Day 9 – [1st July 2025]

TOPICS COVERED

DOM Manipulation – Deeper Concepts

In JavaScript, manipulating web pages dynamically involves not only selecting elements but also modifying their content, structure, and behavior. Today, I explored how to read and modify content, attributes, and dynamically update elements in the DOM.

Content Manipulation:

textContent – Gets or sets the text content of an element, ignoring HTML tags.

```
let el = document.getElementById("demo");
el.textContent = "Hello World";  // replaces all content with plain text
```

innerText – Similar to textContent, but takes CSS styles (like display: none) into account.

```
console.log(el.innerText);  // gets visible text only
```

innerHTML – Gets or sets the HTML content (tags + text) inside an element.

```
el.innerHTML = "<strong>Bold Text</strong>";  // renders bold
```

Attribute Manipulation:

getAttribute() – Reads the value of an attribute on an element.

```
let link = document.querySelector("a");
console.log(link.getAttribute("href"));  // prints the href value
```

setAttribute() – Sets or updates the value of an attribute.

```
link.setAttribute("target", "_blank");  // opens link in a new tab
```

Adding & Removing Elements:

Create and Append Elements:

```
let newPara = document.createElement("p");
newPara.textContent = "This is a new paragraph.";
```

```
document.body.appendChild(newPara);
```

Remove Elements:

```
let oldPara = document.getElementById("removeMe");
oldPara.remove();  // removes the element from DOM
```

DOM Events:

JavaScript allows you to add interactivity to elements through event listeners.

```
let btn = document.getElementById("clickMe");
btn.addEventListener("click", function() {
    alert("Button Clicked!");
});
```

Common Events:

Event Type	Description
click	When element is clicked
mouseover	When mouse hovers
keydown	When a key is pressed
input	When input value changes

TOOLS USED:

Visual Studio Code (VS Code)

Chrome Browser (Developer Tools Console)

TASK:

Mini Practice – DOM Element Addition & Event

09_Day09_code.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Day 9 Task</title>
    <link rel="stylesheet" href="09_day9.css">
</head>
<body>
    <button id="addButton">Add Item</button>
    <ul id="list"></ul>
    <script src="09_day9.js"></script>
</body>
</html>
```

09_Day09_code.css

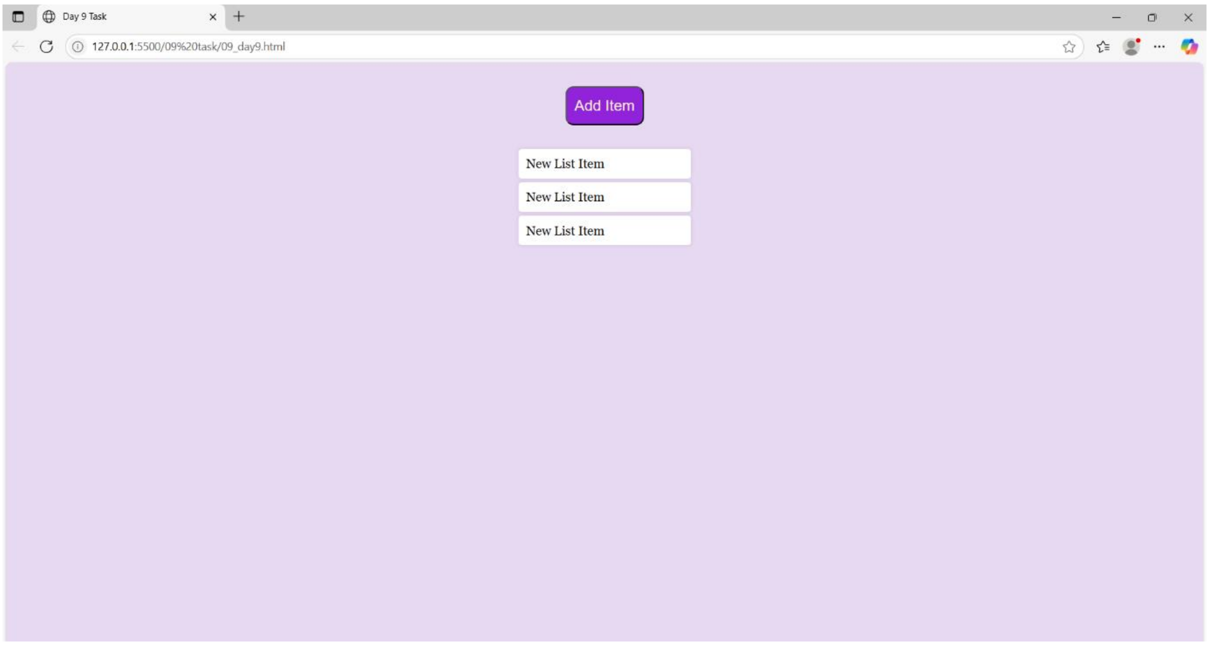
```
* {
    margin: 0;
    padding: 0;
}
body {
    background-color: #e7d9f2;
    text-align: center;
    font-family: Georgia, 'Times New Roman', Times, serif;
}
#addButton {
    background-color: #9124da;
    width: 100px;
    height: 50px;
    font-size: larger;
    color: antiquewhite;
    border-radius: 10px;
    font-weight: 100;
    margin: 30px auto;
    cursor: pointer;
    display: flex;
    justify-content: center;
    align-items: center;
    transition: linear 0.2s;
}
#addButton:hover {
    background-color: #ce4af0;
    color: black;
}
ul {
    list-style-type: none;
    padding: 0;
    margin-top: 20px;
    text-align: center;
}
li {
    background-color: #ffffff;
    padding: 10px;
    margin: 5px auto;
    width: 200px;
    border-radius: 4px;
    box-shadow: 0 0 5px rgba(0,0,0,0.1);
    display: flex;
    justify-content: space-between;
    align-items: center;
    text-align: center;
}

.delete-btn {
    background-color: crimson;
    color: white;
    border: none;
    border-radius: 5px;
    padding: 5px 8px;
    margin-left: 10px;
    cursor: pointer;
    transition: 0.2s ease-in-out;
}
.delete-btn:hover {
    background-color: darkred;
}
```

09_Day09_code.js

```
const addButton = document.getElementById("addButton");
const list = document.getElementById("list");

addButton.addEventListener("click", () => {
    const newItem = document.createElement("li");
    newItem.textContent = "New List Item";
    list.appendChild(newItem);
});
```


Day 9 Task
127.0.0.1:5500/09%20task/09_day9.html
Add Item
New List Item
New List Item
New List Item