JavaScript Basics

Web Development Essentials - Session 9

Session Overview

Learning Goals for Today:

- Understand what JavaScript is and its role in web development
- Learn about variables, data types, and operators
- Write your first basic JavaScript code

What is JavaScript?

Definition: JavaScript (JS) is a versatile, dynamic programming language that allows you to create interactive and dynamic content for websites.

Role in Web Development:

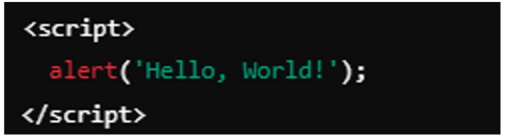
- **HTML**: Defines the structure and content of the webpage.
- **CSS**: Styles the content.
- **JavaScript**: Adds interactivity and behavior, like form validation, animations, and dynamic updates without refreshing the page.

JavaScript in Action

- **Where is JavaScript Used?**
    - Web browsers (client-side): Interactivity (e.g., dropdowns, carousels, form validation)
    - Server-side (Node.js): Back-end development
- **Example Uses**:
    - Real-time form validation
    - Dynamic content loading (without refreshing the page)
    - Animations and user interaction

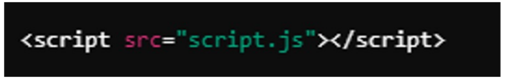
Embedding JavaScript in a Webpage

Inline JavaScript: Adding JavaScript directly in the HTML.



```
<script>
  alert('Hello, World!');
</script>
```

External JavaScript: Linking an external JavaScript file



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

It's good practice to keep JavaScript in external files to organize code better.

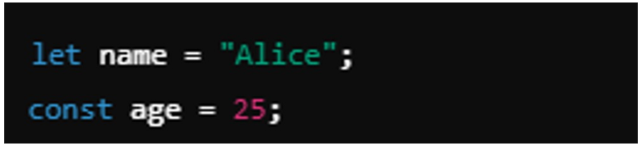
JavaScript Basics: Variables

What are Variables?

- Containers for storing data values.
- In JavaScript, variables are declared using `let`, `const`, or `var`.

Variable Declaration:

- `let`: Used for variables that can change.
- `const`: Used for variables that are constant and cannot change.
- `var`: Older way of declaring variables (use `let` or `const` in modern JavaScript).

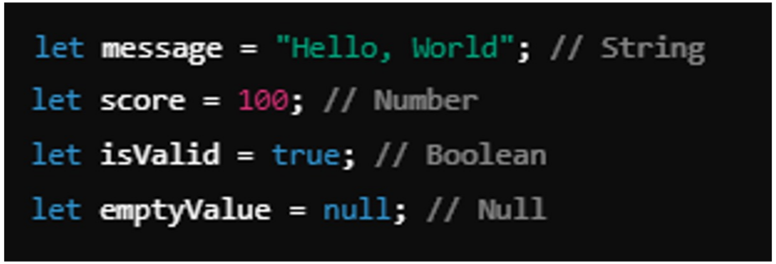


```
let name = "Alice";
const age = 25;
```

Data Types in JavaScript

Primitive Data Types:

1. **String**: Text, written inside quotes (`"Hello"`, `'World'`).
2. **Number**: Numerical values (`25`, `100.5`).
3. **Boolean**: Logical values (`true` or `false`).
4. **Undefined**: A variable declared but not assigned a value.
5. **Null**: Represents the intentional absence of any value.

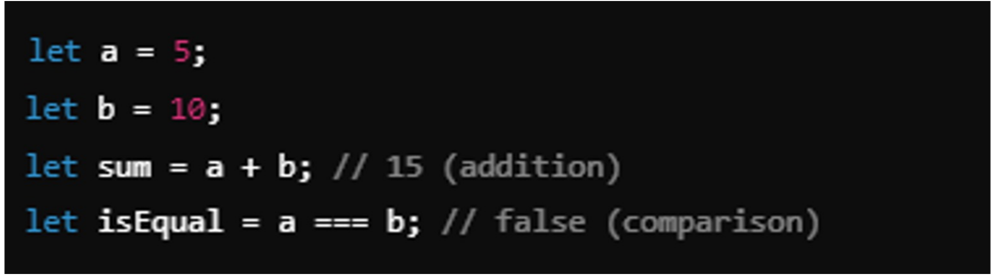


```
let message = "Hello, World"; // String
let score = 100; // Number
let isValid = true; // Boolean
let emptyValue = null; // Null
```

JavaScript Operators

Types of Operators:

1. **Arithmetic Operators**: Perform mathematical operations.
   - `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division), `%` (modulus)
2. **Assignment Operators**: Assign values to variables.
   - `=` (assign), `+=`, `-=`, etc.
3. **Comparison Operators**: Compare values.
   - `==`, `===`, `!=`, `>`, `<`, etc.
4. **Logical Operators**: Combine conditions.
   - `&&` (AND), `||` (OR), `!` (NOT)

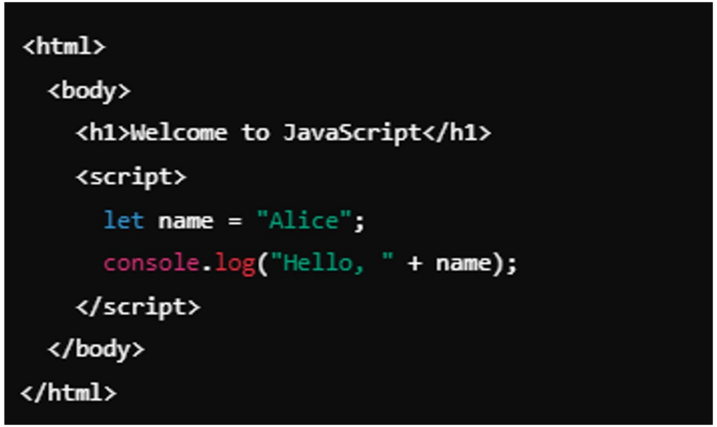


```javascript
let a = 5;
let b = 10;
let sum = a + b; // 15 (addition)
let isEqual = a === b; // false (comparison)
```

Writing Your First JavaScript Code

Step-by-Step Example:

1. Create a basic HTML file.
2. Add a `<script>` tag to write JavaScript.
3. Use `console.log()` to display output in the browser's developer console.



```
<html>
  <body>
    <h1>Welcome to JavaScript</h1>
    <script>
      let name = "Alice";
      console.log("Hello, " + name);
    </script>
  </body>
</html>
```

Result: The message "Hello, Alice" is displayed in the console.

Hands-On Activity

Goal: Create a simple webpage that uses JavaScript.

- Declare a few variables (`let` or `const`).
- Use basic operators to perform a calculation.
- Output the result using `console.log()`.

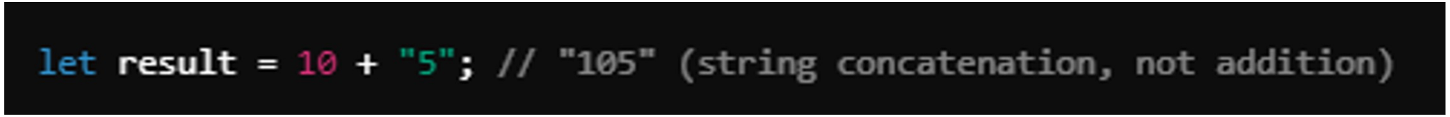
Instructions:

- Open your HTML file and add a `<script>` tag.
- Declare a variable and assign a value.
- Write a simple operation and print the result in the console.

Common JavaScript Mistakes

Forget to Declare Variables: Make sure to use `let`, `const`, or `var` to declare variables.

Mismatched Data Types: Be mindful of mixing data types (e.g., trying to add a string to a number).



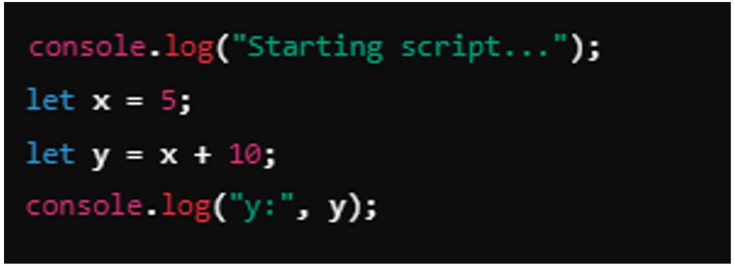
```
let result = 10 + "5"; // "105" (string concatenation, not addition)
```

Case Sensitivity: JavaScript is case-sensitive (`let Name` is different from `let name`).

Debugging JavaScript Code

Using the Console:

- The browser's developer tools provide a console for viewing errors and messages.
- Use `console.log()` to track the flow of your code and debug issues.



```
console.log("Starting script...");
let x = 5;
let y = x + 10;
console.log("y:", y);
```

Summary

- **What We Learned Today**:
  - Introduction to JavaScript and its importance in web development.
  - Declaring variables using `let`, `const`, and `var`.
  - JavaScript data types, operators, and basic syntax.
  - How to write and debug simple JavaScript code.

Questions?

Q&A Session

- Any questions before we wrap up?

Thank You & See You in the Next Class!