

JavaScript - DOM Methods

In JavaScript, **DOM methods** are used to perform a particular action on HTML elements. The DOM represents a HTML document or web page with a logical tree. In the tree, each branch ends in a **node**, and each node contains objects. DOM **methods** allow us programmatic access to the tree. Using the DOM methods, you can change the document's structure, style, or content.

For example, you can use DOM methods to access HTML elements using id, attribute, tag name, class name, etc., add events to the document or HTML elements, add new HTML elements to the DOM, etc.

Syntax

Following is the syntax to access and execute the DOM method in JavaScript –

```
window.document.methodName();
```

OR

```
document.methodName();
```

You may or may not use the 'window' object to access the document object's method.

Here, we have explained some HTML DOM methods with examples and covered other methods in the reference.

JavaScript document.getElementById() Method

The JavaScript getElementById() method of the document object is the most popular method to access the HTML elements using the id.

Syntax

Follow the syntax below to use the document.getElementById() method.

```
const ele = document.getElementById("id");
```

The getElementById() method takes an id of the HTML element as a parameter.

Example

In the below code, the id of the 'div' element is 'output'.

In JavaScript, we use the `document.getElementById()` method to access the div element using its id.

Also, we use the 'innerHTML' property of the element to add extra HTML to the 'div' element.

[Open Compiler](#)

```
<html>
<body>
  <button onclick = "accessEle()"> Access output and write </button>
  <p id = "output"> </p>
  <script>
    function accessEle() {
      document.getElementById("output").innerHTML =
        "Hello User! You have just clicked the button!";
    }
  </script>
</body>
</html>
```

JavaScript document.addEventListener() Method

The `addEventListener()` method is used to add the event to the document.

Syntax

Follow the syntax below to use the `addEventListener()` method with a document.

```
document.addEventListener('mouseover', function () {
  // Perform action on the document.
});
```

The `addEventListener()` method takes the event name as the first parameter and the callback function as a second parameter.

Example

In the below code, we added the 'mouseover' event to the document. Whenever you hover over the document, it will change the background color of the document body to red.

[Open Compiler](#)

```
<html>
<body>
  <h3>JavaScript - document.addEventListner() Method </h3>
  <p> Hover over here to change background color </p>
  <script>
    document.addEventListner('mouseover', function () {
      document.body.style.backgroundColor = 'red';
    });
  </script>
</body>
</html>
```

Learn **JavaScript** in-depth with real-world projects through our **JavaScript certification course**. Enroll and become a certified expert to boost your career.

JavaScript document.write() Method

The document.write() method adds the text or HTML to the document. It replaces the content of the document with the HTML passed as an argument of the method.

Syntax

Follow the syntax below to use the document.write() method.

```
document.write(HTML)
```

You can replace the 'HTML' argument with the text or HTML.

Example

In the below code, we execute the writeText() function when you click the button.

In the function, we use the document.write() method to add HTML to the web page. It replaces the HTML of the whole web page.

[Open Compiler](#)

```
<html>
<body>
  <button onclick = "writeText()"> Add HTML </button>
  <script>
```

```
function writeText() {  
    document.write("<p>Hello Users, Text is written using the document.wri  
    }  
</script>  
</body>  
</html>
```

List of DOM Methods

In the below table, we have listed all methods related to the HTML DOM. You can access all methods using the 'document' object.

Method	Description
<code>addEventListener()</code>	It is used to add an event listener to the document.
<code>adoptNode()</code>	It is used to adopt the node from the other documents.
<code>append()</code>	It appends the new node or HTML after the last child node of the document.
<code>caretPositionFromPoint()</code>	It returns the <code>caretPosition</code> object, containing the DOM node based on the coordinates passed as an argument.
<code>close()</code>	It closes the output stream opened using the <code>document.open()</code> method.
<code>createAttribute()</code>	It creates a new attribute node.
<code>createAttributeNS()</code>	It creates a new attribute node with the particular namespace URI.
<code>createComment()</code>	It creates a new comment node with a specific text message.
<code>createDocumentFragment()</code>	It creates a <code>DocumentFragment</code> node.
<code>createElement()</code>	It creates a new element node to insert into the web page.
<code>createElementNS()</code>	It is used to create a new element node with a particular namespace URI.
<code>createEvent()</code>	It creates a new event node.
<code>createTextNode()</code>	It creates a new text node.

<code>elementFromPoint()</code>	It accesses the element from the specified coordinates.
<code>elementsFromPoint()</code>	It returns the array of elements that are at the specified coordinates.
<code>getAnimations()</code>	It returns the array of all animations applied on the document.
<code>getElementById()</code>	It accesses the HTML element using the id.
<code>getElementsByClassName()</code>	It accesses the HTML element using the class name.
<code>getElementsByName()</code>	It accesses the HTML element using the name.
<code>getElementsByTagName()</code>	It accesses the HTML element using the tag name.
<code>hasFocus()</code>	It returns the boolean value based on whether any element or document itself is in the focus.
<code>importNode()</code>	It is used to import the node from another document.
<code>normalize()</code>	It removes the text nodes, which are empty, and joins other nodes.
<code>open()</code>	It is used to open a new output stream.
<code>prepend()</code>	It is used to insert the particular node before all nodes.
<code>querySelector()</code>	It is used to select the first element that matches the css selector passed as an argument.
<code>querySelectorAll()</code>	It returns the nodelist of the HTML elements, which matches the multiple CSS selectors.
<code>removeEventListener()</code>	It is used to remove the event listener from the document.
<code>replaceChildren()</code>	It replaces the children nodes of the document.
<code>write()</code>	It is used to write text, HTML, etc., into the document.
<code>writeln()</code>	It is similar to the <code>write()</code> method but writes each statement in the new line.