HTML Element Selectors In JavaScript | JavaScript Tutorial In Hindi #14

If you are reading this tutorial, then you must be familiar with DOM (Document Object Model). If not, then check ***tutorial#12*** to get the concept of DOM. Today's tutorial is about DOM selector, so one must have a fair idea about DOM before moving forward. As we know that the JavaScript is most commonly used to get or modify the content or value of the HTML elements. However, before we perform any action like change style or add animation, our first task is to need to find or select the target HTML element. In today's tutorial, we will see some of the common ways of selecting the elements on a page using different methods.

Gets	**Selector Syntax**	**Method**
ID	#name	getElementById()
Class	.name	getElementsByClassName()
Tag	name	getElementsByTagName()
Selector (single)		querySelector()
Selector (all)		querySelectorAll()

DOM Selectors is used to selecting HTML elements within a document using JavaScript. There are two types of a selector, i.e., single element selector and multiple element selector.

Single Element Selector:-

It is used to select single HTML elements within a document. Following are some single element selectors:

Accessing Elements by ID:-

To access the single element in the DOM is by its unique ID. We can get or modify an element by ID with the **getElementById()** method of the document object. It will return null if no elements with the specified ID exists.

An ID should be unique within a page. However, if more than one element with the specified ID exists, the getElementById() method returns the first element in the source code.

```
document.getElementById(elementID);
```

In order to be accessed by ID, the HTML element must have an id attribute. We have a h1 element with an ID of demo.

```
<h1 id="demo">Heading 1</h1>
var x = document.getElementById("demo");
x.style.color = "red";
x.className;
x.childNodes;
x.parentNode;
x.innerText = 'JavaScript Tutorial';
x.innerHTML = '<b>Harry is a good boy</b>';
```

Explanation of above Example:-

- **className:** The className property sets or returns the class name of an element
- **childNodes:** The childNodes property returns a collection of a node's child nodes.
- **parentNode:** The parentNode property returns the parent node of the specified node.
- **innerText:** The innerText property sets or returns the text content of the specified node.
- **innerHTML:** The innerHTML property sets or returns the HTML content of an element.

querySelector:-

It will return the first element that matches the specified CSS selector in the document. The querySelector() method only returns the first element that matches the specified selectors.

```
document.querySelector('#myclass');
document.querySelector('.myclass');
document.querySelector('li');
```

The above statement will return the first element that matches the CSS selector #myclass, .myclass and li. We can use all kinds of CSS selectors within the querySelector method.

Multiple Element Selector:-

It is used to select multiple HTML elements within a document. There are three ways in which we can select elements in a DOM using multiple element selectors.

- querySelectorAll()
- getElementsByTagName()

- getElementsByClassName()

querySelectorAll:-

It will return a list of the document's elements that match the specified group of selectors. The querySelectorAll() method returns all elements in the document as a static NodeList object. The NodeList object represents a collection of nodes. The nodes can be accessed by index numbers. The index starts at 0.

```
document.querySelectorAll('.heading')
```

It will return a list of all elements that matches the specified CSS selector.

```
document.querySelectorAll("p.animation ");
```

It will return all <p> elements in the document with class="animation"

Accessing Elements by Tag Name:-

We can also select HTML elements with their tag name by using getElementsByTagName() method. This method returns an array-like object of all child elements with the given tag name. The syntax is:
document.getElementsByTagName(tagname);
Here is an example:

```
<article> My article 1</article>
<article> My article 2</article>
```

We can modify every tag in the document with a for and forEach loop. In this example, we are using for loop

```
let mytag = document.getElementsByTagName('article');
for (let i = 0; i < mytag.length; i++) {
mytag[i].style.border = '1px solid blue';
}
```

Accessing Elements by Class:-

The class attribute is used to access one or more specific elements in the DOM. The ID should be unique, whereas one or more elements can have the same class. Using ***getElementsByClassName()*** method., we can get all the elements with a given class name. Here is the syntax

```
document.getElementsByClassName(class_name);
```

We can access more than one element, and in our example we have two elements with a demo class. For Example:

```
<div class="demo"> class 1</div>
<div class="demo"> class 2</div>
```

To find how many elements with class="demo" there are in the document, we use the length property of the HTMLCollection object. Here is an example

```
document.getElementsByClassName("demo").length;
```

Website.html code as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <div class="container">
        <h1 id="heading"> Welcome to Code With Harry</h1>
        <div id="myfirst" class="child red good" id="first">child 1

            <ul class="this">
                <li class="childul">this</li>
                <li class="childul">is</li>
                <li class="childul">a</li>
                <li class="childul">list </li>
                <li class="childul">of my dreams</li>
            </ul>
        </div>
        <div class="child">child 2</div>
        <div class="child red">child 3</div>
        <div class="child">child 4</div>
        <form action="none.html" method="post">
            <a href="//codewithharry.com">Go to Code With Harry</a>
```

```
            <br>
            <br>
            Search this website: <input type="text" name="Hello" id="">
            <input type="button" value="submit">
        </form>
    </div>
    <br>
    <div class="no">this is a dummy div1</div>
    <div class="no">this is a dummy div2</div>
    <div class="no">this is a dummy div3</div>
</body>
<!-- <script src="js/tut12.js"></script> -->
<!-- <script src="js/tut14.js"></script> -->
<script src="js/tut15.js"></script>
</html>
```

JavaScript code as described/written in the video

```
console.log('Welcome to tutorial 14');
/*
element selectors:
1. Single element selector
2. Multi element selector

*/

// 1. Single element selector
let element = document.getElementById('myfirst');
// element = element.className;
// element = element.childNodes;
// element = element.parentNode;
element.style.color = 'red';
element.innerText = 'Harry is a good boy';
element.innerHTML = '<b>Harry is a good boy</b>';
// console.log(element.innerText);


let sel = document.querySelector('#myfirst');
```

```
sel = document.querySelector('.child');
sel = document.querySelector('div');
sel.style.color = 'green';
console.log(sel)

// 2. Multi element selector
let elems = document.getElementsByClassName('child');
elems = document.getElementsByClassName('container');
elems = document.getElementsByTagName('div');
console.log(elems)

for (let index = 0; index < elems.length; index++) {
    const element = elems[index];
    console.log(element);
   element.style.color = 'blue';
}
// Array.from(elems).forEach(element => {
//     console.log(element);
//    element.style.color = 'blue';
// });
// console.log(elems[0].getElementsByClassName('child'))
```

Previous Next

CodeWithHarry