



Ministerio de  
**TRABAJO, EMPLEO  
Y SEGURIDAD SOCIAL**



*Paraguay  
de la gente*

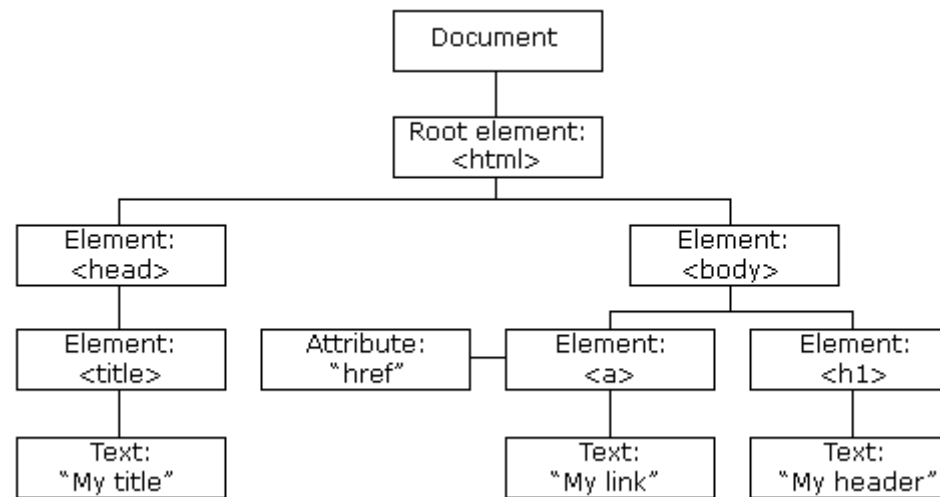
# DOM-Javascript

---

# Qué es el DOM?

---

El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. En resumen, es la representación de la página web en la memoria del navegador, a la que podemos acceder a través de JavaScript. El DOM es un árbol donde cada nodo es un objeto con todas sus propiedades y métodos que nos permiten modificarlo. Estas son algunas funciones que nos permiten acceder y modificar los elementos del DOM.



# Acceso a elementos del DOM

---

```
// Obtiene un elemento por id
document.getElementById('algunid');

// Obtiene una lista con los elementos que tienen esa clase
document.getElementsByClassName('algunclass');

// Obtiene una HTMLCollection con todos los elementos 'li'
document.getElementsByTagName('LI');

// Devuelve el primer elemento del documento que cumpla la selección (la notación es como en CSS)
document.querySelector('.algunclass');

// Devuelve una lista de elementos que cumplen con la selección (notación como en CSS)
. document.querySelectorAll('div.note, div.alert');
```

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var miElemento = document.getElementById("byId"); document.getElementById("demold").innerHTML = "El texto obtenido de byId es " + miElemento.innerHTML;  var x = document.getElementsByTagName("p"); document.getElementById("demoTag").innerHTML = 'El texto del tercer parrafo es paragraph (index 2) es: ' + x[2].innerHTML;  var y = document.getElementsByTagName("byClassName"); document.getElementById("demoClassName").innerHT ML = 'El texto del quinto parrafo de (index 0) class="byClassName" es: ' + y[0].innerHTML;  var z = document.querySelectorAll("p.byQuerySelector"); document.getElementById("demoQuerySelector").inne rHTML = 'El texto del septimo parrafo de (index 0) class="byQuerySelector" es: ' + z[0].innerHTML;</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;p id="byId"&gt;Demostracion getElementById!&lt;/p&gt;  &lt;p id="demold"&gt;&lt;/p&gt;  &lt;p id="byTag"&gt;Demostracion getElementByTag!&lt;/p&gt;  &lt;p id="demoTag"&gt;&lt;/p&gt;  &lt;p class="byClassName"&gt;Demostracion getElementByClassName&lt;/p&gt;  &lt;p id="demoClassName"&gt;&lt;/p&gt;  &lt;p class="byQuerySelector"&gt;Demostracion querySelectorAll &lt;/p&gt;  &lt;p id="demoQuerySelector"&gt;&lt;/p&gt; &lt;script src=codigo.js&gt;&lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<div>Demostracion getElementById! El texto obtenido de byId es Demostracion getElementById!  Demostracion getElementByTag! El texto del tercer parrafo es paragraph (index 2) es: Demostracion getElementByTag!  Demostracion getElementByClassName El texto del quinto parrafo de (index 0) class="byClassName" es: Demostracion getElementByClassName  Demostracion querySelectorAll El texto del septimo parrafo de (index 0) class="byQuerySelector" es: Demostracion querySelectorAll</div>

# Acceder a hijos/padres de un elemento

---

```
// Obtener los hijos de un elemento
var elem = document.getElementById('someid');
var hijos = elem.childNodes;

// Su nodo padre
var padre = document. elem.parentNode;
```

# Crear nuevos elementos en el DOM

---

```
// Para crear elementos llamamos a createElement con el nombre del elemento
var nuevoH1 = document.createElement('h1');
var nuevoParrafo = document.createElement('p');

// Crear nodos de texto para un elemento
var textoH1 = document.createTextNode('Hola mundo!');
var textoParrafo = document.createTextNode('lorem ipsum...');

// Añadir el texto a los elementos
nuevoH1.appendChild(textoH1);
nuevoParrafo.appendChild(textoParrafo);

// también podemos asignar directamente el valor a la propiedad innerHTML
nuevoH1.innerHTML = textoH1
nuevoParrafo.innerHTML = textoParrafo
// los elementos estarían listos para añadirlos al DOM, ahora mismo solo existen en memoria,
//pero no serán visibles hasta que no los añadamos a un elemento del DOM
```

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var para = document.createElement("p"); var node = document.createTextNode("Este es nuevo."); para.appendChild(node); var element = document.getElementById("div1"); element.appendChild(para);</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;div id="div1"&gt; &lt;p id="p1"&gt;Este es un parrafo.&lt;/p&gt; &lt;p id="p2"&gt;Este es otro parrafo.&lt;/p&gt; &lt;/div&gt;  &lt;script src=codigo.js&gt;&lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<div>Este es un parrafo.</div> <div>Este es otro parrafo.</div> <div>Este es nuevo.</div>

# Añadir elementos al DOM

---

```
// seleccionamos un elemento
var cabecera = document.getElementById('cabecera');

// Añadir elementos hijos a un elemento
cabecera.appendChild(nuevoH1);
cabecera.appendChild(nuevoParrafo);

// También podemos añadir elementos ANTES del elemento seleccionado

// Tomamos el padre
var padre = cabecera.parentNode;

// Insertamos el h1 antes de la cabecera
padre.insertBefore(nuevoH1, cabecera);
```



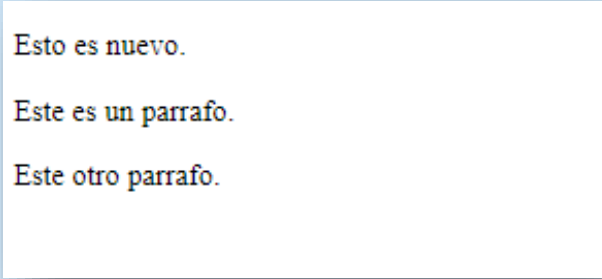
También podemos añadir directamente un trozo de HTML antes o después de un elemento del DOM, supongamos que tenemos estos elementos en la página:

```
<div id='box1'>
    <p>aquí algo de texto</p>
</div>
<div id='box2'>
    <p>otro parrafo bla bla bla</p>
</div>
```

Podemos hacer:

```
var box2 = document.getElementById('box2');
box2.insertAdjacentHTML('beforebegin', '<div><p>un parrafo nuevo.</p></div>');

// beforebegin - El nuevo HTML es insertado justo antes del elemento, a la misma altura (hermano).
// afterbegin - El nuevo HTML se inserta dentro del elemento, antes del primer hijo.
// beforeend - El nuevo HTML se inserta dentro del elemento, después del último hijo.
// afterend - El nuevo HTML es insertado justo después del elemento, a la misma altura (hermano).
```

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>var para = document.createElement("p"); var node = document.createTextNode("Esto es nuevo."); para.appendChild(node);  var element = document.getElementById("div1"); var child = document.getElementById("p1"); element.insertBefore(para,child);</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;div id="div1"&gt; &lt;p id="p1"&gt;Este es un parrafo.&lt;/p&gt; &lt;p id="p2"&gt;Este otro parrafo.&lt;/p&gt; &lt;/div&gt;  &lt;script src=codigo.js&gt;&lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	

# Añadir/eliminar/modificar Clases

```
// Tomamos un elemento
var cabecera = document.getElementById('cabecera');

// elimina una clase del elemento
cabecera.classList.remove('foo');

// Añade una clase si no existe
cabecera.classList.add('otra');

// añade o elimina varias clases a la vez
cabecera.classList.add('foo', 'bar');

cabecera.classList.remove('foo', 'bar');

// Si la clase existe la elimina, si no existe, la crea
cabecera.classList.toggle('visible');

// Devuelve true si el elemento contiene esa clase
cabecera.classList.contains('foo');
```

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>function myFunction() {   var element =   document.getElementById("miDIV" );   element.classList.add("miestilo"); }</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;style&gt;   .miestilo {     width: 100%;     padding: 25px;     background-color: coral;     color: white;     font-size: 25px;     box-sizing: border-box;   } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;    &lt;p&gt;Haga clic en el botón "Probar" para agregar la clase   "miestilo" al elemento DIV&lt;/p&gt;    &lt;button onclick="myFunction()"&gt;Probar&lt;/button&gt;    &lt;div id="miDIV"&gt;   Este es un elemento-modelo de caja   &lt;/div&gt;   &lt;script src=codigo.js&gt;&lt;/script&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<div>Haga clic en el botón "Probar" para agregar la clase "miestilo" al elemento DIV</div> <div><div>Probar</div><div>Este es un elemento-modelo de caja</div></div> <div>Despues....</div> <div>Haga clic en el botón "Probar" para agregar la clase "miestilo" al elemento DIV</div> <div><div>Probar</div><div>Este es un elemento-modelo de caja</div></div>

# Fuentes Consultadas

<https://www.w3schools.com/js/default.asp>

<http://juanmirod.github.io/2019/06/24/chuleta-dom.html>