

script.js

```
1  console.clear();
2
3  // Functions in JavaScript
4
5  // Functions are reusable blocks of code that perform a specific task or set of tasks.
6
7  // Functions are defined using the function keyword.
8
9  // Functions are called using the function name followed by parenthesis.
10
11 // Parameters are data that are passed into a function when it is declared.
12
13 // Arguments are data that are passed into a function when it is called.
14
15 // Return statements are used to return data from a function.
16
17 // Example of a function:
18
19 function addTwoNumbers(num1, num2) {
20
21     // Here num1 and num2 are parameters
22
23     return num1 + num2;
24 }
25
26 console.log(addTwoNumbers(10, 20)); // 30 - Here 10 and 20 are arguments
27
28 console.log(addTwoNumbers(100, 200)); // 300 - Here 100 and 200 are arguments
29
30 // Function Expression
31
32 // A function expression is a function that is assigned to a variable.
33
34 // Example of a function expression:
35
36 const multiplyTwoNumbers = function (num1, num2) {
```

```
37
38     return num1 * num2;
39 }
40
41 console.log(multiplyTwoNumbers(40, 20)); // 200 - Here 40 and 20 are arguments
42
43 console.log(multiplyTwoNumbers(100, 200)); // 20000 - Here 100 and 200 are arguments
44
45 // Anonymous Function
46
47 // An anonymous function is a function that is without a name.
48
49 // Example of an anonymous function:
50
51 const subtractTwoNumbers = function (num1, num2) {
52
53     return num1 - num2;
54 }
55
56 console.log(subtractTwoNumbers(100, 20)); // 80 - Here 100 and 20 are arguments
57
58 console.log(subtractTwoNumbers(100, 200)); // -100 - Here 100 and 200 are arguments
59
60 // Immediately Invoked Function Expression (IIFE)
61
62 // An IIFE is a function that is called immediately after it is defined.
63
64 // Example of an IIFE:
65
66 (function (num1, num2) {
67
68     console.log(num1 + num2);
69
70 })(5, 10);
```