

Actividades extra UD1

Introducción al BOM (*Browser Object Model*)

María Rodríguez Fernández mariarfer@educastur.org

Al final de este documento...

- Conoceremos los objetos `window`, `location`, `navigation` y `screen`

BOM

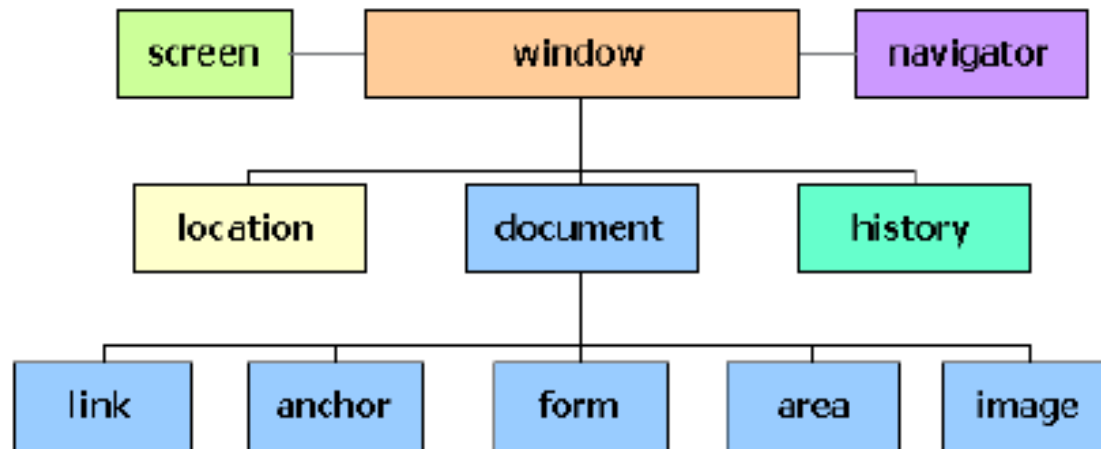
(Browser Object Model)

- Permite acceder y modificar las propiedades de las ventanas del propio navegador:
 - Redimensionar y mover la ventana del navegador
 - Modificar el texto de la barra de estado
 - Acceder a la URL
 - Obtener información sobre el propio navegador...

PROBLEMA: Ninguna entidad se encarga de estandarizarlo, garantizando mínimos de compatibilidad entre navegadores

Objetos de alto nivel

- El BOM está formado por varios objetos relacionados entre sí
- Aquí se muestran los principales objetos:



Objeto screen

- Se usa para obtener información sobre la pantalla del usuario

availHeight availWidth	Altura/Anchura total para las ventanas Tiene en cuenta elementos del S.O. como la barra de tareas o los anchos de pantalla
colorDepth	Profundidad de color
height width	Altura/anchura total de la pantalla

Objeto navigator

- Contiene información sobre el navegador que estamos usando
 - Habitualmente se emplea para detectar tipo/versión del navegado y para comprobar si las cookies están habilitadas



appCodeName	Cadena que representa el nombre del navegador
appName	Cadena que representa el nombre oficial del navegador
appVersion	Cadena que representa la versión del navegador
platform	Cadena que representa la plataforma sobre la que se ejecuta el navegador
cookieEnabled	Indica si las cookies están habilitadas

EJERCICIO PROPUESTO



- Usa el objeto **navigator** para mostrar la siguiente información:
 - Nombre del navegador
 - Nombre oficial del navegador
 - Versión del navegador
 - Si las cookies están habilitadas
 - Plataforma sobre la que se está ejecutando el navegador
 - Vendedor del navegador
 - Idioma en el navegador

Objeto `window`



- Contenedor principal de todo el contenido que se visualiza en el **navegador**
 - Representa cada ventana/pestaña
- Acceso (podemos omitir `window`):
 - `window.nombrePropiedad`
 - `window.nombreMetodo([parámetros])`

name	Nombre de la ventana
outerWidth, outerHeight	Ancho y alto incluyendo toolbar y scrollbar
innerWidth, innerHeight	Ancho y alto sin incluir toolbar y scrollbar
screenX, screenY	Distancia desde la esquina superior izquierda

Manipulando el tamaño de la ventana

- Los navegadores cada vez son menos permisivos con la **modificación mediante JS** de las propiedades de sus ventanas
 - La mayoría de los navegadores permiten a los usuarios bloquear mediante JS este tipo de cambios

moveBy(x,y) moveTo(x,y)	Desplaza la posición de la ventana x a la derecha e y píxeles hacia abajo (By) / a la posición exacta (To)
resizeBy(x,y) resizeTo(x,y)	Redimensiona la ventana añadiendo x a la anchura e y píxeles a la altura (By) / a un tamaño exacto (To)

Otros métodos de `window`

<code>close()</code>	Cierra la ventana actual
<code>open(url)</code>	Abre una nueva ventana que carga la URL especificada
<code>focus()</code>	Coloca el foco en la ventana actual
<code>blur()</code>	Elimina el foco de la ventana actual
<code>alert()</code>	Muestra un mensaje con un botón de aceptar
<code>prompt()</code>	Muestra una ventana de diálogo para introducir datos
<code>confirm()</code>	Muestra un mensaje con un botón de aceptar y otro de cancelar

Objeto **location**



- Contiene información referente a la URL actual
- Podemos acceder con **window.location** o **location**

href	URL completa
host	Nombre del servidor
pathname	Contenido tras el host
hash	Contenido tras # (secciones)
port	Puerto
protocol	Protocolo
query	Contenido tras ? (<i>query string</i>)

href es la más usada para leer o establecer la dirección de la página que muestra el navegador

Ejemplo

href	http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion
host	www.ejemplo.com
pathname	ruta1/ruta2/pagina.html#seccion
hash	seccion
port	-
protocol	http:
query	-

Métodos de `location`

<code>assign(url)</code>	Establece la URL de la página Es equivalente a <code>location.href=url</code>
<code>replace(url)</code>	Establece la URL de la página Similar a <code>assign()</code> , salvo que se borra la página del historial del navegador
<code>reload(bool)</code>	Recarga la página. Si el argumento es <code>true</code> se carga desde el servidor, de lo contrario se carga desde la caché del navegador.

Temporizadores

- Función **setTimeout(nombreFuncion,miliseg)**
 - Permite ejecutar una función una vez que haya **transcurrido** un periodo de tiempo determinado.
- Función **setInterval(nombreFuncion,miliseg)**
 - Permite ejecutar una función **cada** intervalo de tiempo especificado.

OJO! La referencia a **nombreFuncion** se especifica sin paréntesis
De lo contrario estamos haciendo una llamada a la misma

Intervalos de tiempo

```
setTimeout(hazAlgo, 1000);
```

```
function hazAlgo(){  
  console.log("Hola");  
}
```

```
setInterval(hazAlgo, 1000);
```

```
function hazAlgo(){  
  console.log("Hola");  
}
```

SINTAXIS 1 Función nominal

SINTAXIS2 Función anónima

```
setTimeout(function(){ console.log("Muestra mensaje");}, 1000);
```

Parando el timeout

- La llamada a `setTimeout()/setInterval()` devuelve un identificador de reloj
- Si queremos impedir que se ejecute, podemos usar dicho identificador junto con la función `clearTimeout(id)/clearInterval(id)`

Parando el timeout

```
function muestraMensaje() {  
    alert("Han transcurrido 3 segundos");  
}  
var id = setTimeout(muestraMensaje, 3000);  
  
// Cancelamos la ejecución antes de que pasen 3seg  
clearTimeout(id);
```

Parando el intervalo

```
function muestraMensaje() {  
    alert("Aparezco cada segundo");  
}  
var id = setInterval(muestraMensaje, 1000);  
  
// Cancelamos la ejecución del intervalo  
clearInterval(id);
```

Intervalos de tiempo y parámetros

- Si queremos pasar parámetros a las funciones de intervalo podemos hacerlo añadiendo la lista de parámetros tras el tiempo

setTimeout(nombreFuncion,miliseg,[param1,param2...])
setInterval(nombreFuncion,miliseg,[param1,param2...])

```
function muestraMensaje(mensaje) {  
    alert(mensaje);  
}  
  
setTimeout(muestraMensaje, 1000, "Hola");
```

EJERCICIO PROPUESTO



- Realiza un script de que cambie automáticamente la URL de la página una vez transcurridos 10 segundos.
 - La URL se tomará aleatoriamente de entre un array con 10 URLs, definido estáticamente al comienzo del script.
 - Utiliza `setTimeout()` para establecer el tiempo de espera.

Función	Recibe	Devuelve	Descripción
cambiaURL(url)	url:cadena	-	Cambia la URL del navegador