**MODULE 6-INTRODUCTION TO JAVASCRIPT**

**💡 What is JavaScript?**

**JavaScript** is a **high-level, interpreted programming language** that is most commonly used to create **interactive and dynamic content** on the web. It is one of the core technologies of the **World Wide Web**, alongside **HTML** and **CSS**.

**✅ Key Characteristics:**

* **Client-side scripting** language (runs in the browser)
* Can also run on the **server-side** (with tools like **Node.js**)
* Supports **Object-Oriented**, **Functional**, and **Event-Driven** programming
* Runs in **web browsers** without needing any installation

**🧩 Where is JavaScript Used?**

|  |  |
| --- | --- |
| **Area** | **Use Case Examples** |
| 🌐 **Web Development** | Interactive forms, animations, sliders, dynamic content updates (DOM manipulation) |
| 📱 **Mobile App Development** | Cross-platform mobile apps using frameworks like React Native, Ionic |
| 🖥 **Desktop Applications** | Building apps with frameworks like Electron (e.g., VS Code, Slack) |
| 💾 **Server-Side Programming** | Back-end logic using Node.js, APIs, databases, and authentication |
| 🤖 **Game Development** | Browser-based games using libraries like Phaser.js |
| 📊 **Data Visualization** | Interactive charts and dashboards using libraries like D3.js, Chart.js |
| 📡 **Web APIs** | Fetching data from external servers, handling HTTP requests |

**✨ Common Uses in a Website**

1. **Form validation** before submitting to server
2. **Updating content dynamically** without refreshing the page (AJAX)
3. **Creating animations** and interactive effects
4. **Responsive navigation menus**
5. **Detecting user events** like clicks, input, scrolling, etc.

**🧠 JAVASCRIPT Variables**

Variables in JavaScript are **containers** for storing data values.  
They can be declared using:

* var – **old** way (avoid using in modern code)
* let – **modern** and allows reassignment
* const – for **constants** (cannot be reassigned)

**✅ Example (with HTML)**

<!DOCTYPE html>

<html>

<head><title>JavaScript Variables</title></head>

<body>

<h2>Check your browser console</h2>

<script>

var oldWay = "This is var";

let name = "Alice";

const country = "Kenya";

console.log(oldWay);

console.log(name);

console.log(country);

</script>

</body>

</html>

**🔢 JavaScript Data Types**

JavaScript supports **two types** of data types:

**1. Primitive Data Types (single values)**

| **Data Type** | **Example** | **Description** |
| --- | --- | --- |
| String | "Hello" | Text data |
| Number | 42, 3.14 | Integer or floating point |
| Boolean | true, false | Logical values |
| Null | null | No value (intentional empty value) |
| Undefined | undefined | Variable declared but not assigned |
| Symbol | Symbol("id") | Unique identifiers (rarely used) |
| BigInt | 123456789n | Large integers beyond Number limit |

**✅ Example: Primitive Data Types**

<!DOCTYPE html>

<html>

<head><title>Data Types</title></head>

<body>

<script>

let username = "John"; // String

let age = 25; // Number

let isStudent = true; // Boolean

let emptyValue = null; // Null

let notAssigned; // Undefined

console.log(typeof username); // "string"

console.log(typeof age); // "number"

console.log(typeof isStudent); // "boolean"

console.log(typeof emptyValue); // "object" (this is a known quirk in JS)

console.log(typeof notAssigned); // "undefined"

</script>

</body>

</html>

**2. Reference (Non-Primitive) Data Types**

| **Data Type** | **Description** |
| --- | --- |
| Object | Collection of key-value pairs |
| Array | Ordered list of items (object type) |
| Function | Block of code that can be reused |

**✅ Example: Objects and Arrays**

<!DOCTYPE html>

<html>

<head><title>Objects and Arrays</title></head>

<body>

<script>

// Object

const person = {

name: "Grace",

age: 30,

country: "Kenya"

};

// Array

let fruits = ["Mango", "Banana", "Orange"];

console.log(person.name); // Grace

console.log(fruits[1]); // Banana

</script>

</body>

</html>

**2. JavaScript Output**

**✅ Output Methods:**

* console.log() – for developer debugging
* alert() – for popup messages
* document.write() – writes directly to the HTML document (rarely used now)
* innerHTML – updates HTML elements

**🧪 Examples**

## ✅ 1. console.log() – Developer Debugging

Used mainly for **debugging**. It prints messages to the **browser console** (Press F12 > Console tab in browser).

### 🧪 Example:

<!DOCTYPE html>

<html>

<head>

<title>Console Log Example</title>

</head>

<body>

<h2>Open the Console (F12) to see the output</h2>

<script>

let name = "Alex";

let age = 28;

console.log("Name: " + name);

console.log("Age: " + age);

console.log("Debugging done!");

</script>

</body>

</html>

## ✅ 2. alert() – Popup Messages

Used to display **alert boxes** with information for users.

### 🧪 Example:

<!DOCTYPE html>

<html>

<head>

<title>Alert Example</title>

</head>

<body>

<h2>Alert Box Example</h2>

<script>

let message = "Welcome to JavaScript!";

alert(message);

</script>

</body>

</html>

## ✅ 3. document.write() – Writes Directly to the Document

Writes HTML or text directly into the page. Not recommended for modern web development, but useful for simple examples.

### 🧪 Example:

<!DOCTYPE html>

<html>

<head>

<title>Document Write Example</title>

</head>

<body>

<script>

let today = new Date();

document.write("<h3>Today is: " + today + "</h3>");

</script>

</body>

</html>

## ✅ 4. innerHTML – Updates HTML Elements

Used to **dynamically change** the content of an HTML element.

### 🧪 Example:

<!DOCTYPE html>

<html>

<head>

<title>innerHTML Example</title>

</head>

<body>

<h2 id="greeting">Hello!</h2>

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

<script>

function changeMessage() {

document.getElementById("greeting").innerHTML = "You clicked the button!";

}

</script>

</body>

</html>

## 📝 Summary Table

|  |  |  |
| --- | --- | --- |
| **Method** | **Usage** | **Best For** |
| console.log() | Debugging in browser console | Developers |
| alert() | Popup message | User notifications |
| document.write() | Direct page writing (old style) | Simple demos, not recommended |
| innerHTML | Update specific HTML content | Dynamic webpages |

**JAVASCRIPT Variables**

**✅ Declaring Variables:**

* var – old way (function scoped)
* let – block scoped
* const – block scoped, cannot be reassigned

**🧪 Example:**

<script>

var name = "Alice";

let age = 25;

const country = "Kenya";

document.write(`Name: ${name}, Age: ${age}, Country: ${country}`);

</script>

**JAVASCRIPT Data Types**

**✅ JavaScript has:**

* Strings
* Numbers
* Booleans
* Arrays
* Objects
* Null
* Undefined

**🧪 Example:**

<script>

let text = "Hello";

let number = 42;

let isHappy = true;

let user = null;

let score;

console.log(typeof text); // string

console.log(typeof number); // number

console.log(typeof isHappy); // boolean

console.log(typeof user); // object

console.log(typeof score); // undefined

</script>

**JAVASCRIPT Strings**

**✅ String operations:**

* Concatenation
* Length
* Template literals
* String methods

**🧪 Example:**

<script>

let firstName = "John";

let lastName = "Doe";

let fullName = `${firstName} ${lastName}`;

console.log(fullName); // John Doe

console.log(fullName.length); // Length of string

console.log(fullName.toUpperCase()); // JOHN DOE

</script>

**JAVASCRIPT Constants**

**✅ const variables cannot be reassigned.**

<script>

const pi = 3.14159;

// pi = 3.14; ❌ This would cause an error

console.log(pi);

</script>

**JAVACSRIPT Arrays**

**✅ Arrays hold multiple values in a single variable.**

<script>

let fruits = ["Apple", "Banana", "Mango"];

console.log(fruits[1]); // Banana

fruits.push("Pineapple"); // Add item

fruits.pop(); // Remove last

console.log(fruits.length); // 3

</script>

**JAVACSRIPT Functions**

**✅ Functions are blocks of reusable code.**

<script>

function greet(name) {

return "Hello, " + name;

}

document.write(greet("Jane")); // Hello, Jane

</script>

**console.log() and document.write()**

**✅ Output to:**

* Console (debugging)
* HTML page

html

CopyEdit

<script>

let message = "JavaScript is fun!";

console.log(message);

document.write("<p>" + message + "</p>");

</script>

**🟨 JAVACSRIPT Objects**

**✅ JavaScript objects store key-value pairs.**

<script>

let person = {

name: "Mike",

age: 30,

city: "Nairobi"

};

console.log(person.name); // Mike

person.age = 31;

console.log(person["city"]); // Nairobi

</script>

**JAVACSRIPT Classes**

**✅ ES6 introduced class syntax for creating objects.**

<script>

class Car {

constructor(brand, year) {

this.brand = brand;

this.year = year;

}

start() {

return `${this.brand} is starting...`;

}

}

let myCar = new Car("Toyota", 2022);

console.log(myCar.start()); // Toyota is starting...

</script>

**✅ Summary Table**

|  |  |
| --- | --- |
| **Feature** | **Example Code** |
| Variable | let age = 20; |
| Data Type | typeof "hello" |
| Array | let items = [1,2,3]; |
| String | "Hello".toUpperCase() |
| Constant | const pi = 3.14; |
| Function | function add(a,b){ return a+b; } |
| Object | {name: "Ana", age: 25} |
| Class | class Animal { constructor(name) {} } |
| Output | console.log(), document.write() |