

Manipulating Element Style in JavaScript



1. Changing CSS Properties Directly:	4
2. Adding and Removing CSS Classes:	5
3. Manipulating Element Dimensions:	7
10 Exercises for manipulating element styles	9
Exercise 1: Change Text Color	9
Exercise 2: Toggle Visibility	10
Exercise 3: Increase Font Size	12
Exercise 4: Highlight on Click	13
Exercise 5: Add and Remove Class	14
Exercise 6: Toggle Font Style	16
Exercise 7: Reset Styles	18
Exercise 8: Increase Width	20
Exercise 9: Hover Effect	22
Exercise 10: Change Border Color	23
10 quiz questions	25
Quiz Question 1:	25
Quiz Question 2:	26
Quiz Question 3:	26
Quiz Question 4:	26

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

Quiz Question 5:	27
Quiz Question 6:	27
Quiz Question 7:	28
Quiz Question 8:	28
Quiz Question 9:	28
Quiz Question 10:	29
10 multiple-choice questions	29
Multiple Choice Question 1:	29
Multiple Choice Question 2:	30
Multiple Choice Question 3:	30
Multiple Choice Question 4:	31
Multiple Choice Question 5:	31
Multiple Choice Question 6:	32
Multiple Choice Question 7:	32
Multiple Choice Question 8:	33
Multiple Choice Question 9:	33
Multiple Choice Question 10:	34

JavaScript offers powerful capabilities for dynamically manipulating the styles of HTML elements, allowing developers to change how elements appear and behave on webpages based on user interactions and other conditions. This guide explores the art of manipulating element styles using JavaScript and provides practical coding examples to illustrate these techniques.

- Changing CSS Properties Directly: JavaScript can directly access and modify CSS properties of HTML elements. For example, you can change an element's background color and text color by using the style property.

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

- Adding and Removing CSS Classes: Another approach is to add or remove CSS classes from elements. This is useful when you have predefined styles in your CSS and want to apply or remove them as needed. JavaScript can add and remove classes based on user interactions.
- Manipulating Element Dimensions: JavaScript can also manipulate element dimensions such as width and height. This enables you to create dynamic resizing effects for elements on your webpage.

The provided coding examples demonstrate how to implement these concepts, from changing text colors to toggling element visibility and resizing elements gradually. Additionally, there are ten exercises that provide hands-on practice for manipulating element styles, and ten quiz questions (multiple-choice and regular) to test your understanding of the topic.

By mastering these techniques, developers can create visually appealing and interactive web applications, making the most of JavaScript's capabilities in web development.

One of the powerful capabilities of JavaScript in web development is the ability to manipulate the styles of HTML elements dynamically. This allows you to change how elements look and behave on a webpage based on user interactions or other conditions. In this guide, we'll delve into the art of manipulating element styles using JavaScript, complete with coding examples to illustrate these techniques.

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

1. Changing CSS Properties Directly:

You can directly access and modify CSS properties of an element in JavaScript.

Here's a basic example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Style Manipulation Example</title>
  <style>
    #myElement {
      background-color: lightblue;
      color: black;
    }
  </style>
</head>
<body>
  <div id="myElement">This is my element.</div>
  <script>
    const element =
document.getElementById("myElement");

    // Change the background color and font color
```

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

```
        element.style.backgroundColor = "lightgreen";
        element.style.color = "white";
    </script>
</body>
</html>
```

In this example, we first define a CSS style for an element with the ID 'myElement'. Then, in JavaScript, we access that element by its ID and change its background color and text color directly using the style property.

2. Adding and Removing CSS Classes:

Another common approach is to add or remove CSS classes from an element. This is particularly useful when you have predefined styles in your CSS and want to apply or remove them as needed:

```
<!DOCTYPE html>
<html>
<head>
    <title>Style Manipulation Example</title>
    <style>
        .highlighted {
            background-color: yellow;
            color: black;
        }
    </style>
</head>
</html>
```

```
    </style>
</head>
<body>
  <div id="myElement">This is my element.</div>
  <button id="highlightButton">Highlight</button>
  <button id="unhighlightButton">Remove
Highlight</button>
  <script>
    const element =
document.getElementById("myElement");
    const highlightButton =
document.getElementById("highlightButton");
    const unhighlightButton =
document.getElementById("unhighlightButton");

    // Add a class to highlight the element
highlightButton.addEventListener("click", () =>
{
    element.classList.add("highlighted");
});

    // Remove the class to remove the highlight
```

```
        unhighlightButton.addEventListener("click", ()
=> {
            element.classList.remove("highlighted");
        });
    </script>
</body>
</html>
```

In this example, we define a CSS class called 'highlighted'. We then use JavaScript to add and remove this class from the element when buttons are clicked.

3. Manipulating Element Dimensions:

JavaScript can also manipulate element dimensions, such as width and height.

Here's a basic example of resizing an element:

```
<!DOCTYPE html>
<html>
<head>
    <title>Style Manipulation Example</title>
    <style>
        #resizableElement {
            width: 100px;
            height: 100px;
            background-color: lightblue;
```

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

```

    }
  </style>
</head>
<body>
  <div id="resizableElement"></div>
  <button id="resizeButton">Resize</button>
  <script>
    const resizableElement =
document.getElementById("resizableElement");
    const resizeButton =
document.getElementById("resizeButton");

    // Resize the element when the button is
clicked
    resizeButton.addEventListener("click", () => {
      resizableElement.style.width = "200px";
      resizableElement.style.height = "200px";
    });
  </script>
</body>
</html>

```

In this example, we have an element with a defined width and height in CSS. When the button is clicked, JavaScript modifies the element's width and height.

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

These examples showcase just a glimpse of what you can achieve with JavaScript when it comes to manipulating element styles. You can animate elements, toggle styles based on user interactions, and create dynamic, engaging web experiences. By mastering these techniques, you'll have a powerful toolset for crafting visually appealing and interactive web applications.

10 Exercises for manipulating element styles

Exercise 1: Change Text Color

Description: Change the text color of an element to red when a button is clicked.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exercise 1</title>
</head>
<body>
  <p id="myParagraph">This is a paragraph.</p>
  <button id="changeColorButton">Change
Color</button>
  <script>
    // Your code here
  </script>
```

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

```
</body>
```

```
</html>
```

Instructions:

1. Select the element with ID 'myParagraph'.
2. Select the button with ID 'changeColorButton'.
3. Add a click event listener to the button that changes the text color of the paragraph to red.

Solution:

```
const myParagraph =  
document.getElementById("myParagraph");  
const changeColorButton =  
document.getElementById("changeColorButton");  
  
changeColorButton.addEventListener("click", () => {  
    myParagraph.style.color = "red";  
});
```

Exercise 2: Toggle Visibility

Description: Toggle the visibility of an element when a button is clicked.

Code Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
    <title>Exercise 2</title>
</head>
<body>
    <div id="myDiv">This is a hidden div.</div>
    <button id="toggleButton">Toggle
Visibility</button>
    <script>
        // Your code here
    </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myDiv'.
2. Select the button with ID 'toggleButton'.
3. Add a click event listener to the button that toggles the visibility of the div.

Solution:

```
const myDiv = document.getElementById("myDiv");
const toggleButton =
document.getElementById("toggleButton");

toggleButton.addEventListener("click", () => {
    if (myDiv.style.display === "none" ||
myDiv.style.display === "") {
```

```
        myDiv.style.display = "block";
    } else {
        myDiv.style.display = "none";
    }
});
```

Exercise 3: Increase Font Size

Description: Increase the font size of an element when the mouse hovers over it.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Exercise 3</title>
</head>
<body>
    <p id="myParagraph">Hover over me to increase font
size.</p>
    <script>
        // Your code here
    </script>
</body>
</html>
```

Instructions:

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

1. Select the element with ID 'myParagraph'.
2. Add a mouseover event listener to the paragraph that increases its font size when hovered over.
3. Add a mouseout event listener to reset the font size when the mouse moves out.

Solution:

```
const myParagraph =
document.getElementById("myParagraph");

myParagraph.addEventListener("mouseover", () => {
    myParagraph.style.fontSize = "20px";
});

myParagraph.addEventListener("mouseout", () => {
    myParagraph.style.fontSize = "16px";
});
```

Exercise 4: Highlight on Click

Description: Change the background color of an element when it's clicked.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
```

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

```
    <title>Exercise 4</title>
</head>
<body>
    <div id="myDiv">Click me to highlight</div>
    <script>
        // Your code here
    </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myDiv'.
2. Add a click event listener to the div that changes its background color when clicked.

Solution:

```
const myDiv = document.getElementById("myDiv");

myDiv.addEventListener("click", () => {
    myDiv.style.backgroundColor = "yellow";
});
```

Exercise 5: Add and Remove Class

Description: Add and remove a CSS class from an element to change its appearance.

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

Code Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exercise 5</title>
  <style>
    .highlighted {
      background-color: yellow;
      color: black;
    }
  </style>
</head>
<body>
  <p id="myParagraph">This is a paragraph.</p>
  <button id="addClassButton">Add Class</button>
  <button id="removeClassButton">Remove
Class</button>
  <script>
    // Your code here
  </script>
</body>
</html>
```

Instructions:

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

1. Select the element with ID 'myParagraph'.
2. Select the buttons with IDs 'addClassButton' and 'removeClassButton'.
3. Add click event listeners to the buttons to add and remove the 'highlighted' class from the paragraph.

Solution:

```
const myParagraph =
document.getElementById("myParagraph");
const addClassButton =
document.getElementById("addClassButton");
const removeClassButton =
document.getElementById("removeClassButton");

addClassButton.addEventListener("click", () => {
  myParagraph.classList.add("highlighted");
});

removeClassButton.addEventListener("click", () => {
  myParagraph.classList.remove("highlighted");
});
```

Exercise 6: Toggle Font Style

Description: Toggle between different font styles (e.g., bold and italic) for an element when a button is clicked.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exercise 6</title>
</head>
<body>
  <p id="myParagraph">Toggle Font Style</p>
  <button id="toggleStyleButton">Toggle
Style</button>
  <script>
    // Your code here
  </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myParagraph'.
2. Select the button with ID 'toggleStyleButton'.
3. Add a click event listener to the button that toggles between different font styles (e.g., bold and italic) for the paragraph.

Solution:

```
const myParagraph =
document.getElementById("myParagraph");
```

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

```
const toggleStyleButton =
document.getElementById("toggleStyleButton");
let isBold = false;
let isItalic = false;

toggleStyleButton.addEventListener("click", () => {
  if (!isBold && !isItalic) {
    myParagraph.style.fontWeight = "bold";
    isBold = true;
  } else if (isBold && !isItalic) {
    myParagraph.style.fontWeight = "normal";
    myParagraph.style.fontStyle = "italic";
    isItalic = true;
  } else {
    myParagraph.style.fontStyle = "normal";
    isItalic = false;
    isBold = false;
  }
});
```

Exercise 7: Reset Styles

Description: Reset the styles of an element to their default values when a button is clicked.

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

Code Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exercise 7</title>
</head>
<body>
  <p id="myParagraph" style="color: blue; font-size:
20px;">Styled Paragraph</p>
  <button id="resetStylesButton">Reset
Styles</button>
  <script>
    // Your code here
  </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myParagraph'.
2. Select the button with ID 'resetStylesButton'.
3. Add a click event listener to the button that resets the styles of the paragraph to their default values.

Solution:

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

```
const myParagraph =
document.getElementById("myParagraph");
const resetStylesButton =
document.getElementById("resetStylesButton");

resetStylesButton.addEventListener("click", () => {
  myParagraph.style.color = "";
  myParagraph.style.fontSize = "";
  myParagraph.style.fontWeight = "";
  myParagraph.style.fontStyle = "";
});
```

Exercise 8: Increase Width

Description: Increase the width of an element gradually when a button is clicked.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exercise 8</title>
  <style>
    #myDiv {
      width: 100px;
      height: 50px;
```

```
        background-color: lightblue;
    }
</style>
</head>
<body>
    <div id="myDiv">Resizable Div</div>
    <button id="increaseWidthButton">Increase
Width</button>
    <script>
        // Your code here
    </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myDiv'.
2. Select the button with ID 'increaseWidthButton'.
3. Add a click event listener to the button that gradually increases the width of the div by a certain amount each time it's clicked.

Solution:

```
const myDiv = document.getElementById("myDiv");
const increaseWidthButton =
document.getElementById("increaseWidthButton");
let currentWidth = 100; // Initial width
```

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

```
increaseWidthButton.addEventListener("click", () => {
    currentWidth += 10; // Increase width by 10 pixels
    myDiv.style.width = currentWidth + "px";
});
```

Exercise 9: Hover Effect

Description: Create a hover effect for an element that changes its background color when hovered over.

Code Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Exercise 9</title>
</head>
<body>
    <div id="myDiv">Hover over me</div>
    <script>
        // Your code here
    </script>
</body>
</html>
```

Instructions:

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

1. Select the element with ID 'myDiv'.
2. Add a mouseover event listener to the div that changes its background color when hovered over.
3. Add a mouseout event listener to reset the background color when the mouse moves out.

Solution:

```
const myDiv = document.getElementById("myDiv");
```

```
myDiv.addEventListener("mouseover", () => {  
    myDiv.style.backgroundColor = "lightgreen";  
});
```

```
myDiv.addEventListener("mouseout", () => {  
    myDiv.style.backgroundColor = "";  
});
```

Exercise 10: Change Border Color

Description: Change the border color of an element when a button is clicked.

Code Example:

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
    <title>Exercise 10</title>
```

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

```
<style>
  #myDiv {
    width: 100px;
    height: 100px;
    border: 2px solid blue;
  }
</style>
</head>
<body>
  <div id="myDiv">Click to change border color</div>
  <button id="changeBorderColorButton">Change Border
Color</button>
  <script>
    // Your code here
  </script>
</body>
</html>
```

Instructions:

1. Select the element with ID 'myDiv'.
2. Select the button with ID 'changeBorderColorButton'.
3. Add a click event listener to the button that changes the border color of the div to a different color.

Solution:

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


```
const myDiv = document.getElementById("myDiv");
const changeBorderColorButton =
document.getElementById("changeBorderColorButton");

changeBorderColorButton.addEventListener("click", () =>
{
    myDiv.style.borderColor = "red";
});
```

These coding exercises cover various aspects of manipulating element styles in JavaScript. They provide hands-on practice for dynamically changing styles, enhancing interactivity, and creating engaging web experiences. Enjoy experimenting with these exercises to strengthen your skills in working with element styles!

10 quiz questions

Quiz Question 1:

How can you change the text color of an HTML element when a button is clicked?

Answer 1:

You can change the text color of an HTML element when a button is clicked by using the `style.color` property in JavaScript.

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

Quiz Question 2:

What property allows you to toggle the visibility of an HTML element when a button is clicked?

Answer 2:

You can toggle the visibility of an HTML element when a button is clicked by changing the `style.display` property to "none" or "block."

Quiz Question 3:

How can you increase the font size of an element when the mouse hovers over it and reset it when the mouse moves out?

Answer 3:

You can increase the font size of an element when the mouse hovers over it and reset it when the mouse moves out by adding `mouseover` and `mouseout` event listeners and using the `style.fontSize` property.

Quiz Question 4:

What property allows you to change the background color of an HTML element when it's clicked?

Answer 4:

You can change the background color of an HTML element when it's clicked by using the `style.backgroundColor` property.

Quiz Question 5:

How do you add a CSS class to an HTML element to change its appearance?

Answer 5:

You can add a CSS class to an HTML element to change its appearance by using the `classList.add()` method.

Quiz Question 6:

What property allows you to reset the styles of an HTML element to their default values?

Answer 6:

You can reset the styles of an HTML element to their default values by setting the individual style properties to an empty string (`""`).

Quiz Question 7:

How can you gradually increase the width of an HTML element when a button is clicked?

Answer 7:

You can gradually increase the width of an HTML element when a button is clicked by modifying the `style.width` property in JavaScript.

Quiz Question 8:

What property can you use to create a hover effect that changes the background color of an element when hovered over?

Answer 8:

You can create a hover effect that changes the background color of an element when hovered over by using the `mouseover` and `mouseout` event listeners and the `style.backgroundColor` property.

Quiz Question 9:

What property allows you to change the border color of an HTML element when a button is clicked?

Answer 9:

You can change the border color of an HTML element when a button is clicked by using the `style.borderColor` property.

Quiz Question 10:

What method is used to add an event listener to an HTML element in JavaScript?

Answer 10:

You can add an event listener to an HTML element in JavaScript using the `addEventListener()` method.

These quiz questions cover various aspects of manipulating element styles in JavaScript and should help reinforce your understanding of the topic.

10 multiple-choice questions

Multiple Choice Question 1:

How can you change the text color of an HTML element when a button is clicked?

- A) Using `style.background`
- B) Using `style.text`
- C) Using `style.color`
- D) Using `style.visibility`

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

Answer 1: C) Using style.color

Multiple Choice Question 2:

What property allows you to toggle the visibility of an HTML element when a button is clicked?

- A) style.display
- B) style.visibility
- C) style.opacity
- D) style.position

Answer 2: A) style.display

Multiple Choice Question 3:

How can you increase the font size of an element when the mouse hovers over it and reset it when the mouse moves out?

- A) Using style.fontSize property
- B) Using style.font
- C) Using style.textSize
- D) Using style.hover

Answer 3: A) Using style.fontSize property

Multiple Choice Question 4:

What property allows you to change the background color of an HTML element when it's clicked?

- A) style.color
- B) style.backgroundColor
- C) style.textColor
- D) style.background

Answer 4: B) style.backgroundColor

Multiple Choice Question 5:

How do you add a CSS class to an HTML element to change its appearance?

- A) Using classList.modify()
- B) Using classList.create()
- C) Using classList.toggle()
- D) Using classList.add()

Answer 5: D) Using classList.add()

Multiple Choice Question 6:

What property can you use to create a hover effect that changes the background color of an element when hovered over?

- A) `style.hoverColor`
- B) `style.hoverBackground`
- C) `style.onHover`
- D) `style.backgroundColor`

Answer 6: D) `style.backgroundColor`

Multiple Choice Question 7:

How can you gradually increase the width of an HTML element when a button is clicked?

- A) Using `style.size`
- B) Using `style.width`
- C) Using `style.height`
- D) Using `style.increase`

Answer 7: B) Using `style.width`

Multiple Choice Question 8:

What property allows you to reset the styles of an HTML element to their default values?

- A) style.reset
- B) style.clear
- C) style.default
- D) Setting individual style properties to an empty string ("")

Answer 8: D) Setting individual style properties to an empty string ("")

Multiple Choice Question 9:

What property allows you to change the border color of an HTML element when a button is clicked?

- A) style.border
- B) style.bordercolor
- C) style.borderColor
- D) style.border-style

Answer 9: C) style.borderColor

Multiple Choice Question 10:

What method is used to add an event listener to an HTML element in JavaScript?

- A) addEvent()
- B) attachEvent()
- C) addEventListener()
- D) eventListen()

Answer 10: C) addEventListener()

These multiple-choice questions cover various aspects of manipulating element styles in JavaScript and should help test and reinforce your understanding of the topic.