

# JavaScript

## DOM

## Manipulation



Sayyed Siddique







# JavaScript DOM Manipulation

Enhancing Web Interactivity

**The Document Object Model (DOM) Is The Backbone Of Any Webpage. Let's Dive Into How JavaScript Can Make Your Web Pages More Interactive And Dynamic Through DOM Manipulation!**



Sayyed Siddique



# 2 Selecting Elements

JavaScript Provides Several Methods To Select HTML Elements:

1. **getElementById()**
2. **getElementsByClassName()**
3. **getElementsByTagName()**
4. **querySelector()**
5. **querySelectorAll()**



Sayyed Siddique



# Example Of Selecting Elements

```
// Select element by ID
const element = document.getElementById('myElement');

// Select elements by class name
const elementsByClass = document.getElementsByClassName('myClass');

// Select elements by tag name
const elementsByTag = document.getElementsByTagName('div');

// Select a single element using a CSS selector
const singleElement = document.querySelector('.myClass');

// Select multiple elements using a CSS selector
const multipleElements = document.querySelectorAll('.myClass');
```



Sayyed Siddique



# 3 Modifying Elements

Once Selected, You Can Easily Modify Elements. Here Are Some Common Methods:

1. `innerHTML`
2. `textContent`
3. `setAttribute()`
4. `style`



Sayyed Siddique



# Example Of Modifying Elements

```
const element = document.getElementById('myElement');

// Change inner HTML
element.innerHTML = '<p>Hello, World!</p>';

// Change text content
element.textContent = 'Hello, World!';

// Set an attribute
element.setAttribute('class', 'newClass');

// Change CSS styles
element.style.color = 'blue';
element.style.fontSize = '20px';
```



Sayyed Siddique



# Adding & Removing Elements

JavaScript Allows You To Dynamically Add And Remove Elements From The DOM.

1. **CreateElement()**
2. **AppendChild()**
3. **RemoveChild()**
4. **ReplaceChild()**



Sayyed Siddique



# Example Of Adding & Removing Elements

```
// Create a new element
const newElement = document.createElement('div');
newElement.textContent = 'I am a new element';

// Add the new element to the body
document.body.appendChild(newElement);

// Remove an existing element
const parentElement = document.getElementById('parent');
const childElement = document.getElementById('child');
parentElement.removeChild(childElement);

// Replace an existing element
const anotherNewElement = document.createElement('p');
anotherNewElement.textContent = 'I am another new element';
parentElement.replaceChild(anotherNewElement, newElement);
```



Sayyed Siddique



# 5

# Event Listeners

Make Your Web Pages Interactive By Adding Event Listeners.

1. **AddEventListener()**
2. **RemoveEventListener()**



Sayyed Siddique

# Example Of Event Listeners

```
const button = document.getElementById('myButton');

// Add an event listener
button.addEventListener('click', () => {
  alert('Button clicked!');
});

// To remove the event listener
const handleClick = () => {
  alert('Button clicked!');
};
button.addEventListener('click', handleClick);
button.removeEventListener('click', handleClick);
```



Sayyed Siddique







# Traversing the DOM

Navigate Through The DOM Tree Using:

- 1. parentNode**
- 2..childNodes**
- 3. firstChild**
- 4. lastChild**
- 5. nextSibling**
- 6. previousSibling**



Sayyed Siddique



# Example Of Traversing the DOM

```
const child = document.getElementById('child');

// Get the parent element
const parent = child.parentNode;

// Get the first child of the parent
const firstChild = parent.firstChild;

// Get the last child of the parent
const lastChild = parent.lastChild;

// Get the next sibling of the child
const nextSibling = child.nextSibling;

// Get the previous sibling of the child
const previousSibling = child.previousSibling;
```



Sayyed Siddique



# 7 Manipulating CSS

You Can Change CSS Styles Directly  
Through JavaScript.

1. **Element.Style.Property**
2. **classList.Add()**
3. **classList.Remove()**
4. **classList.Toggle()**



Sayyed Siddique



# Example Of Manipulating CSS

```
const element = document.getElementById('myElement');  
  
// Change CSS style  
element.style.color = 'blue';  
  
// Add a class  
element.classList.add('newClass');  
  
// Remove a class  
element.classList.remove('oldClass');  
  
// Toggle a class  
element.classList.toggle('activeClass');
```



Sayyed Siddique



# Conclusion

**Mastering DOM Manipulation Is Essential For Modern Web Development. Start Experimenting With The Methods And Examples Shared To Create More Engaging And Dynamic Web Experiences!**



Sayyed Siddique





# Sayyed Siddique

**Thanks for reading**

**Follow for more**

“Please like; if you find value,  
consider following! ”