

# Task 4

**ITI**

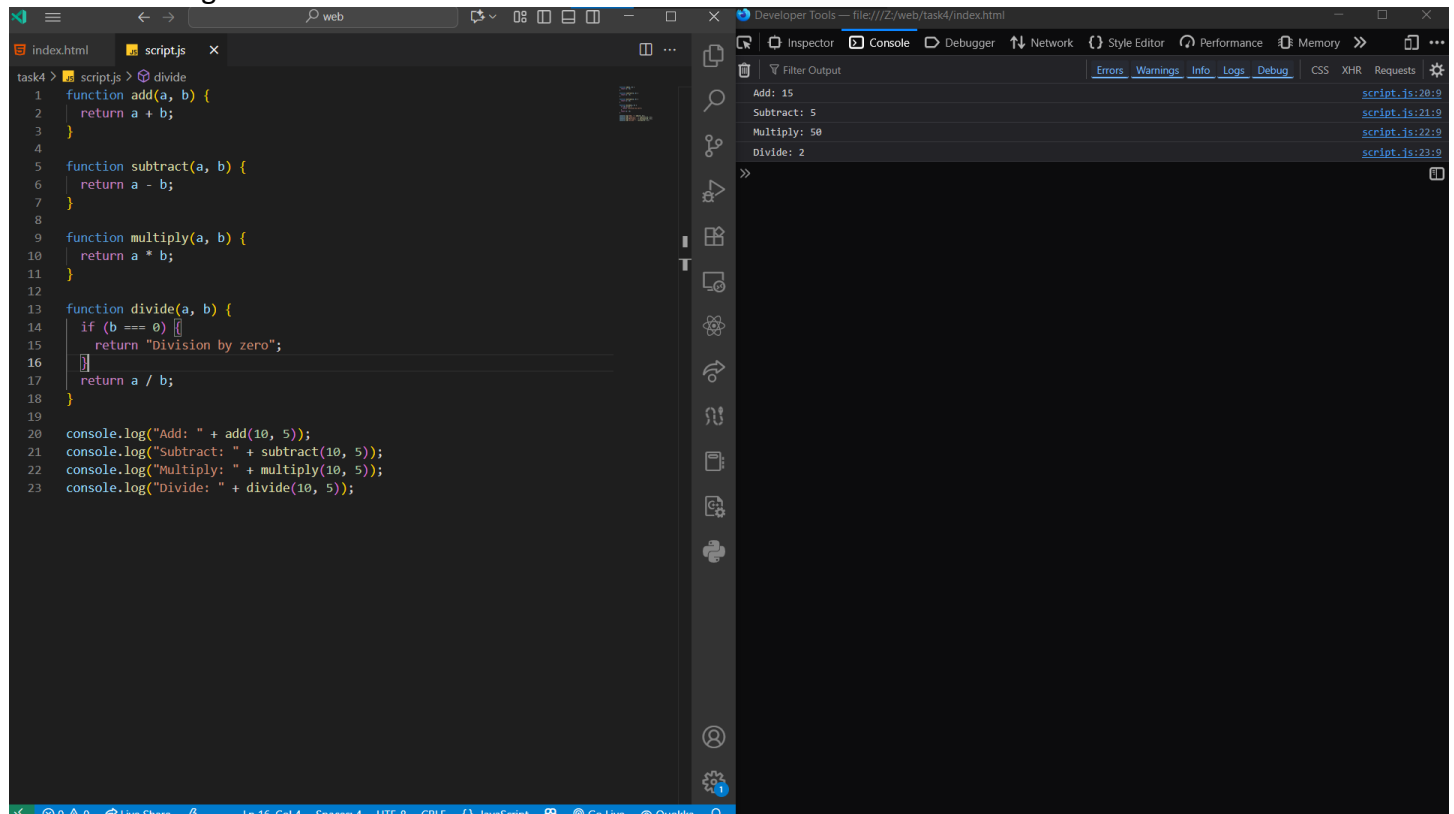
Full Stack

—

Mohamed Salem Mohamed Soliman

—

## Calculator using functions



```
task4 > script.js > divide
1 function add(a, b) {
2   return a + b;
3 }
4
5 function subtract(a, b) {
6   return a - b;
7 }
8
9 function multiply(a, b) {
10  return a * b;
11 }
12
13 function divide(a, b) {
14   if (b === 0) {
15     return "Division by zero";
16   }
17   return a / b;
18 }
19
20 console.log("Add: " + add(10, 5));
21 console.log("Subtract: " + subtract(10, 5));
22 console.log("Multiply: " + multiply(10, 5));
23 console.log("Divide: " + divide(10, 5));
```

Developer Tools — file:///Z:/web/task4/index.html

Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

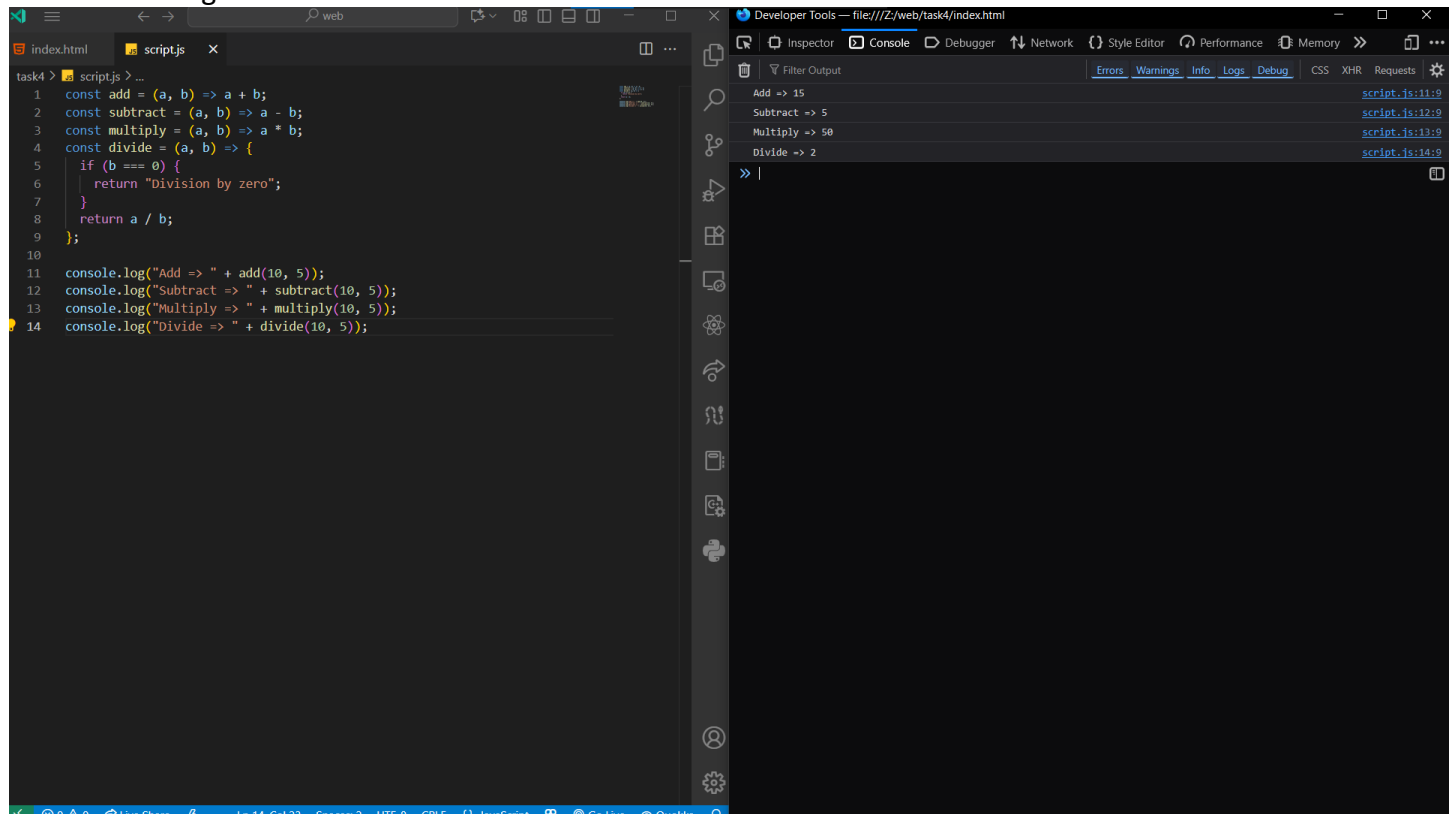
Add: 15 [script.js:20:9](#)

Subtract: 5 [script.js:21:9](#)

Multiply: 50 [script.js:22:9](#)

Divide: 2 [script.js:23:9](#)

## Calculator using arrow function



```
task4 > script.js > ...
1 const add = (a, b) => a + b;
2 const subtract = (a, b) => a - b;
3 const multiply = (a, b) => a * b;
4 const divide = (a, b) => {
5   if (b === 0) {
6     return "Division by zero";
7   }
8   return a / b;
9 };
10
11 console.log("Add => " + add(10, 5));
12 console.log("Subtract => " + subtract(10, 5));
13 console.log("Multiply => " + multiply(10, 5));
14 console.log("Divide => " + divide(10, 5));
```

Developer Tools — file:///Z:/web/task4/index.html

Inspector Console Debugger Network Style Editor Performance Memory

Filter Output

Add => 15 [script.js:11:9](#)

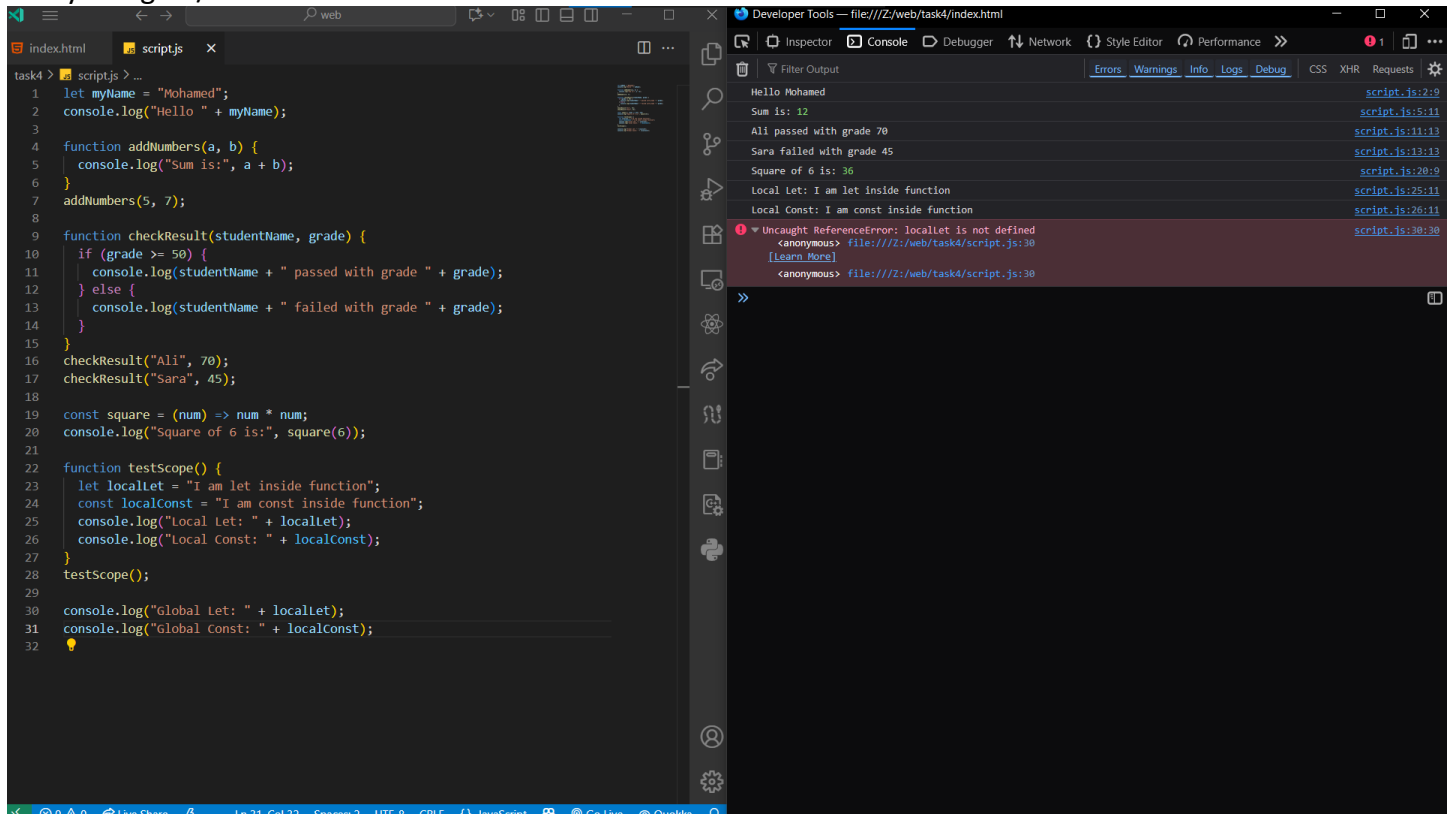
Subtract => 5 [script.js:12:9](#)

Multiply => 50 [script.js:13:9](#)

Divide => 2 [script.js:14:9](#)

## Practice Exercises

1. Define a variable with your name and print 'Hello [name]'.
2. Create a function that takes two numbers and prints their sum.
3. Create a function that takes a student's name and grade, then prints a pass/fail message.
4. Use an arrow function to square a number.
5. Try using let/const inside a function and test access outside.



The screenshot shows a web browser with the developer tools open. The left pane displays the source code of a file named `script.js`. The code implements several JavaScript exercises. The right pane shows the console output, which includes the results of the code execution. An error is visible in the console: `Uncaught ReferenceError: locallet is not defined` at line 30 of `script.js`. The error message also includes a link to [Learn More](#).

```
task4 > script.js > ...
1 let myName = "Mohamed";
2 console.log("Hello " + myName);
3
4 function addNumbers(a, b) {
5   console.log("Sum is:", a + b);
6 }
7 addNumbers(5, 7);
8
9 function checkResult(studentName, grade) {
10  if (grade >= 50) {
11    console.log(studentName + " passed with grade " + grade);
12  } else {
13    console.log(studentName + " failed with grade " + grade);
14  }
15 }
16 checkResult("Ali", 70);
17 checkResult("Sara", 45);
18
19 const square = (num) => num * num;
20 console.log("Square of 6 is:", square(6));
21
22 function testScope() {
23   let locallet = "I am let inside function";
24   const localconst = "I am const inside function";
25   console.log("Local let: " + locallet);
26   console.log("Local Const: " + localconst);
27 }
28 testScope();
29
30 console.log("Global let: " + locallet);
31 console.log("Global Const: " + localconst);
32
```

Console Output:

- Hello Mohamed
- Sum is: 12
- Ali passed with grade 70
- Sara failed with grade 45
- Square of 6 is: 36
- Local let: I am let inside function
- Local Const: I am const inside function
- Uncaught ReferenceError: locallet is not defined

Error Details:

- Uncaught ReferenceError: locallet is not defined
- file:///Z:/web/task4/script.js:30
- Learn More