

Untitled

Chalo, main JavaScript ke DOM (Document Object Model) ke methods ko detail mein samjhaata hoon. Hum DOM ka use HTML document ke elements ko manipulate karne ke liye karte hain. Main tumhe ek ek method ke baare mein explain karunga, uska code example ke saath, aur use cases ke baare mein bataunga ki hum kab use karte hain.

1. getElementById()

Description:

Ye method ek HTML element ko uske ID se select karta hai.

Example:

javascript

Copy

Edit

// HTML

```
<button id="myButton">Click me!</button>
```

// JavaScript

```
let button =
```

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

Untitled

```
button.addEventListener('click',  
function() {  
    alert("Button clicked!");  
});
```

Use Case:

Agar tumhe kisi specific element ko ID se target karna ho, toh ye method useful hai.

2. getElementsByClassName()

Description:

Ye method ek ya zyada elements ko uske class name se select karta hai. Return value ek HTMLCollection hoti hai.

Example:

javascript

Copy

Edit

// HTML

```
<div class="box">Box 1</div>
```

```
<div class="box">Box 2</div>
```

// JavaScript

```
let boxes =
```

Untitled

```
document.getElementsByClassName('box');  
for (let i = 0; i < boxes.length; i++) {  
    boxes[i].style.backgroundColor =  
    'yellow';  
}
```

Use Case:

Agar tumhe ek ya zyada elements ko unke class ke basis pe target karna ho.

3. getElementsByTagName()

Description:

Ye method ek ya zyada elements ko unke tag name se select karta hai. Return value ek HTMLCollection hoti hai.

Example:

javascript

Copy

Edit

// HTML

<p>Paragraph 1</p>

<p>Paragraph 2</p>

// JavaScript

let paragraphs =

Untitled

```
document.getElementsByTagName('p');  
for (let i = 0; i < paragraphs.length;  
i++) {  
    paragraphs[i].style.color = 'blue';  
}
```

Use Case:

Jab tumhe kisi specific tag (e.g., <p>, <div>, etc.) ke saare elements ko target karna ho.

4. querySelector()

Description:

Ye method pehli matching element ko select karta hai jo provided CSS selector se match kare. Agar element milta hai, toh vo element return hota hai, agar nahi milta toh null return hota hai.

Example:

javascript

Copy

Edit

// HTML

<p class="text">This is a paragraph.</p>

Untitled

```
// JavaScript
let paragraph =
document.querySelector('.text');
paragraph.style.fontSize = '20px';
```

Use Case:

Jab tumhe ek specific element ko CSS selector ke through select karna ho.

5. querySelectorAll()

Description:

Ye method saare matching elements ko select karta hai jo provided CSS selector se match kare. Return value ek NodeList hoti hai.

Example:

javascript

Copy

Edit

// HTML

```
<div class="box">Box 1</div>
```

```
<div class="box">Box 2</div>
```

// JavaScript

Untitled

```
let boxes =  
document.querySelectorAll('.box');  
boxes.forEach(function(box) {  
    box.style.border = '1px solid  
black';  
});
```

Use Case:

Agar tumhe ek saath multiple elements ko ek selector ke zariye select karna ho.

6. createElement()

Description:

Ye method ek naya HTML element create karta hai.

Example:

javascript

Copy

Edit

```
// JavaScript
```

```
let newDiv =  
document.createElement('div');  
newDiv.textContent = "This is a new  
div!";  
document.body.appendChild(newDiv);
```

Untitled

Use Case:

Agar tumhe dynamically ek naya element create karna ho aur usse page pe add karna ho.

7. appendChild()

Description:

Ye method ek node ko parent element ke last mein append karta hai.

Example:

javascript

Copy

Edit

```
// JavaScript
let parentDiv =
document.getElementById('parent');
let newElement =
document.createElement('p');
newElement.textContent = 'This is a new
paragraph!';
parentDiv.appendChild(newElement);
```

Use Case:

Agar tumhe kisi element ko dusre element ke andar append karna ho, jaise ek div

Untitled

ke andar ek aur div ya paragraph.

8. removeChild()

Description:

Ye method ek child node ko remove karne ke liye use hota hai.

Example:

javascript

Copy

Edit

```
// JavaScript
let parentDiv =
document.getElementById('parent');
let childDiv =
document.getElementById('child');
parentDiv.removeChild(childDiv);
```

Use Case:

Agar tumhe kisi element ko uske parent se remove karna ho.

9. setAttribute()

Description:

Ye method kisi element ke attribute ko set karne ke liye use hota hai.

Untitled

Example:

javascript

Copy

Edit

// HTML

```

```

// JavaScript

```
let image =
```

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

```
image.setAttribute('src',
```

```
'new_image.jpg');
```

Use Case:

Agar tumhe kisi element ka attribute (jaise src, href, class, etc.) update karna ho.

10. getAttribute()

Description:

Ye method kisi element ke attribute ka value retrieve karta hai.

Example:

javascript

Untitled

Copy

Edit

// HTML

```
<a href="https://example.com"
id="myLink">Click me</a>
```

// JavaScript

```
let link =
document.getElementById('myLink');
let hrefValue =
link.getAttribute('href');
console.log(hrefValue); // Output:
https://example.com
```

Use Case:

Jab tumhe kisi element ka attribute value jaan na ho.

11. addEventListener()

Description:

Ye method ek event listener ko element pe attach karta hai, jo kisi specific event (e.g., click, mouseover, etc.) par trigger hota hai.

Example:

Untitled

javascript

Copy

Edit

// HTML

```
<button id="myButton">Click me!</button>
```

// JavaScript

```
let button =
```

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

```
button.addEventListener('click',
```

```
function() {
```

```
    alert('Button clicked!');
```

```
});
```

Use Case:

Jab tumhe kisi element par events ko handle karna ho, jaise button click, mouseover, etc.

12. removeEventListener()

Description:

Ye method ek event listener ko remove karne ke liye use hota hai.

Example:

javascript

Untitled

Copy

Edit

// HTML

```
<button id="myButton">Click me!</button>
```

// JavaScript

```
function handleClick() {  
    alert('Button clicked!');  
}
```

```
let button =  
document.getElementById('myButton');  
button.addEventListener('click',  
handleClick);
```

```
// Later, remove the event listener  
button.removeEventListener('click',  
handleClick);
```

Use Case:

Agar tumhe kisi element se event listener ko remove karna ho, jaise kisi click event ko disable karna.

Conclusion:

DOM methods ka use tab hota hai jab

Untitled

tumhe apne HTML document ko manipulate karna ho, elements ko select karna ho, unhe add ya remove karna ho, ya events handle karne ho. Har method ka apna specific use case hota hai, jo tumhare requirement pe depend karta hai.

Agar tumhe koi specific method ya example aur detail mein chahiye, toh batao!

You said:

bha in sabhi topics ke output chahiye code

ChatGPT said:

Chalo, main tumhe sabhi DOM methods ke outputs ke saath code example deta hoon,

Untitled

jisse tumhare liye har method ka practical use clear ho sake.

1. getElementById()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>getElementById
```

```
Example</title>
```

```
</head>
```

```
<body>
```

```
    <button id="myButton">Click
me!</button>
```

```
    <script>
```

```
        let button =
```

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

```
        button.addEventListener('click',
```

```
function() {
```

```
            alert("Button clicked!");
```

Untitled

```
        });  
    </script>  
</body>  
</html>
```

Output:

Click karne par ek alert box dikhayi
dega: "Button clicked!"

2. getElementsByClassName()

Code:

html

Copy

Edit

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>getElementsByClassName  
Example</title>  
</head>  
<body>  
    <div class="box">Box 1</div>  
    <div class="box">Box 2</div>  
  
    <script>
```

Untitled

```
let boxes =  
document.getElementsByClassName('box');  
for (let i = 0; i <  
boxes.length; i++) {  
  
boxes[i].style.backgroundColor =  
'yellow';  
}
```

</script>

</body>

</html>

Output:

Donon div elements ka background color
yellow ho jayega.

3. getElementsByTagName()

Code:

html

Copy

Edit

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>getElementsByTagName

Untitled

```
Example</title>
</head>
<body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>

    <script>
        let paragraphs =
document.getElementsByTagName('p');
        for (let i = 0; i <
paragraphs.length; i++) {
            paragraphs[i].style.color =
'blue';
        }
    </script>
</body>
</html>
```

Output:

Sabhi paragraphs ka text color blue ho jayega.

4. `querySelector()`

Code:

html

Copy

Untitled

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>querySelector Example</title>
</head>
<body>
  <p class="text">This is a
paragraph.</p>

  <script>
    let paragraph =
document.querySelector('.text');
    paragraph.style.fontSize =
'20px';
  </script>
</body>
</html>
```

Output:

Paragraph ka font size 20px ho jayega.

5. querySelectorAll()

Code:

html

Untitled

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>querySelectorAll
Example</title>
</head>
<body>
  <div class="box">Box 1</div>
  <div class="box">Box 2</div>

  <script>
    let boxes =
document.querySelectorAll('.box');
    boxes.forEach(function(box) {
      box.style.border = '1px
solid black';
    });
  </script>
</body>
</html>
```

Output:

Donon div elements ke around ek black

Untitled

border dikhayi dega.

6. createElement()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>createElement Example</title>
</head>
<body>
    <div id="parent"></div>

    <script>
        let newDiv =
document.createElement('div');
        newDiv.textContent = "This is a
new div!";

document.getElementById('parent').append
Child(newDiv);
    </script>
```

Untitled

```
</body>
```

```
</html>
```

Output:

Ek new div element jo text "This is a new div!" show karega, wo parent div ke andar append ho jayega.

7. appendChild()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>appendChild Example</title>
```

```
</head>
```

```
<body>
```

```
    <div id="parent"></div>
```

```
    <script>
```

```
        let parentDiv =
```

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

```
        let newElement =
```

page21

Untitled

```
document.createElement('p');  
    newElement.textContent = 'This  
is a new paragraph!';
```

```
parentDiv.appendChild(newElement);
```

```
    </script>
```

```
</body>
```

```
</html>
```

Output:

parent div ke andar ek naya paragraph
append hoga, jisme text "This is a new
paragraph!" dikhayi dega.

8. removeChild()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>removeChild Example</title>
```

```
</head>
```

```
<body>
```

Untitled

```
<div id="parent">
  <p id="child">This is a child
paragraph.</p>
</div>
```

```
<script>
  let parentDiv =
document.getElementById('parent');
  let childDiv =
document.getElementById('child');
  parentDiv.removeChild(childDiv);
</script>
</body>
</html>
```

Output:

Child paragraph remove ho jayega aur
page pe nahi dikhayi dega.

9. setAttribute()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
```

```
<html lang="en">
```

Untitled

```
<head>
  <meta charset="UTF-8">
  <title>setAttribute Example</title>
</head>
<body>
  

  <script>
    let image =
document.getElementById('image');
    image.setAttribute('src',
'new_image.jpg');
  </script>
</body>
</html>
```

Output:

Image ka src attribute update ho jayega
aur naya image new_image.jpg dikhayi
dega.

10. `getAttribute()`

Code:

html

Copy

Untitled

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>getAttribute Example</title>
</head>
<body>
    <a href="https://example.com"
id="myLink">Click me</a>

    <script>
        let link =
document.getElementById('myLink');
        let hrefValue =
link.getAttribute('href');
        alert(hrefValue);    // Output:
https://example.com
    </script>
</body>
</html>
```

Output:

href attribute ka value show hoga:
https://example.com.

Untitled

11. addEventListener()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>addEventListener
Example</title>
</head>
<body>
    <button id="myButton">Click
me!</button>

    <script>
        let button =
document.getElementById('myButton');
        button.addEventListener('click',
function() {
            alert('Button clicked!');
        });
    </script>
</body>
```

Untitled

</html>

Output:

Button pe click karne par ek alert box dikhayi dega: "Button clicked!"

12. removeEventListener()

Code:

html

Copy

Edit

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>removeEventListener
```

```
Example</title>
```

```
</head>
```

```
<body>
```

```
    <button id="myButton">Click  
me!</button>
```

```
    <script>
```

```
        function handleClick() {  
            alert('Button clicked!');  
        }
```

Untitled

```
let button =  
document.getElementById('myButton');  
    button.addEventListener('click',  
handleClick);  
  
    // Remove the event listener  
after 3 seconds  
    setTimeout(function() {  
  
button.removeEventListener('click',  
handleClick);  
    }, 3000);  
</script>  
</body>  
</html>
```

Output:

3 seconds ke baad button click pe koi alert nahi dikhayi dega, kyunki event listener remove ho gaya hoga.

Mujhe ummed hai ki ab tumhe sabhi DOM methods ke outputs samajh aa gaye honge. Agar tumhe koi aur explanation ya example chahiye ho, toh batao!

Untitled

You said:

aur ek bat iske topic hum kab use karte hai ye bhi batado aur solid exapple den abhai

ChatGPT said:

Chalo, main har ek DOM method ke use case ko ek solid example ke saath explain karta hoon, taaki tumhe pata chale ki har method ko kab use karna chahiye.

1. getElementById()

Use Case:

Hum is method ka use tab karte hain jab hume ek specific element ko ID ke through select karna ho. Yeh method

Untitled

unique element ko target karta hai, jaise ek specific button ya div.

Example:

Suppose tumhare paas ek form hai, aur tumhe us form ke "submit" button par click hone par kuch action perform karna hai. To, `getElementById()` ka use karenge.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Submit Form</title>
</head>
<body>
    <form id="myForm">
        <input type="text"
placeholder="Enter something">
        <button type="submit"
id="submitBtn">Submit</button>
```

Untitled

```
</form>
```

```
<script>
```

```
    let submitButton =  
document.getElementById('submitBtn');  
  
submitButton.addEventListener('click',  
function(e) {  
    e.preventDefault(); //  
Prevent form submission  
    alert('Form Submitted!');  
});
```

```
</script>
```

```
</body>
```

```
</html>
```

When to Use:

Jab ek unique element ko target karna ho (ID is unique).

Example: Form submission button ko handle karna.

2. getElementsByClassName()

Use Case:

Hum is method ka use tab karte hain jab

Untitled

hume multiple elements ko ek hi class name se target karna ho.

Example:

Suppose tumhare paas ek list hai jisme multiple items hain, aur tumhe un sabhi items ka background color change karna hai. To, `getElementsByClassName()` ka use karenge.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Change Background</title>
</head>
<body>
    <ul>
        <li class="item">Item 1</li>
        <li class="item">Item 2</li>
        <li class="item">Item 3</li>
    </ul>
```


Untitled

```
<script>
    let items =
document.getElementsByClassName('item');
    for (let i = 0; i <
items.length; i++) {

items[i].style.backgroundColor =
'lightblue';
    }
</script>
</body>
</html>
```

When to Use:

Jab tumhe ek hi class name ke saare elements ko manipulate karna ho (multiple elements).

Example: List items, images, buttons, etc. ko style karna.

3. getElementsByTagName()

Use Case:

Hum is method ka use tab karte hain jab hume kisi specific tag ke saare elements

Untitled

ko target karna ho. For example, sabhi paragraphs ya divs ko target karna.

Example:

Suppose tumhare paas ek webpage hai jisme multiple paragraphs hain, aur tumhe unka font size increase karna hai. To, `getElementsByTagName()` ka use karenge.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Increase Font Size</title>
</head>
<body>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
    <p>Paragraph 3</p>

    <script>
```

Untitled

```
let paragraphs =  
document.getElementsByTagName('p');  
for (let i = 0; i <  
paragraphs.length; i++) {  
    paragraphs[i].style.fontSize  
= '20px';  
}  
</script>  
</body>  
</html>
```

When to Use:

Jab tumhe kisi specific tag ke saare elements ko target karna ho.

Example: Sabhi paragraphs ko style karna.

4. querySelector()

Use Case:

Hum is method ka use tab karte hain jab hume first matching element ko select karna ho, jo CSS selector se match kare. Yeh ek powerful method hai jo classes, IDs, aur other selectors ko support karta hai.

Untitled

Example:

Suppose tumhare paas ek webpage hai jisme tumhe pehle element ko target karna hai, jisme .highlight class ho.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Highlight First
Element</title>
</head>
<body>
    <div class="highlight">This is the
highlighted div.</div>
    <div>This is a normal div.</div>

    <script>
        let highlightedDiv =
document.querySelector('.highlight');
```

Untitled

```
highlightedDiv.style.backgroundColor =  
'yellow';  
    </script>  
</body>  
</html>
```

When to Use:

Jab tumhe first matching element ko CSS selector se target karna ho.

Example: First element ko highlight karna.

5. querySelectorAll()

Use Case:

Hum is method ka use tab karte hain jab hume multiple elements ko select karna ho jo kisi specific CSS selector se match karte hain.

Example:

Suppose tumhare paas multiple buttons hain, aur tumhe unke upar click event listener apply karna hai. To, querySelectorAll() ka use karenge.

Untitled

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Multiple Buttons</title>
</head>
<body>
    <button class="btn">Button
1</button>
    <button class="btn">Button
2</button>
    <button class="btn">Button
3</button>

    <script>
        let buttons =
document.querySelectorAll('.btn');
        buttons.forEach(function(button)
{

button.addEventListener('click',
function() {
```

```
                                Untitled
                                alert('Button
clicked!');
                                });
                                });
                                </script>
</body>
</html>
```

When to Use:

Jab tumhe multiple elements ko target karna ho.

Example: Multiple buttons par click event apply karna.

6. createElement()

Use Case:

Hum is method ka use tab karte hain jab hume dynamically ek naya HTML element create karna ho aur page pe insert karna ho.

Example:

Suppose tumhare paas ek webpage hai jisme tumko dynamically ek new list item add karna hai jab user ek button click

Untitled

kare.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Add Item</title>
</head>
<body>
  <ul id="myList">
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
  <button id="addItemBtn">Add
Item</button>

  <script>

document.getElementById('addItemBtn').ad
dEventListener('click', function() {
    let newItem =
document.createElement('li');
```


Untitled

```
newItem.textContent = 'Item
3';

document.getElementById('myList').append
Child(newItem);
});
</script>
</body>
</html>
```

When to Use:

Jab tumhe dynamic elements add karne ho, jaise new list items ya divs.

Example: User interaction ke baad naye elements create karna.

7. appendChild()

Use Case:

Hum is method ka use tab karte hain jab hume newly created node ko kisi existing parent element ke andar add karna ho.

Example:

Suppose tumne ek div element create kiya aur usse ek existing container div ke

Untitled

andar append karna hai.

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Append New Div</title>
</head>
<body>
    <div id="container"></div>
    <button id="addDivBtn">Add
Div</button>

    <script>
```

```
document.getElementById('addDivBtn').add
EventListener('click', function() {
    let newDiv =
document.createElement('div');
    newDiv.textContent = 'This
is a new div!';
```

Untitled

```
document.getElementById('container').appendChild(newDiv);
    });
</script>
</body>
</html>
```

When to Use:

Jab tumhe kisi new element ko existing parent element ke andar add karna ho.

Example: Dynamic divs ko container ke andar add karna.

8. removeChild()

Use Case:

Hum is method ka use tab karte hain jab hume kisi child element ko parent element se remove karna ho.

Example:

Suppose tumhare paas ek todo list hai, aur tumhe kisi specific item ko remove karna hai jab user delete button click kare.

Untitled

html

Copy

Edit

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Remove Item</title>
</head>
<body>
  <ul id="todoList">
    <li id="todo1">Buy milk <button
onclick="removeItem('todo1')">Delete</bu
tton></li>
  </ul>

  <script>
    function removeItem(id) {
      let item =
document.getElementById(id);

item.parentElement.removeChild(item);
    }
  </script>
</body>
```

Untitled

</html>

When to Use:

Jab tumhe kisi child element ko parent se remove karna ho.

Example: Todo list item ko delete karna.

Is tarah se tum har DOM method ko specific scenarios me use kar sakte ho. Agar koi aur doubts ho ya koi aur example chahiye ho, toh zaroor poochna!