JavaScript Function Parameters

1. What are parameters?

Parameters are variables listed inside the parentheses of a function definition. They act as placeholders that allow you to pass values (called arguments) into a function when it is called.

```
function greet(name) {
  console.log("Hello, " + name + "!");
}

greet("Alice"); // Output: Hello, Alice!
```

2. Why are parameters used?

Parameters make functions reusable and flexible. Instead of hardcoding values, you can pass different data each time.

```
function add(a, b) {
  return a + b;
}

console.log(add(5, 3));  // 8
console.log(add(10, 20)); // 30
```

3. What can be the name of a parameter?

Parameter names follow the same rules as variable names in JavaScript: ✔ Can contain letters, numbers, _ , and $ ✔ Cannot start with a number ✔ Cannot be a reserved keyword

```
// Valid examples
function calculateSum(num1, num2) { ... }
function greetUser(user_name) { ... }
function $getPrice(value) { ... }

// Invalid examples
function my-function(x, y) { ... }  // Hyphen not allowed
function 2values(a, b) { ... }      // Cannot start with number
function let(x) { ... }             // Reserved word
```

4. How many parameters can be passed to a function?

In JavaScript, a function can take any number of parameters. Technically, there is no strict limit. Additionally, JavaScript provides special ways to handle unknown numbers of parameters: the arguments object and rest parameters (...args).

```
function showValues(a, b, c, d) {
  console.log(a, b, c, d);
}

showValues(1, 2, 3, 4); // Output: 1 2 3 4

// Using rest parameter
function sumAll(...numbers) {
  return numbers.reduce((total, num) => total + num, 0);
}

console.log(sumAll(1, 2, 3, 4, 5)); // 15
console.log(sumAll(10, 20));        // 30
```