



Sheryians
Coding School

Live Cohort

— DAY-36 —

JavaScript Concepts



JavaScript Concepts

1. do...while Loop

The `do...while` loop executes the code block at least once before checking the condition.

It is used when you want the loop to run once even if the condition is false.

✚ Example:

```
let i = 1;

do {
  console.log(i);
  i++;
} while (i <= 5);
```

Output:

```
1
2
3
4
5
```

Key Point:

The condition is checked after the loop body executes, so the loop will always run at least once.

2. Recursion in JavaScript

Recursion means a function calling itself.

It is used when a task can be broken into smaller repetitive sub-tasks, such as calculating a factorial or traversing nested data.

✚ Example:

```
function factorial(n) {
  if (n === 1) return 1;
  return n * factorial(n - 1);
}

console.log(factorial(5)); // 120
```

Key Point:

A recursive function must have a base case, otherwise it will run infinitely.

JavaScript Concepts

3. Functions – Arguments and Parameters

Parameters are variables listed in a function's definition. Arguments are the actual values passed to the function when it is called.

✚ Example:

```
function greet(name) { // 'name' is a parameter
  console.log("Hello " + name);
}

greet("Shery"); // "Shery" is an argument
```

Key Point:

Parameters act as placeholders inside a function, and arguments are the real values supplied during the function call.

4. Variable Hoisting

Hoisting is JavaScript's behavior of moving variable declarations to the top of their scope before code execution.

This means a variable can be referenced before it is declared, but its value will be `undefined`.

✚ Example:

```
console.log(a); // undefined
var a = 10;
```

Key Point:

- `var` declarations are hoisted (only the declaration, not the value).
- `let` and `const` are also hoisted but remain in the Temporal Dead Zone (TDZ) until their declaration line is executed.

JavaScript Concepts

5. Function Hoisting

Function declarations are also hoisted. This means you can call a function before it is defined in your code.

🧩 Example:

```
sayHello();

function sayHello() {
  console.log("Hello Sheryians!");
}
```

Key Point:

- Function declarations are hoisted.
- Function expressions and arrow functions are **not hoisted**.

6. Arguments Object

Every normal function in JavaScript automatically gets an ``arguments`` object. It is an array-like object that holds all values passed to the function.

🧩 Example:

```
function show() {
  console.log(arguments);
}

show(10, 20, 30);

Output:
[10, 20, 30]
```