

Module 4: JavaScript (Classes 12–16)

Module Duration: 10 Hours (5 Classes × 2 Hours)

Software: Visual Studio Code, Google Chrome

Programming Language: JavaScript (ES6)

Module Learning Objectives

By the end of this module, students will be able to:

- Understand JavaScript fundamentals and its role in web development.
 - Write JavaScript programs using variables, operators, functions, and loops.
 - Manipulate HTML and CSS using the Document Object Model (DOM).
 - Validate user input with JavaScript.
 - Store data using Local Storage.
 - Fetch data from external APIs.
 - Build interactive and dynamic web applications.
-

Class 12 (2 Hours)

Introduction to JavaScript

Learning Outcomes

After completing this class, students will be able to:

- Understand JavaScript and its uses.
 - Write basic JavaScript code.
 - Declare variables and use data types.
 - Perform calculations using operators.
 - Display output using the browser console and alerts.
-

12.1 What is JavaScript?

JavaScript is a programming language used to make web pages interactive and dynamic.

Uses of JavaScript

- Form Validation
 - Image Sliders
 - Pop-up Messages
 - Interactive Menus
 - Games
 - Dynamic Content
 - API Integration
 - Single Page Applications (SPA)
-

12.2 Adding JavaScript to HTML

Inline JavaScript

```
<button onclick="alert('Welcome!')">Click Me</button>
```

Internal JavaScript

```
<script>  
console.log("Hello World");  
</script>
```

External JavaScript (Recommended)

```
<script src="script.js"></script>
```

12.3 Variables

JavaScript provides three ways to declare variables:

- var
- let
- const

Example:

```
let name = "John";  
const country = "Bangladesh";
```

12.4 Data Types

- String
 - Number
 - Boolean
 - Null
 - Undefined
 - Object
 - Array
-

12.5 Operators

Arithmetic Operators

- -
 -
 -
 -
 -
- /
- %
- **

Comparison Operators

- ==
- ===
- !=
-
- <
- - =
- <=

Logical Operators

- &&
 - ||
 - !
-

12.6 Input & Output

Functions:

- `alert()`
- `prompt()`
- `console.log()`

Example:

```
let name = prompt("Enter your name");  
alert("Welcome " + name);
```

Practical Exercise

Create a simple calculator that performs:

- Addition
 - Subtraction
 - Multiplication
 - Division
-

Assignment

Create a program that calculates the average marks of five subjects.

Class 13 (2 Hours)

Conditions, Loops & Functions

Learning Outcomes

Students will be able to:

- Use decision-making statements.
- Repeat tasks using loops.
- Create reusable functions.
- Work with arrays and strings.

13.1 Conditional Statements

- if
- if...else
- else if
- switch

Example:

```
let marks = 75;

if(marks >= 50){
    console.log("Pass");
}else{
    console.log("Fail");
}
```

13.2 Loops

Types of loops:

- for
- while
- do...while
- for...of

Example:

```
for(let i=1;i<=10;i++){
    console.log(i);
}
```

13.3 Functions

Example:

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

13.4 Arrays

Example:

```
let colors = ["Red", "Green", "Blue"];
```

Useful methods:

- push()
 - pop()
 - shift()
 - unshift()
 - length
-

13.5 String Methods

- toUpperCase()
 - toLowerCase()
 - trim()
 - includes()
 - replace()
-

Practical Exercise

Create a Student Grade Calculator.

Requirements:

- Input marks
 - Calculate grade
 - Display result
-

Assignment

Build a multiplication table generator.

Class 14 (2 Hours)

DOM Manipulation & Events

Learning Outcomes

Students will be able to:

- Access HTML elements.
 - Modify webpage content dynamically.
 - Handle user events.
 - Validate forms.
-

14.1 What is the DOM?

DOM (Document Object Model) represents the HTML page as objects that JavaScript can access and modify.

14.2 Selecting Elements

Methods:

- `getElementById()`
 - `getElementsByClassName()`
 - `getElementsByTagName()`
 - `querySelector()`
 - `querySelectorAll()`
-

14.3 Changing Content

Properties:

- `innerHTML`
 - `textContent`
 - `value`
-

14.4 Changing CSS

Example:

```
document.getElementById("title").style.color = "blue";
```

14.5 Events

Common events:

- click
 - dblclick
 - mouseover
 - mouseout
 - keyup
 - keydown
 - submit
 - change
-

14.6 Event Listeners

Example:

```
button.addEventListener("click", function(){  
    alert("Button Clicked");  
});
```

14.7 Form Validation

Validate:

- Required fields
 - Email format
 - Password length
 - Phone number
-

Practical Exercise

Create a registration form with JavaScript validation.

Assignment

Develop an image gallery where clicking a thumbnail displays a larger image.

Class 15 (2 Hours)

Objects, JSON & Local Storage

Learning Outcomes

Students will be able to:

- Work with objects and JSON.
 - Store data in the browser.
 - Build simple offline applications.
-

15.1 JavaScript Objects

Example:

```
const student = {  
  name:"Rahim",  
  age:20,  
  department:"CSE"  
};
```

15.2 JSON

JSON stands for **JavaScript Object Notation**.

Methods:

- JSON.stringify()
 - JSON.parse()
-

15.3 Local Storage

Methods:

- `setItem()`
- `getItem()`
- `removeItem()`
- `clear()`

Example:

```
localStorage.setItem("username", "John");
```

15.4 Date & Time

Useful methods:

- `Date()`
 - `getDate()`
 - `getMonth()`
 - `getFullYear()`
-

Practical Exercise

Build a To-Do List application that:

- Adds tasks
 - Deletes tasks
 - Saves tasks using Local Storage
-

Assignment

Create a Notes application that stores notes in Local Storage.

Class 16 (2 Hours)

ES6 Features & Fetch API

Learning Outcomes

Students will be able to:

- Use modern JavaScript syntax.
 - Fetch data from APIs.
 - Handle asynchronous operations.
 - Build dynamic web applications.
-

16.1 ES6 Features

let & const

Arrow Functions

Example:

```
const add = (a,b) => a+b;
```

Template Literals

Example:

```
let name = "John";  
console.log(`Welcome ${name}`);
```

Destructuring

```
const person = {  
  name:"Ali",  
  age:25  
};  
  
const {name,age}=person;
```

Spread Operator

```
const numbers = [1,2,3];  
const newNumbers = [...numbers,4,5];
```

16.2 Fetch API

The Fetch API is used to request data from servers.

Example:

```
fetch("https://jsonplaceholder.typicode.com/users")
  .then(response => response.json())
  .then(data => console.log(data));
```

16.3 Introduction to Asynchronous JavaScript

Concepts:

- Synchronous vs Asynchronous
 - Promises
 - async
 - await (Introduction)
-

Practical Exercise

Create a webpage that:

- Fetches user data from a public API
 - Displays user names and email addresses in cards
-

Module Project

Students will develop an **Interactive Student Management Web Application** with the following features:

- Student Registration Form
- Form Validation
- Add Student Information
- Display Student List
- Edit/Delete Records (Frontend Simulation)
- Local Storage Support
- Responsive Design using Bootstrap
- Dynamic DOM Updates

- API Data Display (Optional)
-

Module Assessment

Assessment	Marks
JavaScript Quiz	10
Practical Exercises	20
Assignments	20
Interactive Web Application Project	40
Presentation & Viva	10
Total	100

Learning Outcomes

After completing **Module 4: JavaScript**, students will be able to:

- Write clean and efficient JavaScript code using modern ES6 syntax.
- Use variables, data types, operators, conditions, loops, functions, arrays, and objects to solve programming problems.
- Manipulate HTML and CSS dynamically using the DOM.
- Validate forms and respond to user events.
- Store and retrieve data using Local Storage.
- Consume REST APIs using the Fetch API and display dynamic content.
- Build interactive, responsive web applications that are ready to integrate with backend technologies such as PHP and MySQL in the next module.