**1. INTRODUCTION TO JAVASCRIPT**

**JavaScript (JS)** is a scripting language used to make web pages interactive and dynamic.  
It runs inside the browser and can:

* Manipulate HTML elements
* Respond to user actions (clicks, typing, scrolling)
* Communicate with web servers using APIs

**Adding JavaScript to a Web Page:**

1. **Internal Script:**
2. <script>
3. console.log("Hello JavaScript");
4. </script>
5. **External File:**
6. <script src="script.js"></script>

**2. VARIABLES AND DATA TYPES**

Variables store data values.  
You can declare variables using:

var name = "Udaya"; // old method

let age = 23; // block-scoped

const company = "Dhruv"; // constant value

**JavaScript Data Types:**

1. **String** – "Hello"
2. **Number** – 100, 10.5
3. **Boolean** – true or false
4. **Array** – [1, 2, 3]
5. **Object** – { name: "Udaya", age: 23 }
6. **Null / Undefined**

Example:

let student = {

name: "Udaya",

course: ".NET",

marks: 90

};

console.log(student.name);

**3. OPERATORS AND CONDITIONAL STATEMENTS**

**Arithmetic Operators:** +, -, \*, /, %  
**Comparison:** ==, ===, !=, <, >, <=, >=  
**Logical:** &&, ||, !

**If-Else Example:**

let marks = 85;

if (marks >= 90) {

console.log("Excellent");

} else if (marks >= 75) {

console.log("Good");

} else {

console.log("Needs Improvement");

}

**Switch Example:**

let course = "Java";

switch (course) {

case "Java":

console.log("You selected Java.");

break;

case ".NET":

console.log("You selected .NET.");

break;

default:

console.log("Invalid choice.");

}

**4. LOOPS**

Used to execute code repeatedly.

for (let i = 1; i <= 5; i++) {

console.log("Iteration " + i);

}

let courses = ["Java", ".NET", "Python"];

for (let c of courses) {

console.log(c);

}

**5. FUNCTIONS**

Functions are reusable blocks of code.

**Function Declaration:**

function greet(name) {

return "Hello, " + name + "!";

}

console.log(greet("Udaya"));

**Arrow Function:**

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

console.log(add(5, 3));

**6. ARRAYS AND OBJECTS**

**Array Example:**

let students = ["Udaya", "Rahul", "Arun"];

students.push("Ravi");

console.log(students[0]); // First item

**Object Example:**

let student = {

name: "Udaya",

course: ".NET",

score: 90

};

console.log(student.course);

**7. DOM (DOCUMENT OBJECT MODEL) MANIPULATION**

The DOM is a tree-like structure representing all elements in an HTML page.  
JavaScript can access and change them dynamically.

**Selecting Elements:**

let heading = document.getElementById("title");

heading.innerText = "Welcome to JavaScript!";

**Changing Styles:**

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

**Creating Elements:**

let newPara = document.createElement("p");

newPara.textContent = "This paragraph was created dynamically.";

document.body.appendChild(newPara);

**8. EVENT HANDLING**

Events are actions performed by users — like clicking, typing, or submitting a form.

**HTML Example:**

<button onclick="showMessage()">Click Me</button>

**JavaScript Function:**

function showMessage() {

alert("Button clicked!");

}

**Using Event Listener (Recommended):**

document.getElementById("myBtn").addEventListener("click", function() {

alert("Button was clicked!");

});

**9. FETCH API (CALLING REST ENDPOINTS)**

**Fetch API** is used to communicate with backend servers (e.g., ASP.NET Core APIs).  
It helps retrieve or send data asynchronously.

**GET Request Example:**

fetch("https://jsonplaceholder.typicode.com/users")

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.error("Error:", error));

**POST Request Example:**

fetch("https://jsonplaceholder.typicode.com/posts", {

method: "POST",

headers: { "Content-Type": "application/json" },

body: JSON.stringify({

title: "New Post",

body: "This is a demo post.",

userId: 1

})

})

.then(res => res.json())

.then(data => console.log("Created:", data))

.catch(err => console.log(err));

**10. BEST PRACTICES**

* Use const and let instead of var
* Write modular, reusable functions
* Handle errors with try...catch
* Avoid global variables
* Use === for strict comparisons
* Comment code for clarity

**Mini Practice Task (Day 3)**

**Objective:** Add JavaScript to your Day 2 project to make it interactive.

**Tasks:**

1. When the user submits the form, show a confirmation alert.
2. Display entered student details in the table dynamically.
3. Change heading color using a button click event.
4. Fetch mock student data using a public API.

A screenshot of a computer screen

AI-generated content may be incorrect.**Snapshots:**

Code : Day3\_JavaScript\_Basics.html

A screenshot of a computer program

AI-generated content may be incorrect.

Code : script.js

A screenshot of a computer

AI-generated content may be incorrect.

Output: Day3\_JavaScript\_Basics.html **and** script.js

A screen shot of a computer

AI-generated content may be incorrect.

JSON Data of Sample Users