https://www.w3schools.com/js/js_htmldom_document.asp

JavaScript HTML DOM Document

- The HTML DOM document object is the owner of all other objects in your web page.

The HTML DOM Document Object

- The document object represents your web page.
- If you want to access any element in an HTML page, you always start with accessing the document object.
- Below are some examples of how you can use the document object to access and manipulate HTML.

Finding HTML Elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

Changing HTML Elements

Property	Description
element.innerHTML = *new html content*	Change the inner HTML of an element

element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

Adding and Deleting Elements

Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new, old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

Adding Events Handlers

Method

document.getElementById(*id*).onclick = function(){*code*}

ความหมาย : Adding event handler code to an onclick event

```
<h1 onclick="this.innerHTML='Ooops!'">Click on this text!</h1>
```

```
<h1 onclick="changeText(this)">Click on this text!</h1>
        <script>
                function changeText(id) {
                        id.innerHTML = "Ooops!";
                }
        </script>
```

```
<div onmousedown="mDown(this)" onmouseup="mUp(this)"
style="background-color:#D94A38;">Click Me</div>


<script>
        function mDown(obj) {
                obj.style.backgroundColor = "#1ec5e5";
                obj.innerHTML = "Release Me";
        }
```

```
        function mUp(obj) {
                obj.style.backgroundColor="#D94A38";
                obj.innerHTML="Thank You";
        }
</script>
```

Finding HTML Objects

The first HTML DOM Level 1 (1998), defined 11 HTML objects, object collections, and properties. These are still valid in HTML5.

Later, in HTML DOM Level 3, more objects, collections, and properties were added.

Property	Description	DOM

document.anchors	Returns all <a> elements that have a name attribute	1
document.applets	Returns all <applet> elements (Deprecated in HTML5)	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3
document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1

document.domConfig	Obsolete. Returns the DOM configuration	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all <img> elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements that have a href attribute	1
document.readyState	Returns the (loading) status of the document	3

document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

= = = = = + = = = = = + = = = = = + = = = = =

```
<script>
	var person = {
		firstName: "John",
		lastName : "Doe",
	};
	document.getElementById("demo").innerHTML = person.fullName();
</script>
```

= = = = = + = = = = = + = = = = = + = = = = =

```
<script>
// Create an object:
        var person = {
                firstName: "John",
                lastName : "Doe",
                language : "EN" ,
                id     : 5566,
          fullName : function() {
                return this.firstName + " " + this.lastName;
          }

};
// Change a property
        Object.defineProperty(person, "firstName", {value:"NO"})
        document.getElementById("demo").innerHTML = person.firstName;
</script>
```

= = = = = + = = = = = + = = = = = + = = = = =

JavaScript Object Constructors

```
// กำหนดให้ element ที่มี id = demo แสดงผลจาก innerHTML
// ตาม ตัวแปรอาร์เรย์   person.firstname และ person.age
<p id="demo"></p>
<script>
        var person = {
                firstname:"John",
                lastname:"Doe",
                age:50,
                eyecolor:"blue"
        };

//delete person.age;
        document.getElementById("demo").innerHTML =
        person.firstname + " is " + person.age + " years old.";
</script>
```

= = = = = + = = = = = + = = = = = + = = = = =

```
<p id="demo"></p>
<script>

// Constructor function for Person objects
        function Person(first, last, age, eye) {
                this.firstName = first;
                this.lastName = last;
                this.age = age;
                this.eyeColor = eye;
        }
// Create a Person object
        var myFather = new Person("John", "Doe", 50, "blue");
// Display age
        document.getElementById("demo").innerHTML =m"My father is " + myFather.age + ".";

</script>
```