****

**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 };**

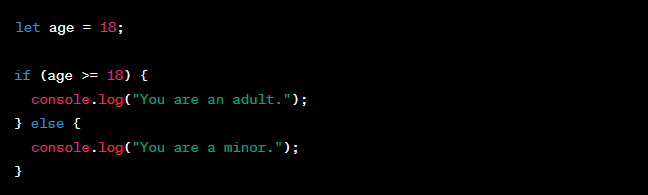
**3. Operators**

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

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

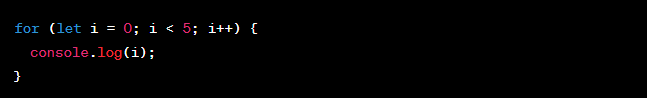
**4. Conditional Statements**

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



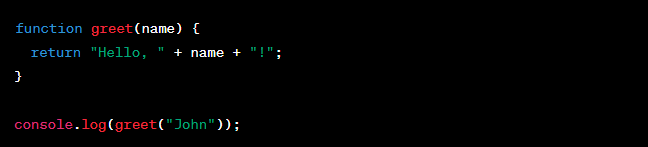
**5. Loops**

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



**6. Functions**

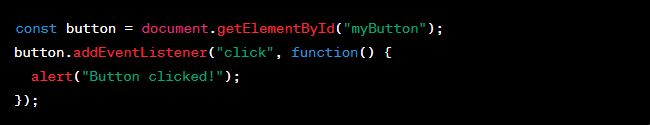
Functions are reusable blocks of code that perform specific tasks. They allow you to organize your code and make it more maintainable. Here's an example:



**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:





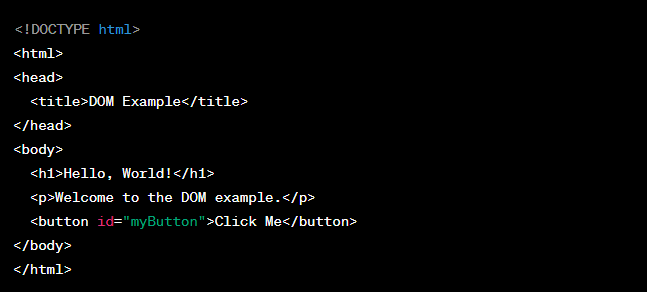
**8. Events**

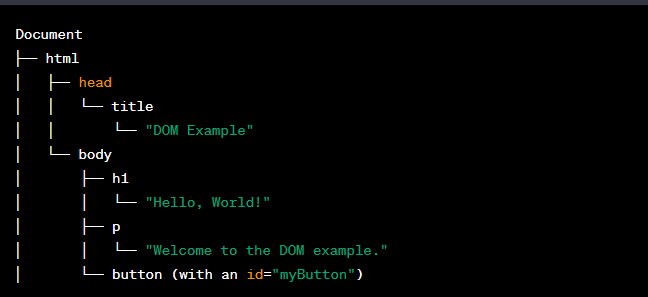
JavaScript enables you to respond to various events like clicks, keypresses, and mouse movements. Event handling allows you to make your websites interactive and responsive to user actions.

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).

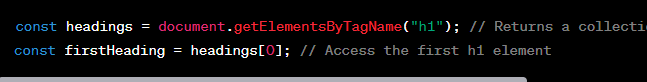
**2. Accessing DOM Elements with JavaScript**

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:**

****

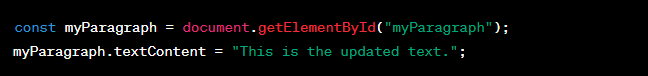
**By Query Selector:**

****

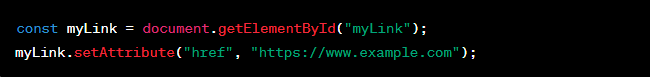
**3. Manipulating DOM Elements**

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

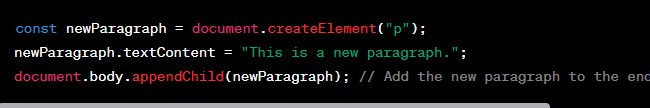
**Changing Text Content:**



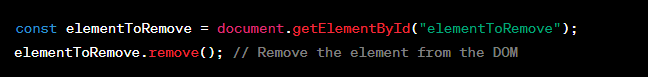
**Modifying Attributes:**



**Creating New Elements and Appending:**



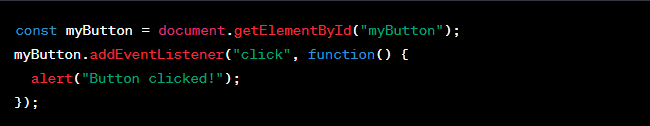
**Removing Elements:**



**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!"

In this session, we'll introduce you to **jQuery**, a fast and lightweight JavaScript library that simplifies client-side scripting and allows you to interact with HTML elements and handle events more easily.

**jQuery** is a popular ***JavaScript library*** that makes it easier to write JavaScript code for web development. It provides a set of powerful and easy-to-use functions that allow you to manipulate HTML elements, handle events, make AJAX requests, and perform various other tasks with less code and better cross-browser compatibility.

To use jQuery in your web projects, you first need to include the jQuery library. You can do this by adding the following code inside the **<head>** section of your HTML file:

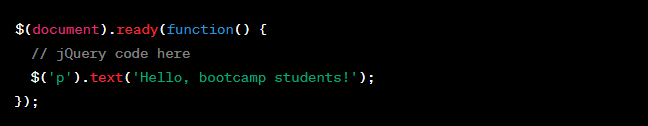


This code includes the latest version of jQuery from a Content Delivery Network (CDN). Now, you're ready to start using jQuery in your scripts.

**Basic Syntax**

The basic syntax of jQuery is similar to CSS selectors. You use the $ symbol (alias for jQuery) to select HTML elements and perform actions on them.

For example, to select all paragraphs (<p>) in your document, you can use the following code:



In this example, the **$(document).ready()** function ensures that the jQuery code executes only after the **entire HTML document has loaded**.

**DOM Manipulation**

jQuery allows you to manipulate the Document Object Model (DOM) with ease. Here are some commonly used DOM manipulation methods:

**text():** Sets or gets the text content of an element.

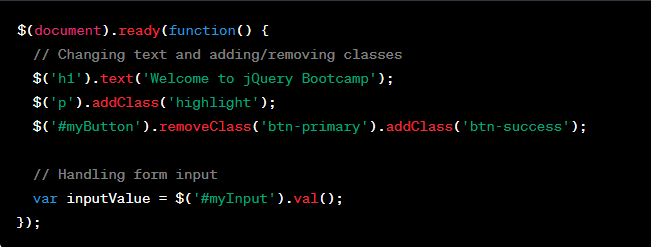
**html():** Sets or gets the HTML content of an element.

**val():** Gets or sets the value of form elements like input, select, etc.

**addClass():** Adds a CSS class to an element.

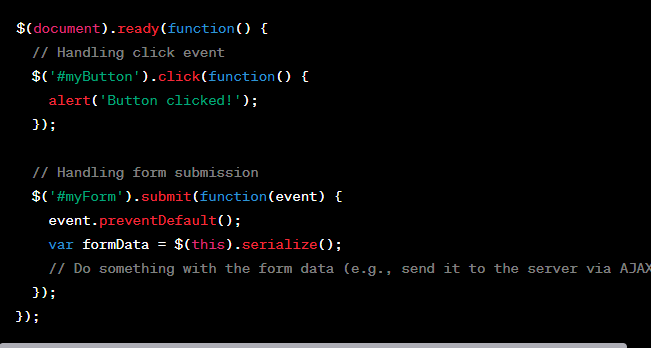
**removeClass():** Removes a CSS class from an element.

**attr():** Gets or sets attributes of an element.

****

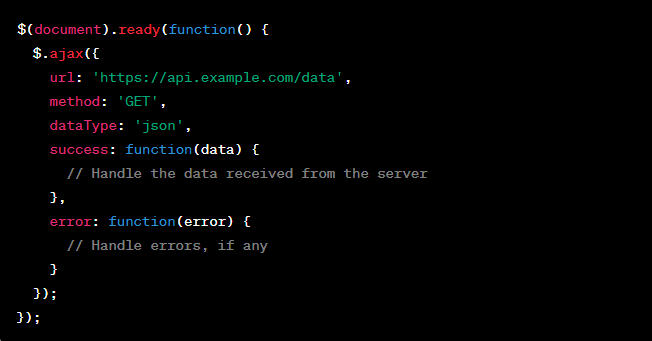
**Event Handling**

jQuery simplifies event handling. You can easily attach event listeners to elements and define what happens when the event occurs.



**AJAX Requests**

jQuery makes AJAX requests a breeze. You can use the **$.ajax()** method to fetch data from the server without reloading the page.

****

Try hitting this URL and list the data in the browser: **https://jsonplaceholder.typicode.com/posts**