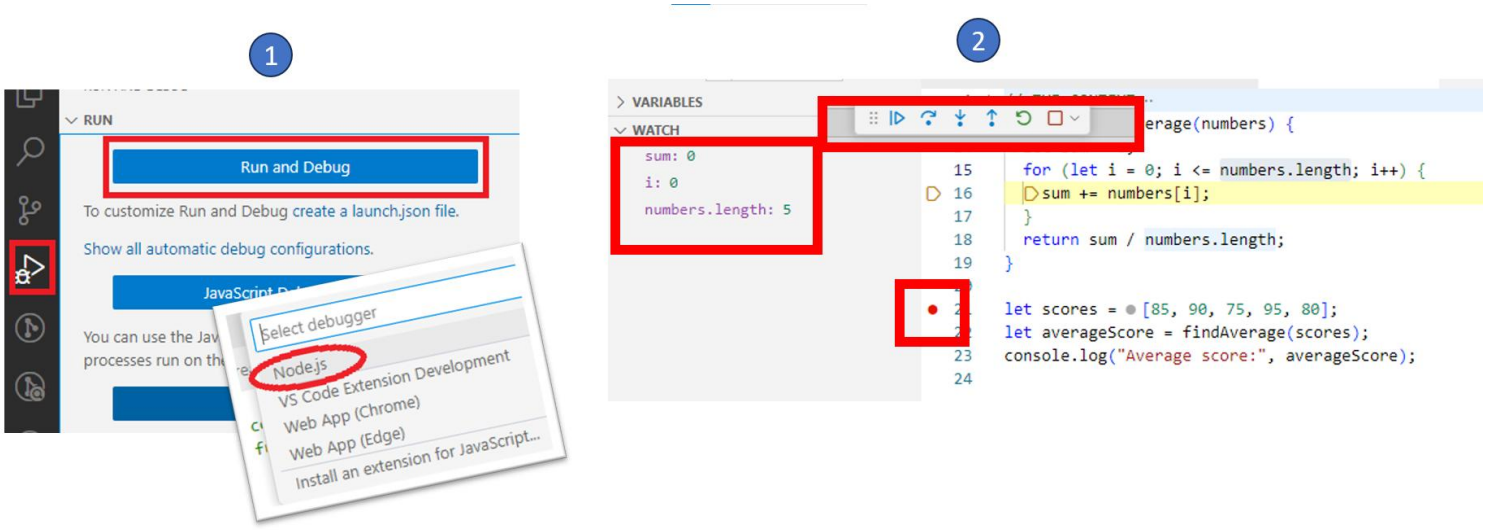
```
function findAverage(numbers) {
  let sum = 0;
  for (let i = 0; i <= numbers.length; i++) {
    sum += numbers[i];
  }
  return sum / numbers.length;
}
```

THE PROBLEM

There's a bug in the implementation of the findAverage function that causes it to **produce incorrect results** (it displays `NaN`)

YOUR JOB

Your task is to **identify and fix the bug** in the **findAverage** function to ensure that it correctly calculates the average of numbers in the array.



For this exercise, you should launch VScode debugger and diagnostic the bug, step by step

EXERCISE 4

In this exercise you will need to work with 3 array functions: map, filter, reduce
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reduce

YOUR JOB

Complete the 3 functions with the corresponding code. *Everything should ft in 1 line!*

```
// Write the function to filter out even numbers from an array
// You need to use the array.filter() function
function filterEvenNumbers(arr) {
  // TODO - Your code here (1 line)
}

// Function to map array elements to their squared values
// You need to use the array.map() function
function squareNumbers(arr) {
  //  TODO - Your code here (1 line)
}

// Function to compute the sum of array elements
// You need to use the array.reduce() function
function sumArray(arr) {
  //  TODO - Your code here (1 line)
}
```

EXERCISE 5

In the provided code, there's an array named students, containing objects representing students with their IDs, names, and grades.

```
let students = [
  { id: 1, name: "Trang", grade: "A" },
  { id: 2, name: "Hai", grade: "B" },
  { id: 3, name: "Linh", grade: "C" },
];
```

There's a function named updateStudentGrade() that takes three parameters:

- studentsArray the array of students)
- idToUpdate the ID of the student to update)
- newGrade the new grade to assign to the student).

```
function updateStudentGrade(studentsArray, idToUpdate, newGrade) { … }
```

YOUR JOB

Your task is to complete the updateStudentGrade function, following the steps.

Note : You will need to use the following elements : array.findIndex(), spread operator