

JavaScript DOM to Update Styles

Recap to DOM Manipulation

- What is the DOM?
- Importance of manipulating the DOM
- Basic methods to access DOM elements

Selecting Elements

- `getElementById`
- `getElementsByClassName`
- `querySelector` and `querySelectorAll`
- Examples and practice

getElementById

`getElementById` is a method that returns the element that has the ID attribute with the specified value.

Syntax

```
document.getElementById( 'id' );
```

Example

```
<!DOCTYPE html>
<html>
<body>

<h1 id="myHeader">Hello World!</h1>

<script>
    var element = document.getElementById('myHeader');
    console.log(element.innerHTML); // Outputs: Hello World!
</script>

</body>
</html>
```

getElementsByClassName

`getElementsByClassName` returns a collection of all elements in the document with the specified class name.

```
document.getElementsByClassName('className');
```

```
<!DOCTYPE html>
<html>
<body>
<div class="myClass">Hello World!</div>
<div class="myClass">Hello Again!</div>
<script>
    var elements = document.getElementsByClassName('myClass');
    console.log(elements.length); // Outputs: 2
    console.log(elements[0].innerHTML); // Outputs: Hello World!
</script>
</body>
</html>
```

querySelector

`querySelector` returns the first element that matches a specified CSS selector(s) in the document.

Syntax

```
document.querySelector('selector');
```

Example

```
<!DOCTYPE html>
<html>
<body>

<p class="myClass">Hello World!</p>
<p class="myClass">Hello Again!</p>

<script>
    var element = document.querySelector('.myClass');
    console.log(element.innerHTML); // Outputs: Hello World!
</script>

</body>
</html>
```


querySelectorAll

`querySelectorAll` returns a static `NodeList` of all elements that match a specified CSS selector(s) in the document.

Syntax

```
document.querySelectorAll('selector');
```

Example

```
<!DOCTYPE html>
<html>
<body>

<p class="myClass">Hello World!</p>
<p class="myClass">Hello Again!</p>

<script>
    var elements = document.querySelectorAll('.myClass');
    console.log(elements.length); // Outputs: 2
    console.log(elements[1].innerHTML); // Outputs: Hello Again!
</script>

</body>
</html>
```

Changing Element Styles

- Inline styles vs. CSS classes
- Using `style` property
- Adding and removing classes
- Examples and practice

What is the `style` Property?

- The `style` property allows you to directly apply CSS styles to an HTML element.
- It is a property of the DOM element.

Syntax

```
element.style.property = "value";
```

Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Style Property Example</title>
</head>
<body>
  <p id="myParagraph">Hello, World!</p>
  <button onclick="changeStyle()">Change Style</button>

  <script>
    function changeStyle() {
      document.getElementById("myParagraph").style.color = "blue";
      document.getElementById("myParagraph").style.fontSize = "20px";
    }
  </script>
</body>
</html>
```

Introduction to `classList`

- The `classList` property is a read-only property that returns a live `DOMTokenList` collection of the class attributes of the element.
- It provides methods to add, remove, toggle, and check for the presence of CSS classes.

Adding Classes

```
<button id="myButton">Click Me!</button>
<script>
  const button = document.getElementById('myButton');
  button.classList.add('active');
</script>
```

- Use `element.classList.add('className')` to add a class to an element.

Removing Classes

```
<button id="myButton" class="active">Click Me!</button>
<script>
  const button = document.getElementById('myButton');
  button.classList.remove('active');
</script>
```

- Use `element.classList.remove('className')` to remove a class from an element.

Toggling Classes

```
<button id="myButton">Click Me!</button>
<script>
  const button = document.getElementById('myButton');
  button.classList.toggle('active');
</script>
```

- Use `element.classList.toggle('className')` to toggle a class on or off.

Checking for Classes

```
<button id="myButton" class="active">Click Me!</button>
<script>
  const button = document.getElementById('myButton');
  if (button.classList.contains('active')) {
    console.log('Button is active');
  }
</script>
```

- Use `element.classList.contains('className')` to check if an element has a specific class.

Practical Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ClassList Example</title>
  <style>
    .highlight {
      background-color: yellow;
    }
  </style>
</head>
```

```
<body>
  <p id="myParagraph">Hover over this text to highlight it.</p>
  <script>
    const paragraph = document.getElementById('myParagraph');
    paragraph.addEventListener('mouseover', () => {
      paragraph.classList.add('highlight');
    });
    paragraph.addEventListener('mouseout', () => {
      paragraph.classList.remove('highlight');
    });
  </script>
</body>
</html>
```

Event Listeners

- Adding event listeners to elements
- `addEventListener` method
- Common events: `click`, `mouseover`, `mouseout`
- Examples and practice

Practice Example

- Create a simple webpage
- Add a button to change the background color
- Use DOM methods to update styles

Class Project: Dark/Light Theme Toggle Button

- Project overview
- HTML structure for the toggle button
- CSS for dark and light themes
- JavaScript to toggle themes

HTML Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Theme Toggle</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <button id="theme-toggle">Toggle Theme</button>
  <script src="script.js"></script>
</body>
</html>
```

CSS for Themes

```
body {  
    transition: background-color 0.3s, color 0.3s;  
}  
  
body.light-theme {  
    background-color: white;  
    color: black;  
}  
  
body.dark-theme {  
    background-color: black;  
    color: white;  
}
```

JavaScript to Toggle Themes

```
document.getElementById('theme-toggle').addEventListener('click', function() {  
    document.body.classList.toggle('dark-theme');  
    document.body.classList.toggle('light-theme');  
});
```