

# ***CODE EXERCISE***

## **10 JavaScript Event Handling Exercises**

**CODING EXERCISES TEST YOUR SKILLS**



<b>10 JavaScript Event Handling Exercises</b>	<b>1</b>
<b>Exercise 1: Basic Click Event</b>	<b>2</b>
<b>Exercise 2: Mouse Over Event</b>	<b>2</b>
<b>Exercise 3: Dynamic Input Validation</b>	<b>3</b>
<b>Exercise 4: Form Submit Event</b>	<b>4</b>
<b>Exercise 5: Key Press Event</b>	<b>4</b>
<b>Exercise 6: Event Delegation</b>	<b>5</b>
<b>Exercise 7: Preventing Default Action</b>	<b>6</b>
<b>Exercise 8: Toggling Element Visibility</b>	<b>6</b>
<b>Exercise 9: Changing Image on Hover</b>	<b>7</b>
<b>Exercise 10: Drag and Drop</b>	<b>7</b>

## **10 JavaScript Event Handling Exercises**

🔥 Just released: A comprehensive guide featuring 10 JavaScript Event Handling Exercises! Perfect for those looking to deepen their understanding of creating interactive web pages.

From basic click events to advanced techniques like event delegation and drag & drop interactions, these exercises cover it all. Each one comes with a detailed

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

explanation and complete code snippets, ensuring you grasp the "how" and "why" behind each task.

#JavaScript #EventHandling #WebDevelopment #InteractiveWeb #CodingExercises  
#Programming #LearnToCode #FrontEndDevelopment

Excited to see what you build with these skills! Drop your questions or share your projects below. 🙌

## Exercise 1: Basic Click Event

Objective: Learn to execute a function when a button is clicked.

```
<button id="clickButton">Click Me!</button>  
document.getElementById("clickButton").addEventListener("click", function() {  
  alert("Button clicked!");  
});
```

Explanation: This exercise introduces the `addEventListener` method to handle click events on a button, displaying an alert box as a response.

## Exercise 2: Mouse Over Event

Objective: Trigger an action when the mouse hovers over an element.

```
<div id="hoverDiv">Hover over me!</div>
```

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

```
document.getElementById("hoverDiv").addEventListener("mouseover", function()
{
  this.style.backgroundColor = "yellow";
});
```

Explanation: Demonstrates using the mouseover event to change the background color of a div when the mouse pointer is over it.

### Exercise 3: Dynamic Input Validation

Objective: Validate input as the user types.

```
<input type="text" id="textInput" placeholder="Enter text...">
<p id="feedback"></p>
document.getElementById("textInput").addEventListener("input", function() {
  const feedback = document.getElementById("feedback");
  if(this.value.length < 5) {
    feedback.textContent = "Input must be at least 5 characters.";
  } else {
    feedback.textContent = "";
  }
});
```

Explanation: Teaches real-time input validation using the input event, providing immediate feedback to the user.

## Exercise 4: Form Submit Event

Objective: Handle a form submission with JavaScript.

```
<form id="myForm">
  <input type="text" placeholder="Name">
  <button type="submit">Submit</button>
</form>

document.getElementById("myForm").addEventListener("submit",
function(event) {
  event.preventDefault(); // Prevent form from submitting
  alert("Form submitted!");
});
```

Explanation: Focuses on handling form submissions using the submit event, preventing the default form submission with preventDefault().

## Exercise 5: Key Press Event

Objective: Execute an action when a specific key is pressed.

```
<input type="text" id="keyInput" placeholder="Press 'Enter'...">

document.getElementById("keyInput").addEventListener("keypress",
function(event) {
  if(event.key === "Enter") {
    alert("Enter key pressed!");
  }
});
```

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

Explanation: Introduces handling keyboard events, specifically detecting the "Enter" key press within an input field.

## Exercise 6: Event Delegation

Objective: Use event delegation to handle clicks on multiple items.

```
<ul id="itemList">
```

```
  <li>Item 1</li>
```

```
  <li>Item 2</li>
```

```
  <li>Item 3</li>
```

```
</ul>
```

```
document.getElementById("itemList").addEventListener("click", function(event) {  
  if(event.target.tagName === "LI") {  
    alert(event.target.textContent + " clicked!");  
  }  
});
```

Explanation: Demonstrates event delegation, allowing a single event listener to manage clicks on multiple elements, improving performance and memory consumption.

## Exercise 7: Preventing Default Action

Objective: Prevent the default action of an anchor tag and replace it with custom behavior.

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

```
<a href="https://www.example.com" id="preventLink">Go to Example.com</a>
document.getElementById("preventLink").addEventListener("click",
function(event) {
  event.preventDefault();
  alert("Default action prevented!");
});
```

Explanation: Shows how to use `preventDefault()` to stop the default navigation action of an anchor tag, enabling custom behavior instead.

## Exercise 8: Toggling Element Visibility

Objective: Toggle the visibility of an element on button click.

```
<button id="toggleButton">Toggle Visibility</button>
<div id="toggleDiv">Toggle me!</div>
document.getElementById("toggleButton").addEventListener("click", function() {
  const div = document.getElementById("toggleDiv");
  div.style.display = div.style.display === "none" ? "block" : "none";
});
```

Explanation: Uses click events to toggle the CSS `display` property of an element, showing or hiding it.

## Exercise 9: Changing Image on Hover

Objective: Change an image when the mouse hovers over it.

```

const img = document.getElementById("changeImage");
img.addEventListener("mouseover", function() {
  this.src = "image2.jpg";
});
img.addEventListener("mouseout", function() {
  this.src = "image1.jpg";
});
```

Explanation: Teaches how to swap images using the mouseover and mouseout events, enhancing user interaction.

## Exercise 10: Drag and Drop

Objective: Implement a basic drag and drop interaction.

```
<div id="dragDiv" draggable="true" style="width: 100px; height: 100px;
background: red;"></div>
const dragDiv = document.getElementById("dragDiv");
dragDiv.addEventListener("dragstart", function(event) {
  event.dataTransfer.setData("text/plain", event.target.id);
});
```

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

Explanation: Introduces the dragstart event and the DataTransfer object, laying the groundwork for more complex drag-and-drop interactions.