

JavaScript – DOM – Objeto Style

Artículos web:

[Cómo modificar atributos, clases y estilos en el DOM | DigitalOcean](#)

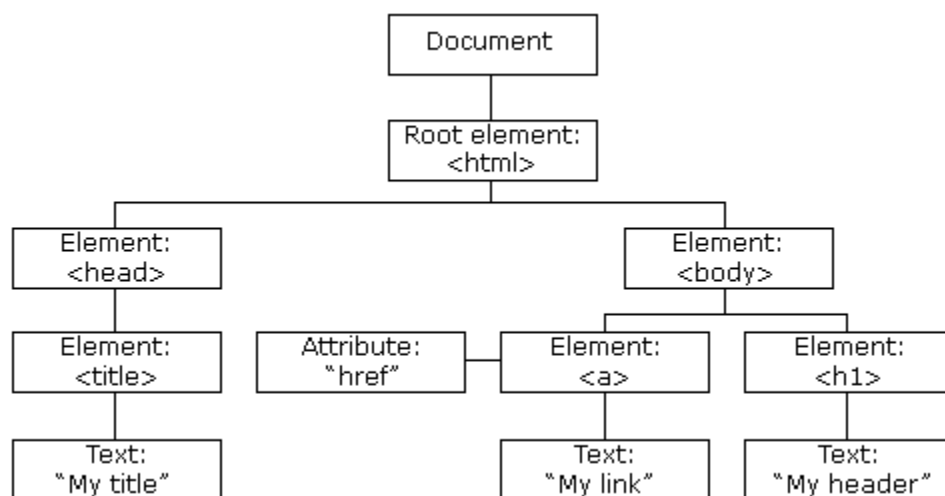
Document Object Model (DOM)

Los navegadores web transforman automáticamente todas las páginas web en una estructura más eficiente de manipular. Esta transformación la realizan todos los navegadores de forma automática y nos permite utilizar las herramientas de DOM de forma muy sencilla. El HTML DOM se construye como un “árbol de nodos”

<https://dom.spec.whatwg.org/>

[JavaScript and HTML DOM Reference](#)

[5.1. Árbol de nodos \(Introducción a JavaScript\)](#)



Tipos de nodos

Los principales tipos de nodos son:

- Document, nodo raíz del que derivan todos los demás nodos del árbol.
- Element, representa cada una de las etiquetas. Se trata del único nodo que puede contener atributos y el único del que pueden derivar otros nodos.
- Attr, se define un nodo de este tipo para representar cada uno de los atributos de las etiquetas, es decir, uno por cada par atributo="valor".
- Text, nodo que contiene el texto encerrado por una etiqueta.

Referencia a los elementos HTML

La referencia a los elementos HTML del documento se puede realizar:

- Por el id.
- Por el nombre de la etiqueta (tag).
- Por el nombre de la clase (class).
- Por el name.

Por el identificador: getElementById()

Si tenemos un elemento cuyo id="intro", para referenciarlo en JavaScript:

```
const x = document.getElementById("intro");
```

Si el elemento existe, se devolverá el elemento como un objeto (en la constante x). Si no lo encuentra, devuelve null.

Por el nombre de la etiqueta: getElementsByTagName()

Ejemplo:

```
let x = document.getElementById("main");  
let y = x.getElementsByTagName("p");
```

Devuelve un HTMLCollection (de tipo array)
<https://developer.mozilla.org/es/docs/Web/API/HTMLCollection>

Por el atributo class: getElementsByClassName

Este selector nos permite encontrar elementos en el documento por medio del valor de su atributo class.

Por el atributo name: document.getElementsByName(name)

Obtiene los elementos con ese nombre.

Ejemplo:

```
<script>  
  let x = document.getElementsByName("x");  
  alert(x.length);  
</script>
```

Ejemplo:

```
function comprueba() {  
  let x = document.getElementsByName("campo");  
  let cad=""  
  for (let i=0; i < x.length; i++) {  
    cad+=x[i].value  
    if (x[i].checked) {  
      valor=x[i].value
```

```

    }
}
alert(x.length+cad+"=>" + valor);
}

<form name="formulario">
  <label>a</label>
  <input type="radio" name="campo" checked="checked" value="a"><br/>

  <label>b</label>
  <input type="radio" name="campo" value="b"><br/>

  <label>c</label>
  <input type="radio" name="campo" value="c"><br />

  <input type="button" onclick="comprueba()">
</form>

```

http://www.w3schools.com/jsref/dom_obj_all.asp

Con las funciones `querySelector()` y `querySelectorAll()`.

<https://developer.mozilla.org/es/docs/Web/API/Document/querySelector>
<https://developer.mozilla.org/es/docs/Web/API/Document/querySelectorAll>
<https://developer.mozilla.org/es/docs/Web/API/Element/querySelector>
<https://developer.mozilla.org/es/docs/Web/API/Element/querySelectorAll>
http://www.w3schools.com/jsref/met_document_queryselector.asp
http://www.w3schools.com/jsref/met_document_queryselectorall.asp
[Selectors API](#)

Tened en cuenta que estas funciones están disponibles tanto para `document` y para `element`:

- `document.querySelector()`
- `document.querySelector()`
- `element.querySelector()`
- `element.querySelector()`

Repaso selectores CSS para usar con `querySelector`.

https://www.w3schools.com/cssref/css_selectors.asp

La función **`querySelector()`** Devuelve el primer elemento del documento o elemento (utilizando un [recorrido primero en profundidad pre ordenado](#) de los nodos del documento) que coincida con el grupo especificado de selectores.

La función **`querySelectorAll()`** devuelve una lista de los elementos dentro del documento o del elemento que coinciden con el grupo especificado de selectores. El objeto devuelto es un `NodeList`.

Modificar el contenido de un elemento HTML

Se utiliza la propiedad **innerHTML**: `document.getElementById(id).innerHTML=new HTML`

Ejemplo:

```
<p id="p1">Hola</p>
<script>
    document.getElementById("p1").innerHTML="Texto nuevo";
</script>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>
    <h1 id="header">Cabecera</h1>
<script>
    let element      = document.getElementById("header");
    element.innerHTML = "Nueva cabecera ";
</script>

</body>
</html>
```

Cambiar el atributo de un elemento HTML

`document.getElementById(id).atributo=nuevo valor`

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>
    <img id="image" src="">
    <script>
        document.getElementById("image").src="landscape.jpg";
    </script>
</body>
</html>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title></title>
    <script type="text/javascript" >
        function cambia() {
            document.getElementById("p1").innerHTML="Imagen:";
            document.getElementById("image").src="tin.PNG";
        }
    </script>
</head>
<body>
```

```

        <p id="p1">No hay imagen</p>
        <img id="image" src=""><br/>
        <button onclick="cambia()">Cambiar</button>
    </body>
</html>

```

Cambiar el estilo

`document.getElementById(id).style.propiedad=nuevo estilo`

Si queremos cambiar la clase a la que pertenece ese elemento:

`document.getElementById("nombreId").className = "nombreDeClase";`

```

function cambia_clase(id_del_objeto, nueva_clase){
    let objeto      = getElementById(id_del_objeto);
    objeto.className = nueva_clase;
}

```

Ejemplo:

```

let color      = prompt("introduce un color");
let elemento   = getElementById("parra");
elemento.className = color;

```

(Donde color sería el nombre de una clase y los colores serían cada uno una clase)

Ejemplo:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title></title>
    <script type="text/javascript" >
        function cambia() {
            document.getElementById("p2").style.color="yellow";
        }
    </script>
</body>
    <p id="p2">Hola</p>
    <script>
        document.getElementById("p2").style.color="blue";
    </script>
    <button onclick="cambia()">Cambiar</button>
</body>
</html>

```

Ejemplo:

```

<!DOCTYPE html>
<html>
<body>
    <h1 id="id1">My Heading 1</h1>
    <button type="button"
    onclick="document.getElementById('id1').style.color='red'">
    Click Me!</button>

```

```
</body>
</html>
```

Objeto Style

<https://developer.mozilla.org/es/docs/DOM/Element.style>

Devuelve un objeto que representa el atributo style del elemento.

Ejemplo:

```
let div = document.getElementById("div1");
div.style.marginTop = ".25in";
```

La propiedad style tiene la misma (y más alta) prioridad en la cascada CSS que las declaraciones *in-line* hechas mediante el atributo style, resulta muy útil para establecer el estilo en un elemento específico. Pero no resulta útil para aprender acerca del estilo original de un elemento, ya que representa sólo la declaración CSS en el atributo style *in-line* y no aquellos atributos que vienen de alguna otra parte, como las declaraciones en la sección <head> o en hojas de estilo externas.

<https://developer.mozilla.org/es/docs/DOM/Window.getComputedStyle>

Generalmente es mejor usar la propiedad style que usar **elemento.setAttribute('style', '...')** ya que el uso de la propiedad style no reemplazará otras propiedades CSS que puedan especificarse en el atributo style.

Los estilos *no* pueden establecerse asignando una cadena a la propiedad (solo lectura) style, como en elemento.style = "color: blue;". Esto es porque el atributo style devuelve un objeto del tipo CSSStyleDeclaration. En su lugar, puedes establecer las propiedades como:

```
// Asignación directa
elemento.style.color = "blue";
```

```
// Asignación Indirecta
```

```
let style = elemento.style;
```

```
style.color = "blue";
```

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo 1</title>
  <script>
    window.onload=setTimeout(function() {
      document.getElementById("ej1").setAttribute('style',
        "font-size: 1.5em;");
      document.getElementById("ej2").style.fontSize="1.5em";
    },3000);
  </script>
</head>
<body>
```

```
        <div id="ej1" style="color: red;">Escrito en color rojo</div>
        <div id="ej2" style="color: red;">Escrito en color rojo</div>
</body>
</html>
```

Ejemplos:

<http://www.w3schools.com/cssref/default.asp>

http://www.w3schools.com/jsref/dom_obj_style.asp

<https://developer.mozilla.org/es/docs/DOM/CSS>