Web Development

Introduction to JavaScript

Presented by Muhammad Munim

What is JavaScript?

- JavaScript (JS) is a high-level, interpreted programming language used to make web pages interactive.
- Runs directly in web browsers, allowing dynamic and responsive behavior.
- Supports both client-side (browser) and server-side (Node.js) development.

Common Uses of JavaScript:

- **Form validation** (checking user input before submission).
- **Animations and transitions** (smooth effects for UI elements).
- **Interactive elements** (buttons, modals, dropdown menus, carousels).
- **DOM manipulation** (dynamically updating content without reloading the page).
- **Event handling** (responding to user actions like clicks and keypresses).

A Brief History of JavaScript

- **Created by:** Brendan Eich in 1995 while working at Netscape.
- **Original Name:** Mocha → Later renamed to LiveScript → Finally, JavaScript.
- **Standardization:** ECMAScript (ES) was introduced to standardize JavaScript.

Major Versions:

- **ES5 (2009):** Introduced JSON, strict mode.
- **ES6 (2015):** Introduced let, const, arrow functions, template literals, etc.
- **Latest Versions (ES2023):** Include new features like async/await, optional chaining, and more.

Importance of JavaScript in Web Development

JavaScript is a core technology of modern web development along with HTML and CSS.

- **Enhances User Experience** – Enables dynamic and interactive elements.
- **Client-Side Processing** – Reduces server load by executing code in the browser.
- **Asynchronous Operations** – Allows smooth loading of data without refreshing the page (AJAX, Fetch API).
- **Cross-Platform Compatibility** – Works on all major browsers and devices.
- **Powerful Ecosystem** – Numerous libraries and frameworks for fast development.

Famous JavaScript Frameworks and Libraries

JavaScript has a wide range of libraries and frameworks for front-end and back-end development.

Front-End Frameworks/Libraries:

- **React.js** – Developed by Facebook, used for building UI components.
- **Vue.js** – Lightweight and easy-to-learn framework for interactive UI.
- **Angular.js** – Developed by Google, used for large-scale applications.
- **jQuery** – Simplifies DOM manipulation and event handling.

Back-End Frameworks:

- **Node.js** – Enables JavaScript to run on the server.
- **Express.js** – A lightweight framework for building web applications with Node.js.

JavaScript and the MERN Stack

MERN is a popular JavaScript-based full-stack development framework.

- **M** – MongoDB (NoSQL database)
- **E** – Express.js (Back-end framework)
- **R** – React.js (Front-end framework)
- **N** – Node.js (Server-side runtime)

Why MERN?

- Uses JavaScript across the stack (front-end & back-end).
- Faster development with reusable components.
- Scalable and efficient for modern web applications.

Setting Up JavaScript

Using the Browser Console

- Open Developer Tools (F12 or Ctrl + Shift + I in Chrome).
- Navigate to the Console tab.
- Type and execute JavaScript code directly.

Example:

```
console.log("Hello, World!");
```

Adding JavaScript to HTML

JavaScript can be added inside an HTML file in two ways:

1. Internal JavaScript

Writing JavaScript code directly inside the HTML file within a <script> tag.

Example:

```
<script>
    console.log("Hello, JavaScript!");
</script>
```

External JavaScript File

Writing JavaScript in a separate file and linking it using the <script> tag.

HTML File (index.html):

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

JavaScript File (script.js):

```
console.log("Hello, JavaScript!");
```

Text Editor Setup

- Install Visual Studio Code (VS Code) or any code editor.
- Create an HTML file (index.html) and a JavaScript file (script.js).
- Link the JavaScript file inside the HTML file before the closing </body> tag for better performance.

Writing Your First JavaScript Code

1. Console Output

- The console.log() method is used for debugging and printing messages to the console.

Example:

```
console.log("Hello, JavaScript!");
```

2. Alert Box

- The alert() function displays a popup message to the user.

Example:

```
alert("Welcome to JavaScript!");
```

3. Writing into the HTML Page

- The document.write() method writes content directly into the HTML page.

Example:

```
document.write("Welcome to JavaScript!");
```

Home Task 1

1. What is JavaScript?
2. How do you include JavaScript in an HTML file?
3. What is console.log() used for?
4. Write a JavaScript script to print "Hello, JavaScript!" in the console.
5. What is an alert box in JavaScript?
6. Write a script to display an alert box with the message "Welcome to JavaScript!".
7. Name some popular JavaScript frameworks and their uses.
8. What is the MERN stack?
9. Compare JavaScript and PHP.

Web Development

Class No: 02

linkedin.com/in/m-munim1

github.com/M-Munim

+92 330 2021926

Variables, Data Types & Operators

What is a Variable?

- A variable is a container that stores data values. It allows us to store, update, and reuse values in a program.

Declaring Variables

- JavaScript provides **three ways** to declare variables:

1. **var** – Old method (Not recommended due to scope issues).
2. **let** – Preferred for variables that can change.
3. **const** – Preferred for variables that do not change.

Example:

```
var name = "John";              // Old way (avoid using)
let age = 25;                   // Preferred for changeable values
const country = "Pakistan";     // Preferred for constant values
```

Rules for Naming Variables

- Can contain letters, digits, underscores (_), and dollar signs ($).
- Cannot start with a number.
- Cannot use JavaScript reserved keywords (like var, if, else).
  (https://www.w3schools.com/js/js_reserved.asp)
- Variables are case-sensitive (name and Name are different).

Example:

```
let user1 = "Alice";        // ☑ Valid
let _score = 100;           // ☑ Valid
let $price = 50;            // ☑ Valid
let 1user = "John";         // ❌ Invalid (Cannot start with a number)
let let = 10;               // ❌ Invalid (Reserved keyword)
```

“use strict”;

Treat all JS code as newer version. With strict mode, you can not, for example, use undeclared variables

```
"use strict";
// Example of an error with "use strict"
x = 10;      // Error: x is not defined
console.log(x);
```

Data Types in JavaScript

A data type defines the kind of values a variable can hold. **JS has two main types:**

A. Primitive Data Types (Stores only a single value)

1. String (Text)

```
let greeting = "Hello, World!";
console.log(greeting);          // Output: Hello, World!
```

2. Number (Whole numbers & decimals)

```
let age = 30;
let price = 99.99;
console.log(age, price);
```

3. Boolean (True or False)

```
let isRaining = false;
let isSunny = true;
console.log(isRaining);
```

4. Null (Intentional absence of value, stand alone value)

```
let car = null;
console.log(car);                    // Output: null
```

5. Undefined (Declared but no value assigned)

```
let temperature;
console.log(temperature);     // Output: undefined
```

B. Non-Primitive Data Types (Stores multiple values)

1. Objects (Key-value pairs for data)

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

2. Arrays (Ordered lists for storing data)

```
let colors = ["red", "green", "blue"];
console.log(colors);                    // Output: ['red', 'green', 'blue']
```

Type Conversion (Data Type Conversion) in JavaScript

- It is the process of converting a value from one data type to another.

Implicit Conversion (Automatic by JavaScript)

```
console.log("5" + 5);        // "55" (String)
console.log("10" - 2);       // 8 (Number)
```

Explicit Conversion (Manual by Programmer)

- **String to Number Conversion**

```
let str = "123";
let num = Number(str);
console.log(num);            // 123 (Number)
```

- **Number to String**

```
let num = 456;
let str = num.toString();
console.log(str);                    // "456" (String)
```

- **Boolean to Number**

```
console.log(Number(true));   // 1
console.log(Number(false));  // 0
```

Operators in JavaScript

Operators are symbols that perform operations on variables and values.

A. Arithmetic Operators

```
let a = 10, b = 5;

console.log(a + b);        // 15
console.log(a - b);        // 5
console.log(a * b);        // 50
console.log(a / b);        // 2
console.log(a % b);        // 0
```

B. Comparison Operators

```
let x = 10, y = 5;
console.log(x == y);          // false
console.log(x === y);         // false
console.log(x !== y);         // true
console.log(x > y);           // true
console.log(x < y);           // false
```

C. Logical Operators

```
let a = true, b = false;
console.log(a && b);          // false
console.log(a || b);          // true
console.log(!a);              // false
```

D. Assignment Operators

```
let num = 10;
num += 5;              // Same as num = num + 5;
console.log(num); // 15
```

Home Task 2

1. What are variables in JavaScript?
2. What are the rules for naming variables?
3. What are the different data types in JavaScript?
4. What is the difference between == and ===?
5. Convert "10" to a number using JavaScript.
6. What will be the output of 5 + "5"?
7. What is the result of true && false?
8. What is the difference between null and undefined?
9. What is the difference between alert, prompt and confirm?
10. Write a JavaScript script(JS code) to swap two numbers.

linkedin.com/in/m-munim1 github.com/M-Munim +92 330 2021926

Weekly Assignment 1:

Task 1: Variable Declaration & Usage

- **Declare three variables:** name, age, and isStudent.
- Assign values and print them in the console

Task 2: Type Conversion

- Convert "100" (string) to a number and "true" to a boolean.
- Print the results in the console.

Task 3: Arithmetic Operations

- Take two numbers as input from the user using prompt().
- Perform addition, subtraction, multiplication, and division.
- Show the results using alert().

Task 4: Null vs Undefined

- Declare a variable with null and another with undefined.
- Print both and explain their differences in a comment.

Task 5: Logical Operators

- Take two boolean values (true or false) from the user.
- Perform AND, OR, and NOT operations and show the results.