Lecture-09: DOM Manipulation

- **Teacher:** Sir Arsalan Shah
- **Typist:** Yousuf Naveed

Overview:

In this lecture, we will explore DOM (Document Object Model) manipulation in JavaScript. The DOM is a representation of the HTML document, enabling JavaScript to interact and modify the content of a webpage. We will cover various methods and functionalities provided by the DOM to manipulate and control HTML elements.

Keypoints of this Lecture:

1. Document Methods:
    - document.getElementById
    - document.getElementsByClassName
    - document.getElementsByTagName
2. ClassList.add by JavaScript
3. Create tags dynamically
4. Data Insertion / Removal
5. Query Selector
6. Date
7. String methods

DOM MANIPULATION:

DOM enables JavaScript to interact with HTML elements and manipulate them dynamically. It represents the HTML document as a tree-like structure, where each element is a node.

Accessing Elements:

`document.getElementById("elementID")`: Retrieves an element by its unique ID.

Example:

```
  var element = document.getElementById("myElement");
```

`document.getElementsByClassName("className")`: Retrieves elements by their class name.

Example:

```
var elements = document.getElementsByClassName("myClass");
```

`document.getElementsByTagName("tagName")`: Retrieves elements by their tag name.

Example:

```
var elements = document.getElementsByTagName("div");
```

ClassList.add by JavaScript:

The **`classList.add()`** method is used to add a class to an HTML element.

Example:

```
var element = document.getElementById("myElement");
element.classList.add("newClass");
```

Create Tags Dynamically:

You can create HTML elements dynamically using the **`document.createElement("tagName")`** method. After creating an element, you can set its content using the innerText property and append it to another element using **`appendChild()`** method.

Example:

```
var newElement = document.createElement("p");
```

```
newElement.innerText = "This is a dynamically created paragraph.";
document.body.appendChild(newElement);
```

Replace Child:

The **`replaceChild()`** method replaces an existing child node with a new one.

Query Selector:

`querySelector` and **`querySelectorAll`** are used to select elements based on CSS selectors, allowing you to style or manipulate multiple elements at once.

Example:

```
var elements = document.querySelectorAll(".myClass");
```

Date:

JavaScript's **`Date`** class allows you to work with dates and times. You can create a new Date object and access various methods like **`getDate()`, `getDay()`,** etc.

Example:

```
var currentDate = new Date();
var day = currentDate.getDate();
```

String Methods:

Five commonly used string methods are **`length`**, **`indexOf()`**, **`replace()`**, **`replaceAll()`**, and **`toLowerCase()`**.

Example:

```
var str = "Hello, World!";
var length = str.length;
var index = str.indexOf("World");
var replaced = str.replace("World", "Universe");
var lowercase = str.toLowerCase();
```

Understanding DOM manipulation and its methods will enhance your ability to build dynamic and interactive web applications.