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
```

2. *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;");
```