

# ***CODE EXERCISE***


## **Introduction DOM Manipulation**

**CODING EXERCISES TEST YOUR SKILLS**



<b>Introduction DOM Manipulation</b>	<b>1</b>
<b>Exercise 1: Changing Text Content</b>	<b>2</b>
<b>Exercise 2: Adding Elements</b>	<b>3</b>
<b>Exercise 3: Removing Elements</b>	<b>3</b>
<b>Exercise 4: Handling Click Events</b>	<b>4</b>
<b>Exercise 5: Toggling Classes</b>	<b>4</b>
<b>Exercise 6: Setting Attributes</b>	<b>5</b>
<b>Exercise 7: Form Values</b>	<b>5</b>
<b>Exercise 8: Creating a To-Do List</b>	<b>6</b>
<b>Exercise 9: Changing Element Styles</b>	<b>6</b>
<b>Exercise 10: Displaying Mouse Coordinates</b>	<b>7</b>

### **Introduction DOM Manipulation**

 Excited to share 10 hands-on JavaScript DOM Manipulation exercises for beginners and intermediate developers! Whether you're looking to brush up on your skills or dive into web development, these exercises cover essential techniques from changing text content and handling events to creating dynamic to-do lists.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Each exercise is designed with a clear objective, complete code snippets, and detailed explanations to ensure you not only know "how" but also "why".

#JavaScript #WebDevelopment #LearningToCode #DOMManipulation  
#CodingExercises #FrontEndDevelopment #Programming #TechCommunity

Feel free to share your thoughts, questions, or any cool things you've built using these concepts!

## Exercise 1: Changing Text Content

Objective: Learn how to change the text content of an HTML element using JavaScript.

```
<p id="demo">This is a paragraph.</p>
```

```
<button onclick="changeText()">Change Text</button>
```

```
javascript
```

Copy code

```
function changeText() {  
  document.getElementById("demo").textContent = "Text has been changed!";  
}
```

Explanation: This exercise demonstrates how to use `getElementById()` to select an HTML element and `textContent` to change its text content.

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

## Exercise 2: Adding Elements

Objective: Learn to add new elements to the DOM.

```
<ul id="myList"><li>Item 1</li><li>Item 2</li></ul>  
<button onclick="addItem()">Add Item</button>
```

```
function addItem() {  
  const li = document.createElement("li");  
  li.textContent = "New Item";  
  document.getElementById("myList").appendChild(li);  
}
```

Explanation: Shows how to create a new element with `createElement()` and add it to an existing list using `appendChild()`.

## Exercise 3: Removing Elements

Objective: Learn to remove elements from the DOM.

```
<div id="myDiv"><button onclick="removeMe(this)">Remove Me</button></div>
```

```
function removeMe(element) {  
  document.getElementById("myDiv").removeChild(element);  
}
```

Explanation: This task teaches how to remove an HTML element using `removeChild()`.

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

## Exercise 4: Handling Click Events

Objective: Understand how to handle click events on elements.

```
<button id="clickMe">Click me</button>
```

```
document.getElementById("clickMe").addEventListener("click", function() {  
    alert("Button was clicked!");  
});
```

Explanation: Introduces event listeners and handling click events with `addEventListener()`.

## Exercise 5: Toggling Classes

Objective: Learn to toggle CSS classes for elements.

```
<p id="toggleText">Some text here.</p>  
<button onclick="toggleClass()">Toggle Class</button>
```

```
function toggleClass() {  
    document.getElementById("toggleText").classList.toggle("active");  
}
```

Explanation: Demonstrates how to use `classList.toggle()` to add or remove a CSS class from an element dynamically.

## Exercise 6: Setting Attributes

Objective: Learn how to set attributes on HTML elements.

```
<a id="myLink" href="#">Click me</a>
```

```
document.getElementById("myLink").setAttribute("href",  
"https://www.example.com");
```

Explanation: Shows how to change the attribute of an element using `setAttribute()`.

## Exercise 7: Form Values

Objective: Learn to get and set form input values.

```
<input type="text" id="myInput" value="Type here">  
<button onclick="showValue()">Show Value</button>
```

```
function showValue() {  
  alert(document.getElementById("myInput").value);  
}
```

Explanation: Teaches how to access and manipulate the value of form inputs.

## Exercise 8: Creating a To-Do List

Objective: Build a simple to-do list application.

```
<input type="text" id="todoInput">
<button onclick="addTodo()">Add</button>
<ul id="todoList"></ul>
```

```
function addTodo() {
  const input = document.getElementById("todoInput").value;
  const li = document.createElement("li");
  li.textContent = input;
  document.getElementById("todoList").appendChild(li);
}
```

Explanation: Combines element creation, event handling, and input value manipulation to create a dynamic to-do list.

## Exercise 9: Changing Element Styles

Objective: Learn to change the style of elements dynamically.

```
<p id="styledText">Styling with JavaScript.</p>
<button onclick="changeStyle()">Change Style</button>
```

```
function changeStyle() {
  const element = document.getElementById("styledText");
  element.style.color = "blue";
  element.style.fontSize = "20px";
}
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis  
Courses <https://basescripts.com/>

Explanation: Demonstrates how to modify CSS styles of elements directly through JavaScript.

## Exercise 10: Displaying Mouse Coordinates

Objective: Track and display the mouse coordinates relative to the document.

```
<p id="mousePosition">Move your mouse around.</p>
```

```
document.addEventListener("mousemove", function(event) {  
    document.getElementById("mousePosition").textContent = `X: ${event.clientX},  
Y: ${event.clientY}`;  
});
```

Explanation: Shows how to use `addEventListener()` to handle the `mousemove` event and update text content with the mouse's coordinates.