****

**Web - JavaScript**

In this session, we will cover the fundamental concepts of JavaScript, which will serve as a solid foundation for your journey into web development. Let's get started:

**1. Introduction to JavaScript**

JavaScript is a high-level, interpreted programming language used to add interactivity and dynamic behavior to websites. It is commonly referred to as the **language of the web** because it runs in web browsers, making it an essential part of web development.

**2. Variables and Data Types**

In JavaScript, you can use variables to store and manipulate data. JavaScript is a **dynamically typed language**, meaning you don't need to specify the data type explicitly. The basic data types include:

**Numbers:** Used for numeric values, e.g., **var age = 25;**

**Strings:** Used for text, e.g., **let name = "John";**

**Booleans:** Used for true/false values, e.g., **let isStudent = true;**

**Arrays:** Used to store a collection of values, e.g., **const numbers = [1, 2, 3];**

**Objects:** Used to store key-value pairs, e.g., **let person = { name: "John", age: 25 };**

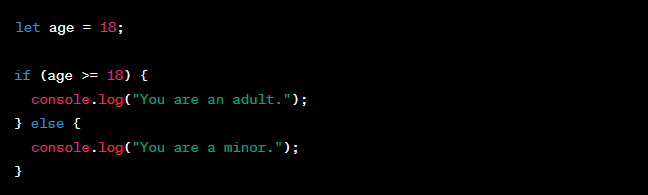
**3. Operators**

JavaScript supports various operators to perform arithmetic, comparison, and logical operations. Common operators include

**Arithmetic operators:** +, -, \*, /  
**Comparison** **operators:** ===, !==, >, <  
**Logical operations:** &&, ||, !

**4. Conditional Statements**

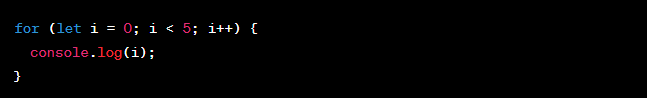
Conditional statements allow you to make decisions in your code. The most common ones are if, else if, and else. For example:



**alert(“Hello World!”);**

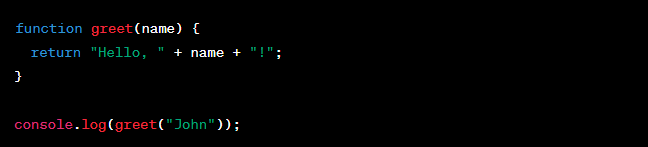
**5. Loops**

Loops enable you to execute a block of code repeatedly. The most common loops are for, while, and do-while. For example:



**6. Functions**

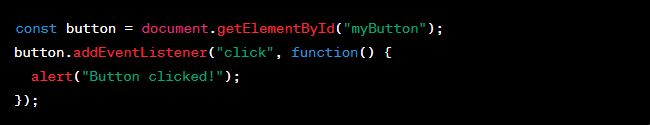
Functions are reusable blocks of code that perform specific tasks. They allow you to organize your code and make it more maintainable. Here's an example:



**7. DOM Manipulation**

The Document Object Model (DOM) allows you to interact with HTML elements on a web page. You can use JavaScript to add, modify, or remove elements dynamically. For example:





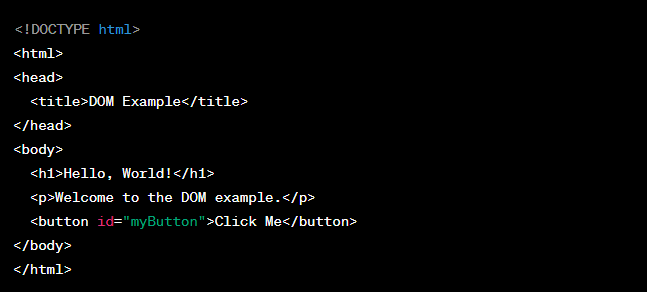
**8. Events**

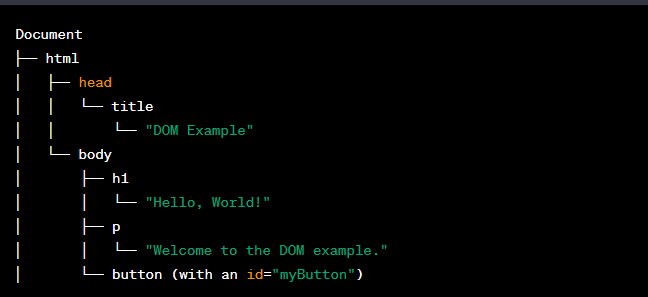
JavaScript enables you to respond to various events like clicks, keypresses, and mouse movements. Event handling allows you to make your websites interactive and responsive to user actions.

The **Document Object Model (DOM)** is a programming interface provided by web browsers that allows you to interact with HTML and XML documents. It represents the structure of a web page as a **tree of objects**, where each HTML element and its attributes are represented as nodes in the tree. With the DOM, you can manipulate the content and structure of a web page dynamically using JavaScript. Let's dive into more details and examples:

**1. Understanding the DOM Tree**

Consider the following HTML code as an example:

  
  
The DOM represents this HTML structure as a tree:



Each node in the tree is called an "element" and can have attributes and child nodes (other elements or text nodes).

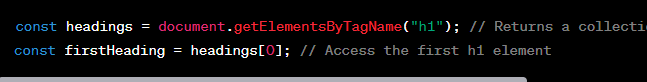
**2. Accessing DOM Elements with JavaScript**

You can use JavaScript to access and manipulate elements in the DOM. There are several methods to access elements:

**By ID:**



**By Tag Name:**

****

**By Class Name:**

****

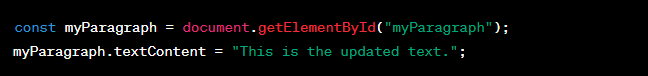
**By Query Selector:**

****

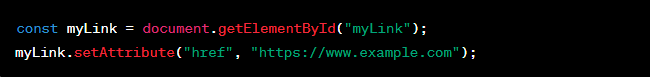
**3. Manipulating DOM Elements**

Once you have accessed an element, you can modify its content, attributes, and even add or remove elements dynamically.

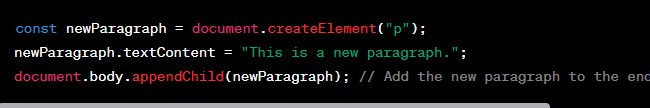
**Changing Text Content:**



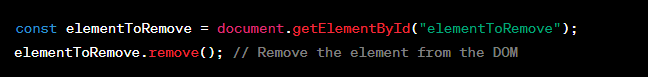
**Modifying Attributes:**



**Creating New Elements and Appending:**



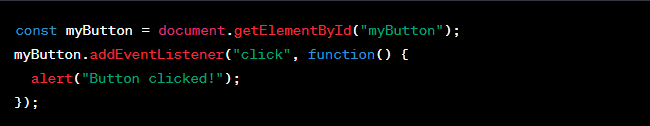
**Removing Elements:**



**4. Event Handling with DOM**

You can also use the DOM to handle events like button clicks or form submissions.





In this example, when the button is clicked, the JavaScript code inside the event listener function will execute, showing an alert with the message "Button clicked!"