CADT IDT

***W1** PRACTICE*

From C++ to JS

At the end of this practice, you can

- ✔ Run JS code
- ✔ Create **variables** and **constants**
- ✔ Call and define **functions**
- ✔ Use JS **loops** and **conditions**
- ✔ Manipulate **arrays**, **objects**, **strings**, **Boolean** and **numbers**

Get ready before this practice!

- ✔ **Read** the following documents to understand JS syntax:

  https://cstart.mines.edu/web/Day2/2-JavaScriptBasicSyntax.pdf
  https://www.integral-domain.org/lwilliams/mis462/JavaScript.pdf

  You can also go further with the following books:
  https://www.gurukultti.org/admin/notice/javascript.pdf
  https://www.w3schools.com/js/default.asp

- ✔ **Complete the quiz** (*you can re-do it until you have 100% score*)

How to submit this practice?

- ✔ **Complete** this document
- ✔ Once finished, join this document to the MS Team assignment and **turn it in**

*3 WAYS TO **RUN JS CODE***

For beginners

To start with, you can just connect to an **online JavaScript editor**, such as this one:
https://playcode.io/javascript

For front-end ninjas

Chrome or any other **Web Browser** can execute JavaScript code while loading HTML

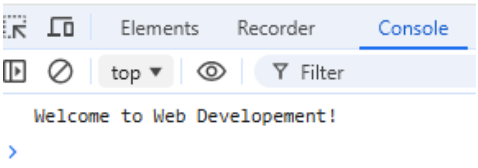
Just create a simple `index.html` file, that links to a `index.js` file:

```
<!DOCTYPE html>
<html>
<head>
    <title>Let s run JS on a Browser</title>
    <script src='index.js'></script>
</head>
<body>
</body>
</html>
```

Then just write some JS code, as example here, we print a message on the Browser console

```
// Example of JS code, printing on console
const courseName = "Web Developement";
console.log("Welcome to " + courseName + "!");
```

Finally open your index.html on a browser and check the console view



For back-end gurus

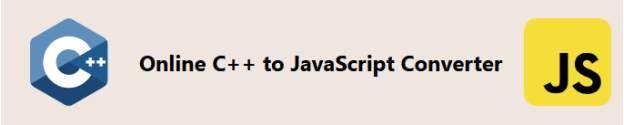
Node.js is also able to **execute JavaScript** code **outside a web browser.**
You will need first to install Node JS on your computer.

You can then just open a terminal on the folder containing your index.js file and run

```
node ./index.js
```

PART 1 - UNDERSTAND **JS SYNTAX**

Note: you can use the C++ to JS converter to compare C++ and JS syntax.



EXERCISE 1- TYPES, OUTPUTS

Analyze the differences between the provided C++ and JavaScript code.

C++	JS

`#include <iostream>`
`using namespace std;`
 	
`int main() {`
`  const int num = 5;`
`  for (int i = 0; i < num; i++) {`
`  cout << i << " ";`
`  }`
`  return 0;`
`}`
	`const num = 5;`
	`for (let i = 0; i < num; i++) {`
	`  console.log(i);`
	`}`

Q1 - What does the **const** key word mean in JS code?

Q2 - Why is it necessary **to specify the type** of variables in C++ but not in JavaScript?

Q3- How to **print in the console** in JS?

Q4- Is there any difference in the **loop syntax** between C++ and JS?

EXERCISE 2 - LOOPS, FUNCTIONS

C++	JS

`#include <iostream>`
`using namespace std;`
 	
`int calculateSum(int array[], int size) {`
`  int sum = 0;`
 	
`  for (int i = 0; i < size; i++) {`
`  // Add the calculation logic`
`  }`
`  return sum;`
`}`
 	
`int main() {`
`  int arr[] = {1, 2, 3, 4, 5};`
`  cout << calculateSum(arr, 5);`
`  return 0;`
`}`
	`function calculateSum(array) {`
	`  let sum = 0;`
	
	`  for (let i = 0; i < array.length; i++) {`
	`  // Complete the calculation logic`
	`  }`
	`  return sum;`
	`}`
	
	`let arr = [1, 2, 3, 4, 5];`
	`console.log(calculateSum(arr));`

Q1 - Complete the given 2 codes to compute the sum of all elements in an array

Q2 – Explain why the function calculateSum in JS code **does not have the size** parameter

EXERCISE 3 - CONDITIONS, EQUALITY

JS

```
function myFunction(min, max) {
  var result = "";
  for (let number = min; number <= max; number++) {
    if (number % 2 === 0) {
      result += number + " ";
    }
  }
  return result;
}
```

Q1 – Observe the above code

- Highlight all **variables in blue**
- Underline all **loops in red**
- Highlight all **conditions in green**

Q2 – What is the significance of the modulo operator % in these programs?

Q3 – What is the difference between === and == in JS? *Highlight the right answer*

4 == 9	TRUE / FALSE
4 == 4	TRUE / FALSE
4 == ‘’4”	TRUE / FALSE
4 === ‘’4”	TRUE / FALSE

Q4 – Why the function calculateSum **does not have the size** parameter in the JS code?

EXERCISE 4 – MEMORY ALLOCATION

Both codes are performing the same job:

C++

```cpp
#include <iostream>
using namespace std;

int main() {
    int size = 5;
    int* arr = new int[size]; // Dynamically allocate memory for an array
    for (int i = 0; i < size; i++) {
        arr[i] = i * 2; // Assign values
    }

    for (int i = 0; i < size; i++) {
        cout << arr[i] << " "; // Print values
    }
    delete[] arr; // Free allocated memory
    return 0;
}
```

JS

```javascript
let size = 5;
let arr = []; // Simple array declaration
for (let i = 0; i < size; i++) {
    arr[i] = i * 2; // Assign values
}

for (let i = 0; i < size; i++) {
    console.log(arr[i]); // Print values
}
```

Q1 – In both codes, are wee using a static array or a dynamic array? Why?

Q2 – Why JavaScript **does not** need explicit **memory allocation** or **deallocation**, as C++ need it?

*PART 2 - JS **CODING CHALLENGES!***

Good job!

Now you should know the basic syntax of JavaScript!

Let’s solve some problem now.