



# Programación con Javascript

## Unidad 4: DOM y BOM



## Indice

### Unidad 4: DOM y BOM

- BOM: Browser Object Model





## Objetivos

### Que el alumno logre:

- Conocer modelo de objetos de documento de Javascript



## BOM (Browser Object Model)

Las versiones 3.0 de los navegadores Internet Explorer y Netscape Navigator introdujeron el concepto de Browser Object Model o BOM, que permite acceder y modificar las propiedades de las ventanas del propio navegador.

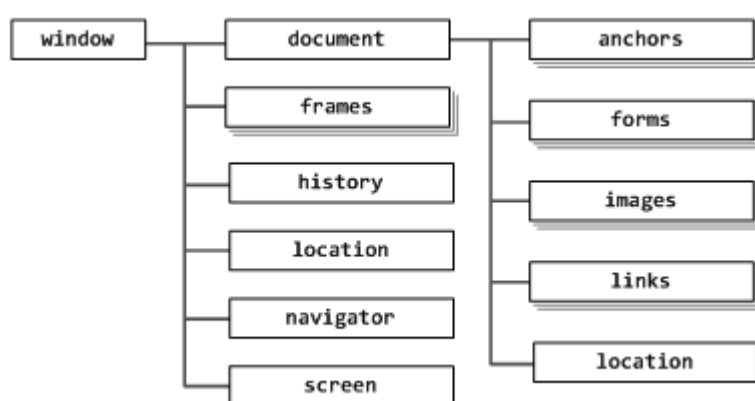
Mediante BOM, es posible redimensionar y mover la ventana del navegador, modificar el texto que se muestra en la barra de estado y realizar muchas otras manipulaciones no relacionadas con el contenido de la página HTML.

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

Algunos de los elementos que forman el BOM son los siguientes:

- Crear, mover, redimensionar y cerrar ventanas de navegador.
- Obtener información sobre el propio navegador.
- Propiedades de la página actual y de la pantalla del usuario.
- Gestión de cookies.
- Objetos ActiveX en Internet Explorer.

El BOM está compuesto por varios objetos relacionados entre sí. El siguiente esquema muestra los objetos de BOM y su relación:



En el esquema anterior, los objetos mostrados con varios recuadros superpuestos son arrays. El resto de objetos, representados por un rectángulo individual, son objetos simples. En cualquier caso, todos los objetos derivan del objeto window.



## Objeto Window

El objeto **window** representa la ventana completa del navegador. Mediante este objeto, es posible mover, redimensionar y manipular la ventana actual del navegador. Incluso es posible abrir y cerrar nuevas ventanas de navegador.

Si una página emplea frames, cada uno de ellos se almacena en el array frames, que puede ser accedido numéricamente (window.frames[0]) o, si se ha indicado un nombre al frame, mediante su nombre (window.frames["nombre del frame"]).

Como todos los demás objetos heredan directa o indirectamente del objeto window, no es necesario indicarlo de forma explícita en el código JavaScript. En otras palabras:

```
window.frames[0] == frames[0]  
window.document == document
```

BOM define cuatro métodos para manipular el tamaño y la posición de la ventana:

- *moveBy(x, y)* desplaza la posición de la ventana x píxel hacia la derecha y y píxel hacia abajo. Se permiten desplazamientos negativos para mover la ventana hacia la izquierda o hacia arriba.
- *moveTo(x, y)* desplaza la ventana del navegador hasta que la esquina superior izquierda se encuentre en la posición (x, y) de la pantalla del usuario. Se permiten desplazamientos negativos, aunque ello suponga que parte de la ventana no se visualiza en la pantalla.
- *resizeBy(x, y)* redimensiona la ventana del navegador de forma que su nueva anchura sea igual a (anchura\_anterior + x) y su nueva altura sea igual a (altura\_anterior + y). Se pueden emplear valores negativos para reducir la anchura y/o altura de la ventana.
- *resizeTo(x, y)* redimensiona la ventana del navegador hasta que su anchura sea igual a x y su altura sea igual a y. No se permiten valores negativos.

Los navegadores son cada vez menos permisivos con la modificación mediante JavaScript de las propiedades de sus ventanas. De hecho, la mayoría de navegadores permite a los usuarios bloquear el uso de JavaScript para realizar cambios de este tipo. De esta forma, una aplicación nunca debe suponer que este tipo de funciones están disponibles y funcionan de forma correcta.

A continuación se muestran algunos ejemplos de uso de estas funciones:

```
// Mover la ventana 20 píxel hacia la derecha y 30 píxel hacia abajo  
window.moveBy(20, 30);
```

```
// Redimensionar la ventana hasta un tamaño de 250 x 250
```



```
window.resizeTo(250, 250);
```

```
// Agrandar la altura de la ventana en 50 píxel  
window.resizeBy(0, 50);
```

```
// Colocar la ventana en la esquina izquierda superior de la ventana  
window.moveTo(0, 0);
```

## Objeto document

El objeto document es el único que pertenece tanto al DOM (como se vio en el capítulo anterior) como al BOM. Desde el punto de vista del BOM, el objeto document proporciona información sobre la propia página HTML.

## Objeto location

El objeto location es uno de los objetos más útiles del BOM. Debido a la falta de estandarización, location es una propiedad tanto del objeto window como del objeto document.

El objeto location representa la URL de la página HTML que se muestra en la ventana del navegador y proporciona varias propiedades útiles para el manejo de la URL:



Propiedad	Descripción
hash	El contenido de la URL que se encuentra después del signo # (para los enlaces de las anclas) <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> hash = #seccion
host	El nombre del servidor <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> host = <code>www.ejemplo.com</code>
hostname	La mayoría de las veces coincide con host, aunque en ocasiones, se eliminan las <code>www</code> del principio <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> hostname = <code>www.ejemplo.com</code>
href	La URL completa de la página actual <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> URL = <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code>
pathname	Todo el contenido que se encuentra después del host <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> pathname = <code>/ruta1/ruta2/pagina.html</code>
port	Si se especifica en la URL, el puerto accedido <code>http://www.ejemplo.com:8080/ruta1/ruta2/pagina.html#seccion</code> port = 8080 La mayoría de URL no proporcionan un puerto, por lo que su contenido es vacío <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> port = (vacío)
protocol	El protocolo empleado por la URL, es decir, todo lo que se encuentra antes de las dos barras inclinadas // <code>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion</code> protocol = <code>http:</code>
search	Todo el contenido que se encuentra tras el símbolo ?, es decir, la consulta o "query string" <code>http://www.ejemplo.com/pagina.php?variable1=valor1&amp;variable2=valor2</code> search = <code>?variable1=valor1&amp;variable2=valor2</code>

De todas las propiedades, la más utilizada es `location.href`, que permite obtener o establecer la dirección de la página que se muestra en la ventana del navegador.

Además de las propiedades de la tabla anterior, el objeto `location` contiene numerosos métodos y funciones. Algunos de los métodos más útiles son los siguientes:

```
// Método assign()
location.assign("http://www.ejemplo.com"); // Equivalente a location.href =
"http://www.ejemplo.com"

// Método replace()
location.replace("http://www.ejemplo.com");
// Similar a assign(), salvo que se borra la página actual del array history del navegador
```



```
// Método reload()
location.reload(true);
/* Recarga la página. Si el argumento es true, se carga la página desde el servidor.
   Si es false, se carga desde la cache del navegador */
```

## Objeto Navigator

El objeto navigator es uno de los primeros objetos que incluyó el BOM y permite obtener información sobre el propio navegador. En Internet Explorer, el objeto navigator también se puede acceder a través del objeto clientInformation.

Aunque es uno de los objetos menos estandarizados, algunas de sus propiedades son comunes en casi todos los navegadores. A continuación se muestran algunas de esas propiedades:

Propiedad	Descripción
appCodeName	Cadena que representa el nombre del navegador (normalmente es Mozilla)
appName	Cadena que representa el nombre oficial del navegador
appMinorVersion	(Sólo Internet Explorer) Cadena que representa información extra sobre la versión del navegador
appVersion	Cadena que representa la versión del navegador
browserLanguage	Cadena que representa el idioma del navegador
cookieEnabled	Boolean que indica si las cookies están habilitadas
cpuClass	(Sólo Internet Explorer) Cadena que representa el tipo de CPU del usuario ("x86", "68K", "PPC", "Alpha", "Other")
javaEnabled	Boolean que indica si Java está habilitado
language	Cadena que representa el idioma del navegador
mimeType	Array de los tipos MIME registrados por el navegador
onLine	(Sólo Internet Explorer) Boolean que indica si el navegador está conectado a Internet
oscpu	(Sólo Firefox) Cadena que representa el sistema operativo o la CPU
platform	Cadena que representa la plataforma sobre la que se ejecuta el navegador





<code>plugins</code>	Array con la lista de plugins instalados en el navegador
<code>preference()</code>	(Sólo Firefox) Método empleado para establecer preferencias en el navegador
<code>product</code>	Cadena que representa el nombre del producto (normalmente, es Gecko)
<code>productSub</code>	Cadena que representa información adicional sobre el producto (normalmente, la versión del motor Gecko)
<code>securityPolicy</code>	Sólo Firefox
<code>systemLanguage</code>	(Sólo Internet Explorer) Cadena que representa el idioma del sistema operativo
<code>userAgent</code>	Cadena que representa la cadena que el navegador emplea para identificarse en los servidores
<code>userLanguage</code>	(Sólo Explorer) Cadena que representa el idioma del sistema operativo
<code>userProfile</code>	(Sólo Explorer) Objeto que permite acceder al perfil del usuario

El objeto **navigator** se emplea habitualmente para detectar el tipo y/o versión del navegador en las aplicaciones cuyo código difiere para cada navegador. Además, se emplea para detectar si el navegador tiene habilitadas las cookies y Java y también para comprobar los plugins disponibles en el navegador.

## Objeto screen

El objeto screen se utiliza para obtener información sobre la pantalla del usuario. Uno de los datos más importantes que proporciona el objeto screen es la resolución del monitor en el que se están visualizando las páginas. Los diseñadores de páginas web necesitan conocer las resoluciones más utilizadas por los usuarios para adaptar sus diseños a esas resoluciones.

Las siguientes propiedades están disponibles en el objeto screen:



Propiedad	Descripción
<code>availHeight</code>	Altura de pantalla disponible para las ventanas
<code>availWidth</code>	Anchura de pantalla disponible para las ventanas
<code>colorDepth</code>	Profundidad de color de la pantalla (32 bits normalmente)
<code>height</code>	Altura total de la pantalla en píxel
<code>width</code>	Anchura total de la pantalla en píxel

La altura/anchura de pantalla disponible para las ventanas es menor que la altura/anchura total de la pantalla, ya que se tiene en cuenta el tamaño de los elementos del sistema operativo como por ejemplo la barra de tareas y los bordes de las ventanas del navegador.

Además de la elaboración de estadísticas de los equipos de los usuarios, las propiedades del objeto `screen` se utilizan por ejemplo para determinar cómo y cuánto se puede redimensionar una ventana y para colocar una ventana centrada en la pantalla del usuario.

El siguiente ejemplo redimensiona una nueva ventana al tamaño máximo posible según la pantalla del usuario:

```
window.moveTo(0, 0);  
window.resizeTo(screen.availWidth, screen.availHeight);
```



## Resumen

### En esta Unidad...

Trabajamos con DOM y BOM

### En la próxima Unidad...

Trabajaremos con condicionales y otras estructuras