# JavaScript DOM Manipulation – Full Tutorial with Real-World Use Case

## Table of Contents

1. What is the DOM?

2. Why Use DOM Manipulation?

3. How JavaScript Interacts with the DOM

4. DOM Manipulation Techniques

5. Real-World Use Case: Interactive To‑Do List App

6. Conclusion

---

## 1. What is the DOM?

DOM (Document Object Model) is a programming interface for web documents. It represents the structure of a web page as a tree of objects (nodes) that can be accessed and modified using JavaScript.

## 2. Why Use DOM Manipulation?

\* Create dynamic content

\* Respond to user input

\* Add or remove elements

\* Style elements dynamically

## 3. How JavaScript Interacts with the DOM

JavaScript uses the `document` object to access and manipulate the DOM, e.g.

`document.getElementById("myElement")` or `document.querySelector(".myClass")`.

## 4. DOM Manipulation Techniques

### Accessing Elements

\* By ID – `document.getElementById("title")`

\* By Class – `document.getElementsByClassName("item")`

\* By Tag – `document.getElementsByTagName("p")`

\* By Selector – `document.querySelector(".item")`

### Modifying Content

```javascript

element.textContent = "Hello";

element.innerHTML = "<b>Hello</b>";

```

### Changing Styles

```javascript

element.style.color = "red";

element.classList.add("highlight");

```

### Adding / Removing Elements

```javascript

let li = document.createElement("li");

parent.appendChild(li);

li.remove();

```

### Event Handling

```javascript

element.addEventListener("click", () => alert("Clicked!"));

```

## 5. Real‑World Use Case — Interactive To‑Do List

HTML:

```html

<input id="taskInput" />

<button onclick="addTask()">Add Task</button>

<ul id="taskList"></ul>

```

JavaScript:

```javascript

function addTask(){

const input = document.getElementById("taskInput");

const li = document.createElement("li");

li.textContent = input.value;

li.addEventListener("click", () => li.remove());

document.getElementById("taskList").appendChild(li);

input.value = "";

}

```

## 6. Conclusion

DOM manipulation lets you:

\* Update UI without reload

\* React to users interactively

\* Build modern web apps

### Summary

| Feature | Method |

|--------------------|-------------------------------------|

| Get element | `getElementById`, `querySelector` |

| Change content | `textContent`, `innerHTML` |

| Modify style | `style`, `classList` |

| Add/remove element | `createElement`, `remove` |

| Handle events | `addEventListener` |