Web Development

# Class No: 04

linkedin.com/in/m-munim1

github.com/M-Munim

+92 330 2021926

# Functions, Arrays, and Objects

## What is a Function?

A function is a block of code designed to perform a **specific task**. You can "call" or "invoke" the function whenever you need that task to be performed.

### Function Syntax

```
function functionName(parameters) {
        // Code to execute
}

functionName(“hello”)        //calling the function
```

# Types of Functions in JavaScript

## Function with No Parameters

A function that does not take any input values (parameters) when called. It simply performs its task and may or may not return a value.

## Example:

```
function greet() {
    console.log("Hello, World!");
}
greet();                    // Output: Hello, World!
```

## Function with Parameters

A function that takes input values (called parameters) when called. The parameters allow the function to work with different data.

**Example:**

```
function greet(name) {
    console.log("Hello, " + name + "!");
}
greet("Ali");                         // Output: Hello, Ali!
greet(“rabia”);
greet(“Abdullah”);
```

## Function with Return Value

A function that sends a result back to the code that called it. The return keyword is used to give back the result.

## Example:

```
function addNumber(a, b) {                //a,b are parameters here
    return a + b;
}
let sum = addNumber(5, 10);
console.log("Sum:", sum);                 // Output: Sum: 15
```

## Arrow Functions in JavaScript

Arrow functions provide a shorter syntax to define functions. They are defined using the => (arrow) symbol.

### Syntax:

```
const functionName = (parameters) => {
        // Code to execute
};
```

### Example (Arrow Function with No Parameters):

```
const greet = () => {
  console.log("Hello, World!");
};
greet();                // Output: Hello, World!
```

## Example (Arrow Function with Parameters):

```
const add = (a, b) => {
    return a + b;
};
console.log(add(5, 10));                    // Output: 15
```

## Simplified Arrow Function (Single Line):

If the function has only one expression, you can omit the braces {} and the return keyword:

```
const multiply = (a, b) => a * b;
console.log(multiply(5, 3));                // Output: 15
```

# Arrays in JavaScript

## What is an Array?

An array is a collection of multiple values stored in a single variable. Each value in an array is called an "element.“

## Array Syntax

```
let arrayName = [value1, value2, value3];
```

## Example (Declaring an Array):

```
let fruits = ["Apple", "Banana", "Mango"];
console.log(fruits);          // Output: ["Apple", "Banana", "Mango"]
```

## Accessing Array Elements

You can access an array element by its index (position). Indexes start from 0.

**Example:**

```
let fruits = ["Apple", "Banana", "Mango"];
console.log(fruits[0]);                    // Output: Apple
```

## Common Array Methods

- **push():** Adds an element to the end of the array.

```
let fruits = ["Apple", "Banana"];
fruits.push("Mango");
console.log(fruits);                       // ["Apple", "Banana", "Mango"]
```

- **pop():** Removes the last element.

```
let fruits = ["Apple", "Banana"];
fruits.pop();
console.log(fruits);               // ["Apple"]
```

- **shift():** Removes the first element.

```
let fruits = ["Apple", "Banana"];
fruits.shift();
console.log(fruits);               // ["Banana"]
```

- **unshift():** Adds an element to the beginning of the array.

```
let fruits = ["Banana"];
fruits.unshift("Apple");
console.log(fruits);                    // ["Apple", "Banana"]
```

# Objects in JavaScript

## What is an Object?

An object is a collection of key-value pairs. Each key represents a property, and the value represents the property's data.

## Object Syntax

```
let objectName = {
    key1: value1,
    key2: value2,
    key3: value3
};
```

**Example:**

```
let student = {
    name: "Ali",
    age: 20,
    isEnrolled: true
};
console.log(student.name);                    // Output: Ali
```

## Accessing and Modifying Object Properties

- You can access object properties using dot notation or bracket notation.
- You can also modify or add new properties.

## Example:

```
let student = { name: "Ali", age: 20 };
console.log(student.name);          // Access property using dot notation
student.age = 21;                   // Modify property
student.grade = "A";                // Add new property
console.log(student);
```

# Activity-Based Task: Student Report Generator

In this task, we will build a Student Report Generator using functions and objects.

## Task Description:

1. Create an object student with properties like name, age, and marks (an array of subject marks).
2. Write a function calculateAverage() to calculate the average marks.
3. Display the student details and average marks.

## **Code for Activity**:

```
let student = {
    name: "Ali",
    age: 20,
    marks: [85, 90, 78, 88, 92]    400/5
};

function calculateAverage(marks) {
    let sum = 0;
    for (let i = 0; i < marks.length; i++) {
      sum += marks[i];
    }
    return sum / marks.length;
}

console.log("Student Name:", student.name);
console.log("Student Age:", student.age);
console.log("Average Marks:",
calculateAverage(student.marks));
```

## Home Task 4

1. What is a function in JavaScript? Create a function to calculate the square of a number.
2. Write a function that takes two numbers as input and returns their product.
3. Declare an array of colors. Add a new color to the end and remove the first color.
4. Create an array of numbers and find the sum of all numbers using a loop.
5. Create an object car with properties make, model, and year. Print each property.
6. Write a function isEven() that checks if a number is even or odd.
7. Create an array of student names. Use a loop to print each name.
8. Write a function that takes an array of numbers and returns the largest number.
9. Create an object person with properties firstName and lastName. Write a function to display the full name.
10. Write a program to reverse the elements of an array without using any built-in methods.