

# **Introduction to JavaScript**

# What is JavaScript?

- ✓ A high-level, dynamic programming language primarily used for enhancing web pages and creating interactive content.
- ✓ Interpreted, object-oriented, first-class functions.
- ✓ Enhances user experience (e.g., DOM manipulation).

# History of JavaScript

**Created by:** Brendan Eich in 1995.

**Originally called:** Mocha, then LiveScript, and finally JavaScript.

**ECMAScript:** Standardized version of JavaScript.

# JavaScript Syntax

**Variables:** *let, const, var*

Example:

```
let name = "Alice";
```

```
const age = 25;
```

# Data Types

**Primitive Types:** Strings, Numbers, Booleans, null, undefined, Symbols.

**Complex Types:** Objects, Arrays.

# Operators

**Arithmetic Operators:** +, -, \*, /, %

**Comparison Operators:** ===, !==, >, <

**Logical Operators:** &&, ||, !

# Control Structures

## **Conditional Statements:** if, else, switch

Example:

```
if (age > 18) {  
    console.log("Adult");  
} else {  
    console.log("Minor");  
}
```

# Loops

## For Loop:

Example:

```
for (let i = 0; i < 5; i++) {  
    console.log(i);  
}
```

## While Loop:

Example:

```
let i = 0;  
while (i < 5) {  
    console.log(i);  
    i++;  
}
```

# Functions

**Definition:** Block of code designed to perform a specific task.

**Syntax:**

Function Declaration:

```
function greet() {  
    console.log("Hello!");  
}
```

# DOM Manipulation

## **What is the DOM?**

Document Object Model, representing the structure of HTML.

## **Example of Manipulation:**

```
document.getElementById("myElement").innerHTML = "Hello World!";
```

# Web Development with JavaScript

## **The Role of JavaScript in Front-End Development:**

- ✓ Allows for real-time updates without reloading the page.
- ✓ Handles events like clicks, form submissions, and more.
- ✓ Works alongside HTML for structure and CSS for styling.

# **HTML, CSS, and JavaScript**

**HTML:** Structure of web pages.

**CSS:** Styling and layout.

**JavaScript:** Adds behavior and interactivity.

# JavaScript in the Browser

- Inside the <script> tag of HTML.
- As an external file linked to HTML.

# DOM (Document Object Model)

- A programming interface for HTML and XML documents.



- Represents the page as a tree of objects.

*document.getElementById("myElement").innerText = "Hello, World!";*

# DOM hierarchy

- Rooted at `window.document`
- Follows HTML document structure
  - `window.document.head`
  - `window.document.body`
- DOM objects have tons (~250) of properties, most private

# Accessing DOM Nodes

- Walk DOM hierarchy (not recommended)

```
element = document.body.firstChild.nextSibling.firstChild;
```

- Use DOM lookup method. An example using ids:

```
element = document.getElementById("div1");
```

- Many: `getElementsByClassName()`, `getElementsByTagName()`,

...

```
document.body.firstChild.getElementsByTagName()
```

# More commonly used Node properties

- textContent
- innerHTML
- getAttribute() / setAttribute()

# DOM and CSS interactions

- `element.className="active";`
- `element.style.color="#ff0000";`
- `document.querySelector(".class1")`

# More DOM operations

- `window.location.href="newPage.html";`
- `console.log("Reached point A");`
- `alert("Wow!");confirm("OK?");`

# Event Handling

## What are Events?

User actions like clicks, typing, and mouse movements.

Adding Event Listeners:

```
document.getElementById("myButton").addEventListener("click",
function() { alert("Button clicked!"); });
```

# JavaScript and DOM Events

- Mouse-related: mouse movement, button click, enter/ leave element
- Keyboard-related: down, up, press
- Focus-related: focus in, focus out (blur)
- Input field changed, Form submitted

# Event handling

1. What happened: the event of interest
2. Where it happened: an element of interest.
3. What to do: JavaScript to invoke when the event occurs on the element.

# Specifying the JavaScript of an Event

- Option #1: in the HTML:
  - `<div onclick="divClicked();">...</div>`
- Option #2: from Javascript using the DOM:
  - `element.addEventListener("click", mouseClick);`
  - `element.onclick = mouseClick;`

# Front-End Frameworks and Libraries

**Frameworks** simplifies development, enhances productivity, and promotes best practices.

**Angular:** Full-fledged MVC framework.

**jQuery :** legacy, still around but mostly avoid for new stuff.

**React:** Most widely used library for building UIs.

