Web Development

Class No: 05

linkedin.com/in/m-munim1

github.com/M-Munim

+92 330 2021926

Introduction to the DOM

What is the DOM?

The DOM (Document Object Model) is a representation of the structure of an HTML document as a tree of nodes. It allows JavaScript to access, manipulate, and modify the structure and content of web pages.

DOM Structure:

HTML elements are represented as nodes (e.g., <html>, <body>, , etc.). Text content is also represented as text nodes.

Example (HTML and its DOM Structure):

```
<body>
   <h1 id="heading">Hello</h1>
   <p>This is a paragraph.</p>
</body>
```

DOM Tree Structure:

body
- h1 (with id heading)
- p (Text: "This is a paragraph.")

Parts of the Document

- console.log(document.title); // Shows the page title
- console.log(document.URL); // Shows the page URL
- console.log(document.body); // Shows the body element
- console.log(document.head); // Shows the head element
- console.log(document.images); // Shows all images on the page
- console.log(document.links); // Shows all links on the page

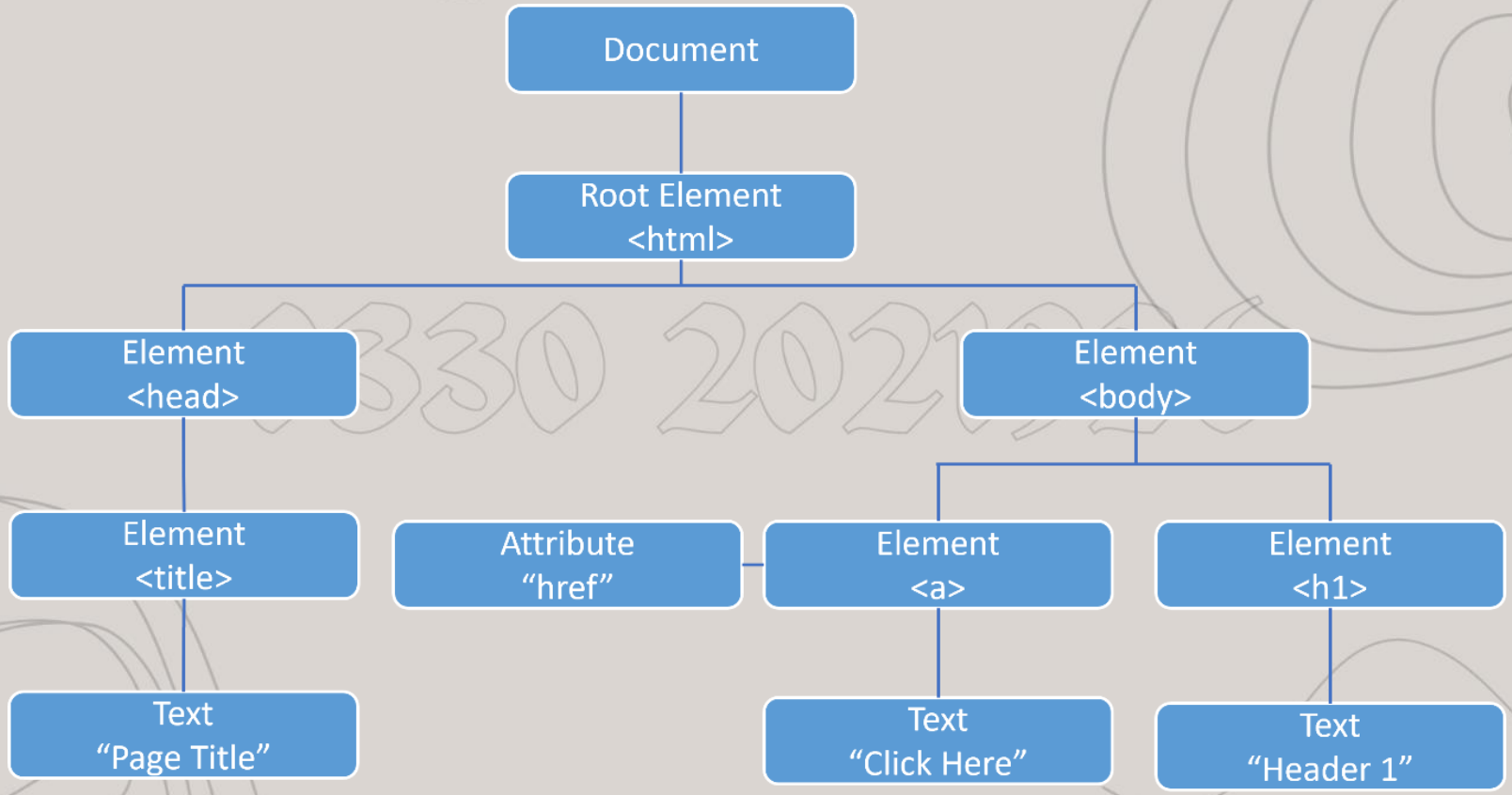
Modify the Document in Real-Time

- document.title = "Hello, Students!"; // Changes the title dynamically
- document.body.style.backgroundColor = "lightblue"; // Changes background color
- document.body.innerHTML += "<h2>Welcome to JavaScript DOM!</h2>"; // Adds a heading

Count Elements/Tag on the Page

- console.log(`This page has ${document.body.children.length} elements.`);

Web Development



linkedin.com/in/m-munim1
github.com/M-Munim
+92 330 2021926

Accessing DOM Elements

We can use JavaScript methods to access elements and manipulate them.

- document.getElementById()
  Retrieves an element by its id.

html

```
<h1 id="heading">Hello, World!</h1>
```

javascript

```
let heading = document.getElementById(“heading");
console.log(heading.innerText);                    // Output: Hello, World!
```

- document.getElementsByClassName()
  Retrieves elements by their class name (returns an HTMLCollection).

html

```
<div class="box">Box 1</div>
<div class="box">Box 2</div>
```

javascript

```
let boxes = document.getElementsByClassName("box");
console.log(boxes[1].innerText);                    // Output: Box 1
```

- document.getElementsByTagName()
  Retrieves all elements with the specified tag name.

html

```
<p>Paragraph 1</p>
<p>Paragraph 2</p>
```

javascript

```
let paragraphs = document.getElementsByTagName("p");
console.log(paragraphs[1].innerText);                    // Output: Paragraph 2
```

- querySelector()
  Selects the first element that matches the CSS selector.

html

```
<h1 id="heading“ class=“head”>Hello!</h1>
```

javascript

```
let heading = document.querySelector("#heading");
console.log(heading.innerText);                    // Output: Hello!
```

- querySelectorAll()
  Selects all elements that match the CSS selector (returns a NodeList).

html

```
<div class="box">Box 1</div>
<div class="box">Box 2</div>
```

javascript

```
let boxes = document.querySelectorAll(".box");
console.log(boxes[0].innerHTML);
```

Modifying the DOM

Change Text Content:

Using innerText or innerHTML, we can change the text inside an element.

html

```
<h1 id="heading">Original Text</h1>
```

javascript

```
let heading = document.getElementById("heading");
heading.innerText = "Updated Text";                    // Changes text to "Updated Text"
```

Change Attributes:

Using setAttribute(), we can modify the attributes of an element.

html

```
<img id="image" src="old-image.jpg“ alt=“img of car” >      // old car image
```

javascript

```
let img = document.getElementById("image");
img.setAttribute("src", "new-image.jpg");            // Changes the image source
img.setAttribute("alt", “image of new car");
```

Change Styles:

Using the style property, we can modify the element’s CSS.

javascript

```
heading.style.color = "blue";          // Changes text color to blue
heading.style.fontSize = "24px";       // Changes font size
```

Advanced DOM Manipulation and Events

Creating and Adding Elements

- createElement(): Creates a new element in JavaScript.
- appendChild(): Adds the newly created element to an existing element in the DOM.

Example:

```
<ul id="list"></ul>
```

javascript

```
let list = document.getElementById("list");
  // Creating a new list item
let newItem = document.createElement("li");
newItem.innerText = "New Item";
  // Appending the new item to the list
list.appendChild(newItem);
```

Removing and Replacing Elements

- removeChild(): Removes a specified child element from its parent.
- replaceChild(): Replaces an existing child element with another.

HTML:

```
<ul id="list">
  <li id="item">Item 1</li>
</ul>

<button id="removeBtn">Remove Item</button>
<button id="replaceBtn">Replace Item</button>
```

```
let list = document.getElementById("list");
let item = document.getElementById("item");
let removeBtn = document.getElementById("removeBtn");
let replaceBtn = document.getElementById("replaceBtn");

removeBtn.addEventListener("click", function () {                    // Removing an Element
    if (item) {
  list.removeChild(item);
   }
});

replaceBtn.addEventListener("click", function () {                   // Replacing an Element
 let newItem = document.createElement("li");
 newItem.innerText = "New Item";
   if (item) {
      list.replaceChild(newItem, item);
 }
});
```

Adding Events Using addEventListener()

Events allow us to add **interactivity** to our webpages. The addEventListener() method listens for user interactions such as clicks, key presses, and mouse movements.

HTML:

```
<button id="btn">Click Me</button>
```

JavaScript:

```
let button = document.getElementById("btn");

// Adding a click event listener
button.addEventListener("click", function () {
    alert("Button was clicked!");
});
```

HTML Form Validation

Form validation ensures users enter valid data before submitting a form. JavaScript helps check if the required fields are filled correctly.

Example:

```
<form id="myForm">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>
   <br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>
   <br>
  <button type="submit">Submit</button>
</form>
<p id="error-message" style="color:red;"></p>
```

```
let form = document.getElementById("myForm");
let errorMessage = document.getElementById("error-message");

// Adding submit event listener
form.addEventListener("submit", function (event) {
    let name = document.getElementById("name").value;
    let email = document.getElementById("email").value;

    if (name === "" || email === "") {
     errorMessage.innerText = "All fields are required!";
event.preventDefault();                          // Prevent form submission
 } else {
    alert("Form submitted successfully!");
 }
});
```