HTML DOM

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

When an HTML document is loaded into a web browser, it becomes a **document object**.

In the HTML DOM (Document Object Model), everything is a **node**:

- The document itself is a document node
- All HTML elements are element nodes
- All HTML attributes are attribute nodes
- Text inside HTML elements are text nodes
- Comments are comment nodes

Tip: The document is a part of the Window object and can be accessed as window.document.

A **property** is a value that you can get or set (like changing the content of an HTML element).

A **method** is an action you can do (like add or deleting an HTML element).

DOM Element Object

Finding HTML Elements

By ID

document.getElementById("intro");

By Tag Name

document.getElementsById("p");

By Class Name

document.getElementsByClassName("intro");

By CSS selectors

document.querySelector(".example").style.backgroundColor = "red";

Note: The querySelector() method only returns the first element that matches the specified selectors. To return all the matches, use the [querySelectorAll()](#) method instead.

```
var x = document.querySelectorAll(".example");

    x[0].style.backgroundColor = "red";
```

Note: Index position is must

Set background color for all p elements

```
var x = document.querySelectorAll("p");

    var i;

    for (i = 0; i < x.length; i++) {

        x[i].style.backgroundColor = "red";
```


Different ways to use queryselector

```
document.querySelectorAll("div > p");

document.querySelectorAll("h2, div, span");
```


Note: There are two ways to access a node at the specified index in a node list:

```
document.body.childNodes.item(0);    // The first child node of <body>

document.body.childNodes[0];          // The first child node of <body>
```


Useful examples:

```
<div id="myDIV">

  <p>First p element in div.</p>

  <p>Another p element in div.</p>

  <p>A third p element in div.</p>

</div>
```

Ex1:

```
var div = document.getElementById("myDIV");
div.getElementsByTagName("P")[0].innerHTML = "Paragraph changed";
```

Ex2:

```
var div = document.getElementById("myDIV");
var nodelist = div.getElementsByClassName("child");
```

```
var i;
for (i = 0; i < nodelist.length; i++) {
    nodelist[i].style.backgroundColor = "red";
}
```

Changing HTML Elements

1. element.innerHTML

Return:

```
var x = document.getElementById("myP").innerHTML;

    document.getElementById("demo").innerHTML = x;
```

Set

```
document.getElementById("myP").innerHTML = "Hello Dolly.";
```

other ex:

```
document.getElementById("myAnchor").innerHTML = "W3Schools";

document.getElementById("myAnchor").href =
"https://www.w3schools.com";

    document.getElementById("myAnchor").target = "_blank";
```

2. *element.attribute = new value*

Ex1: It wil change the id

```
var x = document.getElementById('DIV1').id = "div2";

var x = document.getElementById('DIV1').src = "home.jpg"; //
it will replace current image to home
```

3.*element*.style.*property = new style*

ex1:

```
document.getElementById("myH1").style.color = "red";
```

ex2: Multiple properties

```
document.getElementById("myP").style.cssText = "background-
color:pink;font-size:55px;border:2px dashed green;color:white;"


var x = document.getElementsByTagName('H2')[0];

x.setAttribute("style", "color:red;font-sixe:18px;");
```

Other examples:

For getting inline styelesheet

```
var x = document.getElementsByTagName("STYLE")[0];
    document.getElementById("demo").innerHTML = x.innerHTML;
```

```
document.getElementById("More Text").setAttribute("style", "font-size:50px;color:red;");
```

Attributes

```
var x = document.getElementById("myBtn").attributes.length;
```

```
var x = document.getElementById("myBtn").attributes[1].name; //Prints name
```

```
var x = document.getElementById("myBtn").attributes[1].name; //Prints
value
```

getAttribute

element.getAttribute(*attributename*)

```
var x = document.getElementsByTagName("H1")[0].getAttribute("class");
//returs class name of h1
```

```
var x = document.getElementById("myAnchor").getAttribute("target");
```

//returs _blank

createAttribute

```
var h1 = document.getElementsByTagName("H1")[0];   // Get the first <h1>
element in the document
var att = document.createAttribute("class");       // Create a "class"
attribute
att.value = "democlass";                           // Set the value of the
class attribute
h1.setAttributeNode(att);                          // Add the class
attribute to <h1>
```

The setAttributeNode() method adds the specified attribute node to an element.

Tip: Use the removeAttributeNode() method to remove an attribute node from an element.

removeAttribute

ex1: document.getElementsByTagName("H1")[0].removeAttribute("class");

ex2: document.getElementById("myAnchor").removeAttribute("href");

setAttribute

ex1:
document.getElementById("myAnchor").setAttribute("href", "https://www.w3schools.com");

ex2:
document.getElementsByTagName("H1")[0].setAttribute("class", "democlass");

ex3: document.getElementById("More Text").setAttribute("style", "font-size:50px;color:red;");

Note:

1. **Tip:** Use setAttribute() to add a new attribute or change the value of an existing attribute on an element.
2. The hasAttribute() method returns true if the specified attribute exists, otherwise it returns false.

Adding and Deleting Elements

CreateElement();

removeChild();

appendChild();

replaceChild();

```
var btn = document.createElement("BUTTON");      // Create a <button>
element
var t = document.createTextNode("CLICK ME");     // Create a text node
btn.appendChild(t);                              // Append the text to
<button>
document.body.appendChild(btn);                  // Append <button> to
<body>
```

or

document.getElementById("myDIV").appendChild(para);

Tip: Use the createTextNode() method to create a text node.

Tip: After the element is created, use the _element_.appendChild() or _element_.insertBefore() method to insert it to the document.

node.insertBefore(_newnode, existingnode_)

removeChild()

ex1: `<ul id="myList"><li>Coffee</li><li>Tea</li><li>Milk</li></ul>`

```
var list = document.getElementById("myList");

list.removeChild(list.childNodes[0]); //remove ul firstchild node(index 0)
```

appendChild()

```
EX1: var node = document.createElement("LI");

     var textnode = document.createTextNode("Water");

     node.appendChild(textnode);

     document.getElementById("myList").appendChild(node);
```

```
EX2: var node = document.getElementById("myList2").lastChild;

     document.getElementById("myList1").appendChild(node);
```

replaceChild()

node.replaceChild(_newnode, oldnode_)

ex1: `<ul id="myList"><li>Coffee</li><li>Tea</li><li>Milk</li></ul>`

```
     var textnode = document.createTextNode("Water");

   var item = document.getElementById("myList").childNodes[0];

   item.replaceChild(textnode, item.childNodes[0]);
```

firstChild()&lastChild()

ex1: `<ul id="myList"><li>Coffee</li><li>Tea</li></ul>`

```
var list = document.getElementById("myList").firstChild.innerHTML;

     document.getElementById("demo").innerHTML = list;//coffee
```

ex2:

```html
<select id="mySelect"
size="4"><option>Audi</option><option>BMW</option><option>Saab</option><option>Volvo</option></select>
```

```javascript
function myFunction() {

    var x = document.getElementById("mySelect").firstChild.text;

    document.getElementById("demo").innerHTML = x;  // Audi

}
```

Imp programms

```javascript
function reverseString(str) {

    var newString = "";

    for (var i = str.length - 1; i >= 0; i--) {

        newString += str[i];

    }

    return newString;

}
reverseString('hello');
```

output: "olleh"