Web Development

Class No: 06

linkedin.com/in/m-munim1

github.com/M-Munim

+92 330 2021926

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);
```

0330 2021926

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>
```

0330 2021926

```
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);
 }
});
```

0330 2021926

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!");
});
```

0330 2021926

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>
```

0330 2021926

```
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!");
 }
});
```

0330 2021926

Activity-Based Task: Interactive To-Do List

Create an interactive to-do list that allows users to add tasks, mark them as completed, and delete tasks.

Steps:

1.Create an input field and a button to add tasks.
2.Display each task in a list with a "Remove" button.
3.When a task is clicked, mark it as completed (toggle the CSS class).
4.When the "Remove" button is clicked, remove the task.

0330 2021926

Example Code:

```
let taskList = document.getElementById("tasks");
let input = document.getElementById("taskInput");
let addButton = document.getElementById("addTask");

addButton.addEventListener("click", function () {
  let task = document.createElement("li");
  task.innerText = input.value;
  task.addEventListener("click", function () {
    task.classList.toggle("completed"); // Toggles "completed" class
  });

  let removeButton = document.createElement("button");
  removeButton.innerText = "Remove";
  removeButton.addEventListener("click", function () {
    taskList.removeChild(task);
  });

  task.appendChild(removeButton);
  taskList.appendChild(task);
  input.value = "";
});
```

0330 2021926

Home Task 6

1. Add new elements to the DOM using createElement().
2. Remove an element from the DOM.
3. Create a button that changes its color when clicked.
4. Create a list where each item can be marked as completed.
5. Create a form with input fields and validate that all fields are filled.

0330 2021926