

DOM (Modelo de Objetos del Documento)

El DOM es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante JS para cambiar dinámicamente los contenidos y aspecto de la página.

El modelo de objeto de documento (DOM) es una interfaz de programación para los documentos HTML y XML. Facilita una representación estructurada del documento y define de qué manera los programas pueden acceder, al fin de modificar, tanto su estructura, estilo y contenido.

El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. Esencialmente, conecta las páginas web a scripts o lenguajes de programación

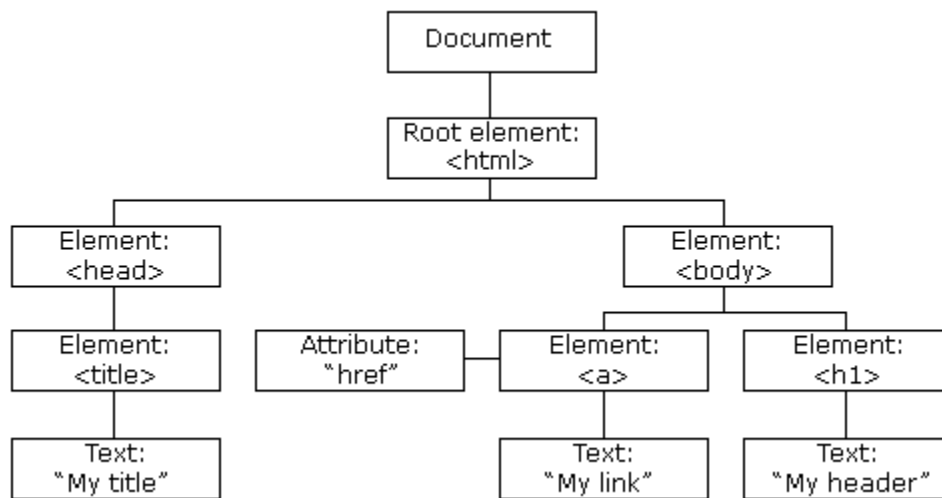
El W3C DOM estándar forma la base del funcionamiento del DOM en muchos navegadores modernos. Varios navegadores ofrecen extensiones más allá del estándar W3C, hay que ir con extremo cuidado al utilizarlas en la web, ya que los documentos pueden ser consultados por navegadores que tienen DOMs diferentes.

En el comienzo, JavaScript y el DOM estaban herméticamente enlazados, pero después se desarrollaron como entidades separadas. El contenido de la página es almacenado en DOM y el acceso y la manipulación se hace vía JavaScript, podría representarse aproximadamente así:

API(web o página XML) = DOM + JS(lenguaje de script)

El DOM fue diseñado para ser independiente de cualquier lenguaje de programación particular, hace que la presentación estructural del documento sea disponible desde un simple y consistente API. Aunque en este manual nos centramos exclusivamente en JavaScript, las directrices del DOM pueden construirse para cualquier lenguaje

Véase como está organizado el DOM cada vez que se carga la página se crea la siguiente estructura,



Tipos de datos importantes

Esta parte intenta describir, de la manera más simple posible, los diferentes objetos y tipos. Pero hay que conocer una cantidad de tipos de datos diferentes que son utilizados por el API. Para simplificarlo, los ejemplos de sintaxis en esta API se refieren a nodos como **elements**, a una lista de nodos como **nodeLists** (o simples elementos) y a nodos de atributo como **attributes**.

Para seleccionar elementos del DOM

Si nos encontramos en nuestro código Javascript y queremos hacer modificaciones en un elemento de la página HTML, lo primero que debemos hacer es buscar dicho elemento. Para ello, se suele intentar identificar el elemento a través de alguno de sus atributos más utilizados, generalmente el **id** o la **clase**.

| Métodos de búsqueda | Descripción |
|--|---|
| ELEMENT <code>.getElementById(id)</code> | Busca el elemento HTML con el id id . Si no, devuelve NULL . |
| ARRAY <code>.getElementsByClassName(class)</code> | Busca elementos con la clase class . Si no, devuelve <code>[]</code> . |
| ARRAY <code>.getElementsByName(name)</code> | Busca elementos con atributo name name . Si no, devuelve <code>[]</code> . |
| ARRAY <code>.getElementsByTagName(tag)</code> | Busca elementos tag . Si no encuentra ninguno, devuelve <code>[]</code> . |

Figura 1. Métodos de búsqueda tradicional

| Método de búsqueda | Descripción |
|--|---|
| ELEMENT <code>.querySelector(sel)</code> | Busca el primer elemento que coincide con el selector CSS sel . Si no, NULL . |
| ARRAY <code>.querySelectorAll(sel)</code> | Busca todos los elementos que coinciden con el selector CSS sel . Si no, <code>[]</code> . |

Figura 2. Métodos de búsqueda actuales

Analizar cómo seleccionar con las siguientes interfaces, véanse los ejemplos:

1. `getElementById("id")`
2. `querySelector("selectorCSS")`
3. `querySelectorAll("selectorCSS")`

Ejemplo 1:

```
<!DOCTYPE html>
<html>
<body>

<p id="ejemplo1">Clic en el botón para cambiar el texto</p>

<button onclick="cambioTexto()">Intentar</button>

<script>
function cambioTexto() {
    document.getElementById("ejemplo1").innerHTML = "Se cambió el texto exitosamente";
}
</script>
</body>
</html>
```

Ejemplo 2:

```
<!DOCTYPE html>
<html>
<body>
<h1>Sumar dos números</h1>
<p>Por favor, introduce dos números:</p>
<input type="number" id="num1"><br>
<input type="number" id="num2"> <br>
<button type="button" onclick=" sumar ()">Sumar</button>
<p id="sumando"></p>
<script>
function sumar() {
    var x,y,suma,text;
    x = document.getElementById("num1").value;
    y = document.getElementById("num2").value;
    if (isNaN(x) || isNaN(y)) {
        text = "Es necesarios introducir dos números válidos";
    } else {
        //si no ponemos parseFloat concatenaría x con y
        suma=parseFloat(x)+parseFloat(y);
        text= suma;
    }
    document.getElementById("sumando").innerHTML = text;
}
</script>
</body>
</html>
```

Ejemplo 3:

```
<!DOCTYPE html>
<html>
<head>
<style>
#myDIV {
  border: 1px solid black;
}
</style>
</head>
<body>

<div id="myDIV">
  <h2 class="example">Un encabezado con una Clase (.)</h2>
  <p class="example">Un parrado con una Clase (.) </p>
</div>

<div id="myDIV2">
  <p id="otro"> Otro parrafo con ID (#)</p>
</div>
<p>Dar clic en el boton para ver el cambio</p>

<button onclick="cambio()">Intentar</button>

<script>
function cambio() {
  var x = document.getElementById("myDIV");
  x.querySelector(".example").innerHTML = "Selector con Clase";

  var y = document.getElementById("myDIV2");
  y.querySelector("#otro").innerHTML = "Selector con ID"

}
</script>

</body>
</html>
```

Ejemplo 4:

```
<!DOCTYPE html>
<html>
<body>

<p>Este es un elemento de parrafo</p>

<p>Este es otro elemento de parrafo</p>

<p>Dar clic en el boton para agregar color de fondo en el primer elemento (indice 0) en el
documento.</p>

<button onclick="myFunction()">Intentar</button>

<p><strong>Note:</strong> The querySelectorAll() No esta soportado en IE 8.</p>

<script>
function myFunction() {
    var x = document.querySelectorAll("p");
    x[0].style.backgroundColor = "red";
}
</script>

</body>
</html>
```

Ejemplo 5

```
<!DOCTYPE html>
<html>
<head>
<style>
.example {
    border: 1px solid black;
    margin: 5px;
}
</style>
</head>
<body>

<h1>Un elemento h1</h1>

<div>Un elemento div</div>

<p>Un elemento de parrafo</p>

<p>otro elemento de parrafo</p>

<div class="example">
    <p>Un elemento de parrafo dentro de un div</p>
</div>

<p>Clic en el boton para cambiar el color de fondo</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction() {
    var x = document.querySelectorAll("p");
    var i;
    for (i = 0; i < x.length; i++) {
        x[i].style.backgroundColor = "green";
    }
}
</script>

</body>
</html>
```

Ejemplo 5:

```
<!DOCTYPE html>
<html>
<body>

<h1>Un elemento h1</h1>

<h2>Un elemento h2</h2>

<div>Un elemento div</div>

<p>Un elemento de parrafo</p>

<p>Un parrafo con un elemento <span style="color:brown;">span</span></p>

<p>Clic en el boton para cambiar el color de fondo en los elementos h2, div y span.</p>

<button onclick="myFunction()">intentar</button>

<script>
function myFunction() {
  var x = document.querySelectorAll("h2, div, span");
  var i;
  for (i = 0; i < x.length; i++) {
    x[i].style.backgroundColor = "grey";
  }
}
</script>

</body>
</html>
```