Introduction to JavaScript

a high-level, interpreted programming language

R.A: Assad Ullah Khan

in assadullahkhan

AssadUllahKhan

What is JavaScript?

- JavaScript is a high-level, interpreted programming language.
- It allows developers to make web pages interactive and dynamic.
- JavaScript runs directly inside web browsers like Chrome, Firefox, Edge, and others.
- It can respond to user actions such as button clicks, typing, and scrolling.
- JavaScript was originally created to make web pages alive. Today, it’s also used in servers, mobile apps, and even desktop software.

```
<script>
  alert("Welcome to JavaScript!");
</script>
```

Why Learn JavaScript?

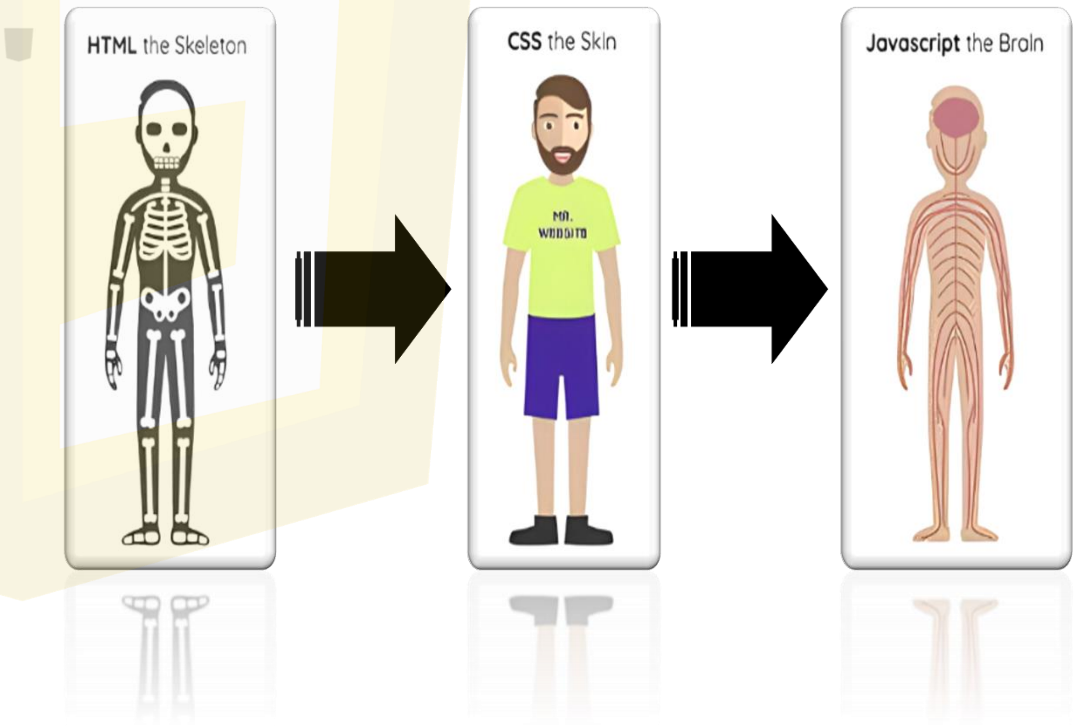
- 💡 JavaScript is the most popular programming language for web development.
- 🌐 It runs on both frontend (browser) and backend (server) using tools like Node.js.
- ⚙️ It helps create interactive, dynamic, and user-friendly websites.
- 🧠 Learning JavaScript builds the foundation for modern frameworks like React, Angular, and Vue.
- Almost every website you visit uses JavaScript — it's the language of the web!

```
<h1 id="title">Hello!</h1>

<script>
 document.getElementById("title").innerHTML = "Hello, JavaScript!";
</script>
```

JavaScript vs HTML & CSS

- 🧱 HTML (HyperText Markup Language):Defines the structure of a webpage — headings, paragraphs, images, buttons, etc.
- 🎨 CSS (Cascading Style Sheets):Adds style and design — colors, fonts, layouts, and animations.
- ⚙️ JavaScript (JS):Adds logic, interactivity, and dynamic behavior — reacts to user actions like clicks, typing, etc.
- HTML builds, CSS beautifies, and JavaScript brings it to life!

```
<button onclick="changeText()">Click Me</button>
<p id="demo">Old Text</p>

<script>
 function changeText() {
  document.getElementById("demo").innerHTML = "Text Changed!";
 }
</script>
```



How to Add JavaScript in HTML

Inline Script:

- Write JavaScript directly inside an HTML element’s attribute using the onclick or similar event.

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

Internal Script:

- Add JavaScript inside the HTML file within the <script> tag.

```
<script>
  alert("This is an internal script!");
</script>
```

External Script:

- Store JavaScript code in a separate .js file and link it using the src attribute.

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

- The external script method is most commonly used in real projects because it keeps your HTML clean and makes your code reusable.

JavaScript Output Methods

JavaScript provides several ways to display or output information:

1. ⚡ **alert()** → Displays a popup message box on the screen.
2. 🧩 **console.log()** → Prints messages in the browser console (useful for debugging).
3. ✍️ **document.write()** → Writes output directly to the HTML page.
4. 🖋️ **innerHTML** → Changes the content of an existing HTML element.

```
<h2 id="message"></h2>

<script>
  alert("Welcome to JavaScript!");
  console.log("This message appears in the console.");
  document.write("This text is written directly to the webpage.");
  document.getElementById("message").innerHTML = "Hello from innerHTML!";
</script>
```

Variables in JavaScript

- 💾 Variables are used to store data values that can be reused or changed later in the program.
- 🧠 Think of a variable as a container that holds information like text, numbers, or objects.
- You can declare variables using three keywords:

1. **Var:** old way (function-scoped, avoid in modern code).
2. **Let:** modern way, allows updating values (block-scoped).
3. **Const:** used for constants that cannot be changed after declaration (block-scoped).

```
let name = "Assad";        // can be changed later
const age = 22;            // cannot be changed
var city = "Peshawar";     // old way
```

JavaScript Data Types

JavaScript variables can hold different types of data.
Here are the most common ones:

String → "Hello" Textual data enclosed in quotes
Number → 25 Numeric values (integer or float)
Boolean → true or false Represents logical values
Object → {key: value} Collection of key-value pairs
Array → [1, 2, 3] – Ordered list of values

```
let name = "Asad";            // String
let age = 22;                 // Number
let isTeacher = true;         // Boolean
let subjects = ["HTML", "CSS", "JS"];         // Array
let person = { name: "Asad", age: 22 };       // Object

console.log(name, age, isTeacher, subjects, person);
```

Operators in JavaScript

Operators are symbols used to perform operations on values and variables.They help in performing calculations, comparisons, and logic in JavaScript.

```
let x = 5, y = 10;

console.log(x + y);          // 15  → Arithmetic
console.log(x > y);          // false → Comparison
console.log(x < y && y > 0);          // true → Logical
```

➕ Arithmetic Operators

Used for basic math operations.

Examples: +, -, *, /, %⚖️

Comparison Operators

Compare two values and return true or false.

Examples: ==, ===, !=, >, <

🔍Logical Operators

Combine or invert Boolean values.

Examples: && (AND), || (OR), ! (NOT)

== checks value equality, while === checks both value and type.

Comments in JavaScript

Comments are notes in your code that the browser ignores.They help you and others understand what the code does.

There are two types of comments in JavaScript:

1) Single-line comment: starts with //

➤ Used to explain one line or a small section.

2) Multi-line comment: enclosed in /* ... */

➤ Used for long explanations or temporarily disabling multiple lines of code.

```
// This is a single-line comment
let name = "Assad"; // Declaring a variable

/*
 This is a multi-line comment
 It can span across multiple lines
*/
console.log("Hello, " + name);
```

Comments make your code readable and easier to maintain.

Numbers in JavaScript

JavaScript uses a single number type for all numeric values. That means integers and floating-point (decimal) numbers are treated the same way.

- No separate type for integers and floats.
- You can perform arithmetic operations like +, -, *, /, %.
- Use built-in objects like Math for advanced operations (e.g., Math.sqrt(), Math.pow()).

```
let a = 10;
let b = 2.5;

let sum = a + b;
console.log(sum);          // Output: 12.5

let result = a * b;
console.log(result);       // Output: 25
```

Booleans in JavaScript

Booleans represent logical values — either true or false.They are mostly used in conditions and decision-making (e.g., if statements).

- Only two possible values → true or false.
- Commonly used to control program flow.
- Results of comparison operations are also Boolean values.

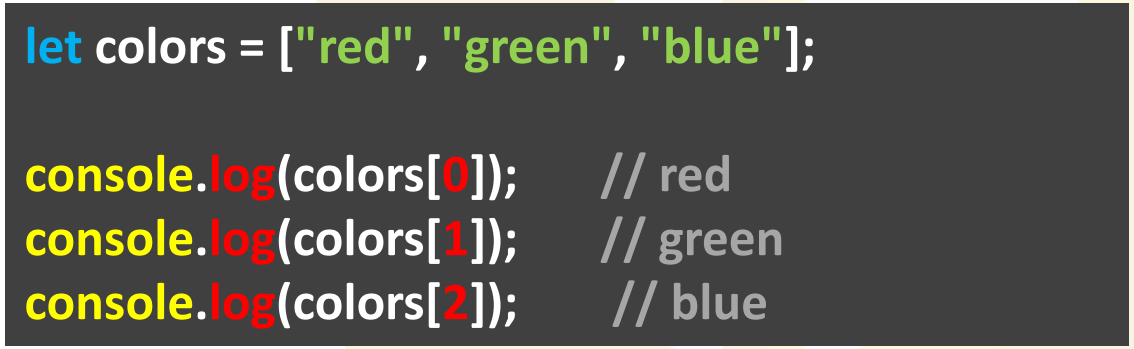
```
let isStudent = true;

if (isStudent) {
  console.log("Welcome Student!");
}

let marks = 80;
console.log(marks > 50);   // true
```

Boolean values (true or false) are used to make decisions in JavaScript programs.

Arrays in JavaScript

An array is used to store multiple values in a single variable.Arrays can hold different data types (numbers, strings, objects, etc.) and are indexed starting from 0.

- Arrays are written with square brackets [ ].
- Each item in an array is called an element.
- The index starts at 0 (first element = index 0).
- Access elements using array[index].



```
let colors = ["red", "green", "blue"];

console.log(colors[0]);     // red
console.log(colors[1]);     // green
console.log(colors[2]);      // blue
```

Arrays store multiple values, and elements are accessed using index numbers starting from 0.

Objects in JavaScript

Objects are used to store data in key–value pairs.Each key (also called a property) has a value, which can be any data type — even another object or array.

- Defined using curly braces { }.
- Each property has a name (key) and a value separated by a colon : .
- Access values using dot notation (object.key) or bracket notation (object["key"]).

```
let student = {
  name: "Asad",
  age: 22,
  course: "Web Development"
};

console.log(student.name);          // Output: Asad
```

Objects store information using key–value pairs and allow easy access to each value.

Any Question

Thank You