

Lección 24

# Manipulación del DOM I

¿DOM? ¿BOM?



# DOM

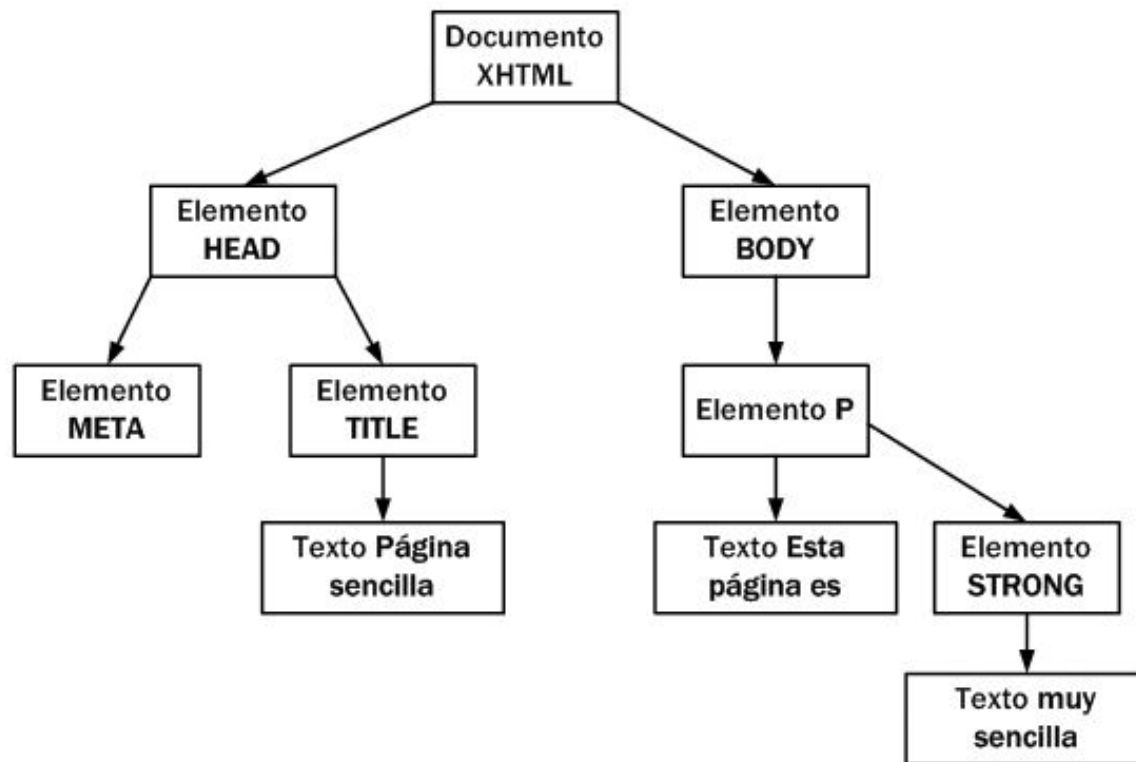
---

El modelo de objeto de documento (DOM) es una interfaz de programación para los documentos HTML y XML.

**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 con el fin de modificar, tanto su estructura, como su estilo y contenido.

# DOM

El DOM es una representación completamente orientada al objeto de la página web y puede ser modificado con un lenguaje de script como JavaScript.



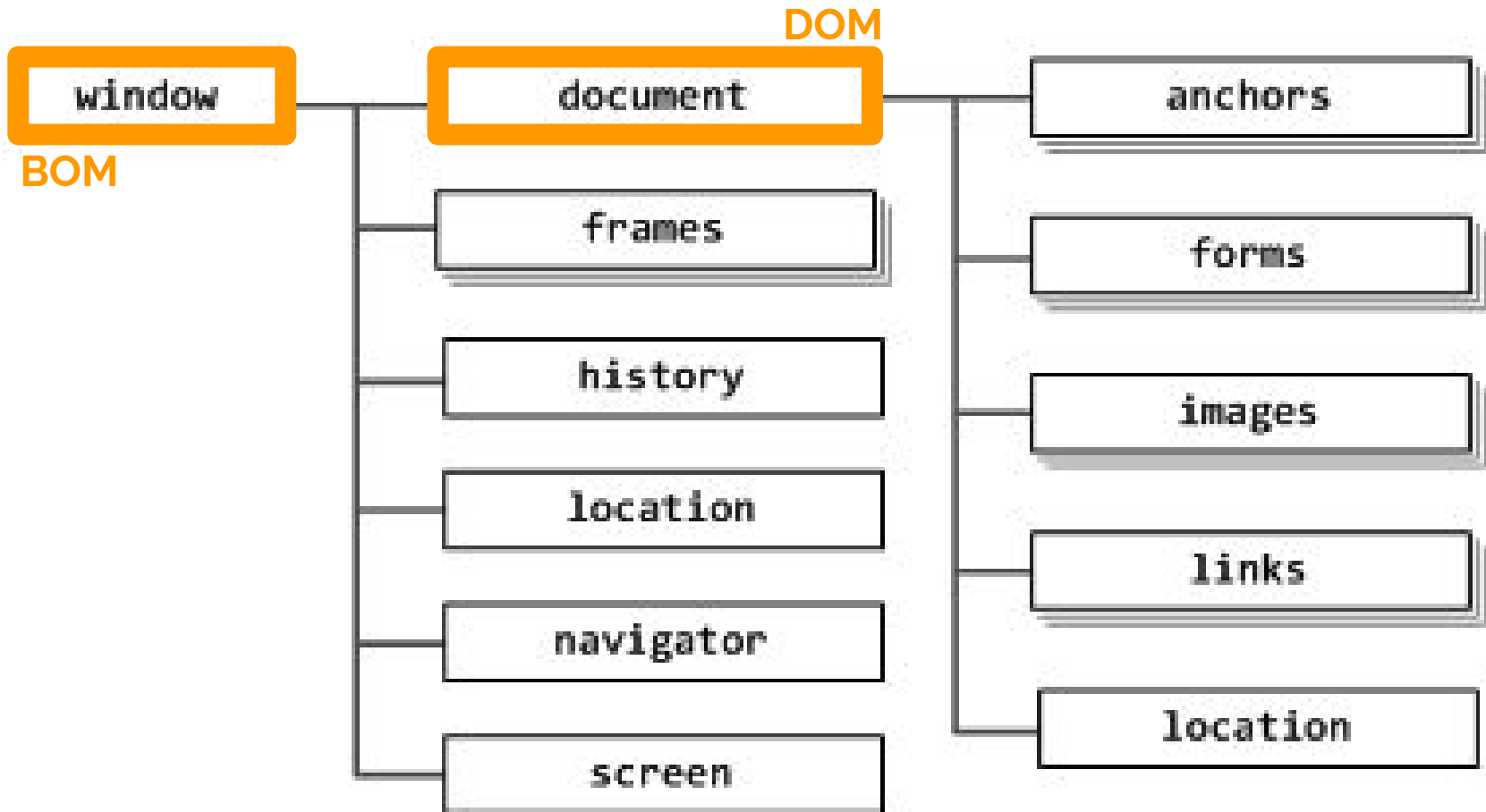
# BOM

---

El BOM (*Browser Object Model*), permite acceder y modificar las propiedades de las ventanas del propio navegador. Con él se puede redimensionar y mover el navegador, modificar textos de la barra de estado y realizar manipulaciones no relacionadas con el HTML.

El mayor inconveniente de BOM es que, al contrario de lo que sucede con DOM, ninguna entidad se encarga de su estandarización o de definir parámetros mínimos de interoperabilidad entre navegadores.

# BOM y DOM



# Obtener elementos: Selectores



## .getElementById

---

```
10 // Seleccionar un elemento HTML por su ID
11
12 document.getElementById('caja');
13
```

La función `getElementById()` devuelve el elemento XHTML cuyo atributo `id` coincide con el parámetro indicado en la función.

Como las `id` son únicas, la función devuelve únicamente el elemento deseado.



## .getElementsByClassName

---

```
23 // Seleccionar un elemento por clase
24
25 document.getElementsByClassName('caja');
26
27 // Seleccionar elementos que tengan más de una clase
28
29 document.getElementsByClassName('rojo prueba');
30
31
```

Esta función obtiene todos los elementos de la página XHTML cuya **clase** sea igual que el parámetro que se le pasa a la función.

## .getElementsByTagName

---

```
25 // Seleccionar un elemento por etiqueta
26
27 document.getElementsByTagName('p');
28
```

Esta función obtiene todos los elementos de la página XHTML cuya **etiqueta** sea igual que el parámetro que se le pasa a la función.

## .querySelector y .querySelectorAll

---

```
1 // Seleccionar el primer selector del grupo
2
3 document.querySelector('.mi-clase');
4
5 // Seleccionar toda la colección de selectores del grupo
6
7 document.querySelectorAll('.mi-clase');
8
```

La función `querySelector()` acepta como parámetro un selector que identifica el elemento (o elementos) a seleccionar.

En el caso de esta función, únicamente es devuelto el primer elemento que cumple la condición. Si no existe el elemento, el valor retornado es `null`.

## .querySelectorAll

---

Esta función toma todos los elementos con el selector asignado y devuelve un objeto de tipo NodeList con los elementos que coincidan.

```
79 // Seleccionar múltiples elementos
80
81 document.querySelectorAll('div, a, script');
82 |
```

La diferencia entre usar `document.querySelector` y `document.getElementById` es que el primero es compatible desde IE8+ mientras que el segundo es compatible desde IE5

Alterar contenido  
de los elementos



## Alterar contenido de los elementos

---

Podemos alterar el contenido de los elementos de nuestro DOM de diversas formas:

- `.innerHTML`: nos permite acceder al código HTML comprendido entre las etiquetas de apertura y cierre que definen el nodo y modificarlo si lo deseamos. Con esta se puede modificar completamente el HTML, no sólo el texto.

## .innerText y .textContent

- .textContent: Sirve para extraer (o modificar) el texto completo contenido en todos los nodos de un nodo padre. Se rescatan todos los elementos anidados.
- .innerText: Su comportamiento es muy similar al de textContent, aunque no exactamente igual. innerText no es reconocido por todos los navegadores y no se considera un estándar válido

```
1 <div id="t"><div>lions,  
2 tigers</div><div style="visibility:hidden">and bears</div></div>
```

text.html hosted with ❤ by GitHub

[view raw](#)

**innerText**

"lions, tigers"

**textContent**

"lions,\ntigersand bears"

## .value

---

- La propiedad value establece o retorna el atributo **value** de un campo de texto, pudiendo ser el valor por defecto o el texto que ingresa el usuario.
  - Valor por defecto: ""

Correo

El atributo **value** del input es:

- Valor ingresado por el usuario:  
"sumate@laboratoria.la"

Correo

El atributo **value** del input es: sumate@laboratoria.la