

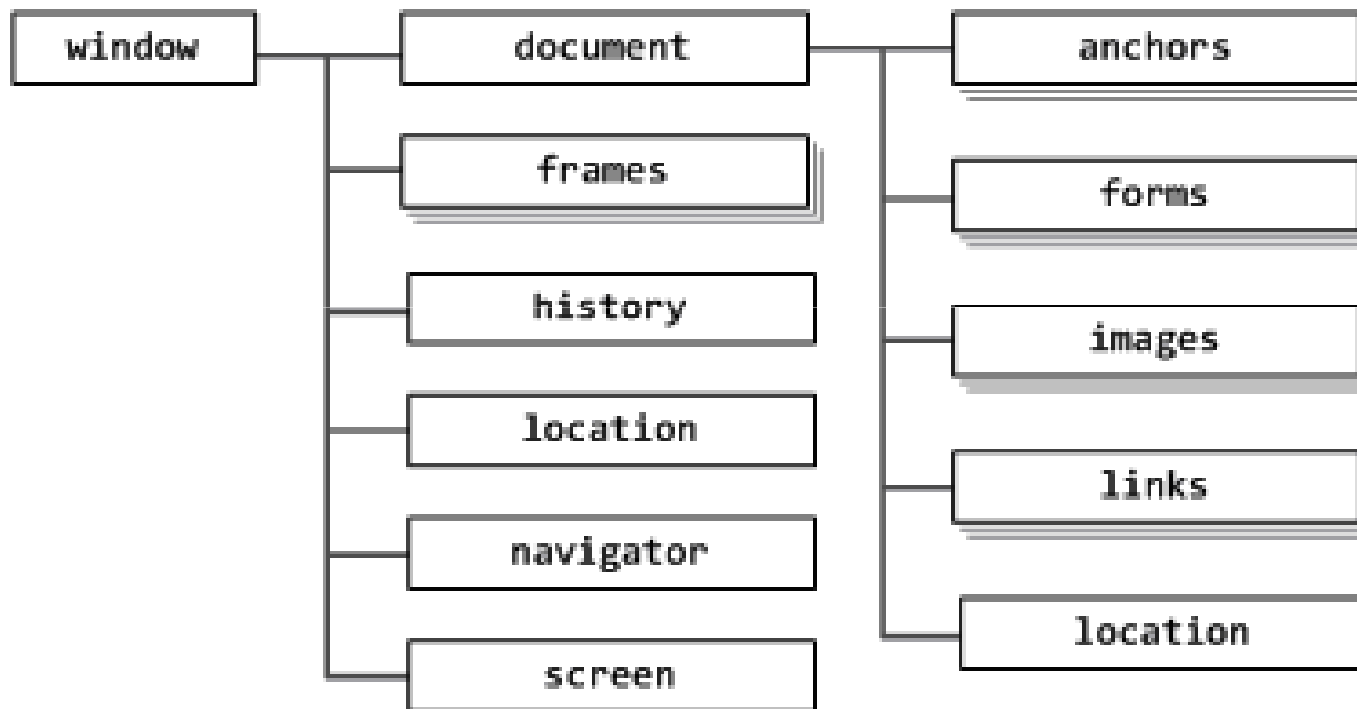
Tema 5. Modelo de Objetos del Navegador (BOM)

- Objetos del navegador
- Jerarquía de Objetos
- window
- open()
- frame
- location
- history
- document
- navigator
- screen

Objetos del navegador

- El Modelo de Objetos del Navegador (*Object Browser Model* - BOM) proporciona una colección de objetos que interactúa con la ventana del navegador independientemente del contenido.
- Estos objetos describen la ventana, el documento y todos los elementos que lo componen.
- Existe una relación asociativa o de agregación entre el conjunto de objetos predefinidos que determinan una jerarquía de objetos.
- El objeto *window* es el elemento central del BOM y todos los demás objetos están conectados al mismo.
- Para acceder a los objetos integrantes de otros objetos se utiliza el operador “.”.
- 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 ninguna entidad se encarga de estandarizarlo o definir unos mínimos de interoperabilidad entre navegadores.

Jerarquía de Objetos



window

- Define la ventana sobre la que estamos trabajando e incluye los objetos que hacen referencia a la barra de tareas, el documento o la secuencia de direcciones de la última sesión.
- 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 no tenemos marcos en el documento, no es necesario indicarlo explícitamente para acceder a sus propiedades o métodos.
- Si una página emplea marcos (*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.

Atributos de *window*

Atributo	Descripción
<i>document</i>	Objeto <i>document</i> . Asociado a la página Web que se muestra
<i>defaultStatus</i>	Texto de la barra de estado por defecto
<i>frames</i>	Array cuyos elementos son los marcos que contiene la ventana
<i>history</i>	Array con las URLS visitadas
<i>length</i>	Número de marcos
<i>location</i>	URL de la barra de dirección
<i>name</i>	Nombre de la ventana, o del marco actual
<i>opener</i>	Referencia al objeto <i>window</i> que abrió la ventana (mediante <i>open()</i>)
<i>parent</i>	Referencia al objeto <i>window</i> que contiene el marco
<i>self</i>	Es un nombre alternativo del <i>window</i> actual.
<i>status</i>	Texto de la barra de estado
<i>top</i>	Referencia a la ventana del nivel superior
<i>window</i>	Igual que <i>self</i>

Métodos de *window* (I)

Método	Descripción
<i>alert(text)</i>	Muestra el mensaje en un cuadro de diálogo
<i>blur()</i>	Elimina el foco del objeto <i>window</i> actual
<i>clearTimeout(id)</i>	Cancela el intervalo referenciado por 'id'
<i>close()</i>	Cierra la ventana activa
<i>confirm(text)</i>	Cuadro de diálogo con mensaje y dos botones (aceptar y cancelar). Devuelve true o false.
<i>focus()</i>	Captura el foco del ratón sobre el objeto <i>window</i> actual
<i>moveBy(x,y)</i>	Mueve el objeto <i>window</i> actual el número de pixels especificados por (x,y).
<i>moveTo(x,y)</i>	Mueve el objeto <i>window</i> actual a las coordenadas (x,y).
<i>open(URL,nom,ops)</i>	Abre la URL en la ventana <i>nom</i> . Si no existe, abrirá una ventana nueva en la que mostrará el contenido con las características especificadas en <i>ops</i>

Métodos de *window* (II)

Método	Descripción
<i>prompt(text,def)</i>	Muestra un cuadro de diálogo con <i>text</i> y espera una respuesta. El parámetro <i>resp</i> (opcional) muestra la respuesta por defecto. El método retorna una cadena con la respuesta introducida
<i>resizeBy(x, y)</i>	Redimensiona la ventana del navegador añadiendo x píxeles al ancho e y al alto. Acepta números negativos.
<i>resizeTo(x, y)</i>	Redimensiona la ventana del navegador a anchura x píxeles y altura y.
<i>scroll(x,y), scrollTo(x,y).</i>	Desplaza el objeto <i>window</i> actual a las coordenadas especificadas por (x,y).
<i>scrollBy(x,y).</i>	Desplaza el objeto <i>window</i> actual el número de píxeles especificado por (x,y).
<i>setTimeout(exp,t)</i>	Evalúa la expresión especificada después de que hayan pasado el número de milisegundos especificados en tiempo. Devuelve un valor que puede ser usado como identificador por <i>clearTimeout()</i> .

open()

- *open(URL, nombre, características)*
- Abre la URL en la ventana nombre. Si no existe, abrirá una ventana nueva en la que mostrará el contenido con las características especificadas.
- Las características indican si esta o no presente el elemento:
 - *toolbar* = [yes|no|1|0]. Barra de herramientas
 - *location* = [yes|no|1|0]. campo de localización
 - *directories* = [yes|no|1|0]. Botones de dirección
 - *status* = [yes|no|1|0]. Barra de estado
 - *menubar* = [yes|no|1|0]. Barra de menús
 - *scrollbars* = [yes|no|1|0]. Barras de desplazamiento
 - *resizable* = [yes|no|1|0]. Ventana redimensionable (con el ratón)
 - *width* = px. Ancho de la ventana en pixels
 - *height* = px. Alto de la ventana en pixels
 - *left* = px. Distancia en píxeles desde el lado izquierdo de la pantalla
 - *top* = px. Distancia en píxeles desde el lado superior de la pantalla

Ejemplo 1 (I)

```
<html>
<head>
<script type="text/javascript">
function moverventana()
{
    mi_ventana.moveBy(5,5);
    i++;
    if (i<40)
        setTimeout('moverventana()',100);
    else
        mi_ventana.close();
}
</script>
</head>
```

Ejemplo 1 (II)

```
<body>
<script language="javascript">
var opciones="left=100,top=100,width=250,height=150";
i= 0;
var mi_ventana = window.open("", "", opciones);
mi_ventana.document.open();
mi_ventana.document.write("una prueba de abrir
    ventanas");
mi_ventana.document.close();
mi_ventana.moveTo(400,100);
moverventana();
</script>
</body>
</html>
```

frame

- La ventana del navegador puede ser dividida en varios marcos que contengan cada uno de ellos una página en la que mostrar contenidos diferentes.
- Cada uno de estos marcos puede ser nombrado y referenciado mediante el vector *frame*, lo que nos permite cargar documentos en un marco sin que esto afecte al resto.
- Realmente cada elemento de *frame* es un objeto *window*, con sus atributos y métodos vistos anteriormente.
- Podemos acceder a cada marco numéricamente (*window.frames[0]*) o, si se ha indicado un nombre al marco, mediante su nombre (*window.frames["nombre del frame"]*).

Ejemplo 2 (I)

```
<html> <!-- exem2a.html -->
<frameset cols="40%,*">
  <frame name="izq" src="exem2b.html" scrolling="auto">
  <frameset rows="50%,*">
    <frame name="der1" src="exem2c.html" scrolling="auto">
    <frame name="der2" src="exem2c.html" scrolling="auto">
  </frameset>
</frameset>
</html>
```

```
<html><!-- exem2c.html -->
<body>
<script language="javascript">
document.write('<h1>soy el frame: ' + window.name +
  "</h1><br>");
</script>
</body>
</html>
```

Ejemplo 2 (II)

```
<html> <!-- exem2b.html -->
<head>
<script type="text/javascript">
function color(){ top.frames['der1'].document.bgColor="#ff5500";}
function cambia(){ top.frames['der2'].location='http://www.google.es';}
</script></head>
<body>
<script type="text/javascript">
document.write('<h1>soy el marco: ' + window.name + "</h1><br>");
for (i=0;i<top.frames.length;i++)
    document.write('nombre del marco ' + i + ': ' + top.frames[i].name +
        "<br>");
</script>
<hr>
<p><a href="javascript:color()">Cambia el color del frame <i>der1</i></a></p>
<p><a href="javascript:cambia()">Carga google <i>der2</i></a></p>
<p><a href="javascript:top.location=top.location">Inicio</a>
</body></html>
```

location

- El objeto *location* contiene la URL.
- Es una propiedad tanto de *window* como de *document*
- Su finalidad es modificar el objeto *location* para cambiar a una nueva URL, y extraer los componentes de dicha URL de forma separada.
 - protocolo://maquina_host[:puerto]/camino_al_recurso#ancla?query

Métodos	Descripción
<i>assign(URL)</i>	Carga la URL especificada y la añade al historial. Similar a <i>location.href=URL</i>
<i>reload(logico)</i>	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. Si se omite el argumento, se considera false
<i>replace(URL)</i>	Carga la URL especificada y reemplaza la página actual en el historial por la nueva.

Atributos de *location*

Atributo	Descripción
<i>hash</i>	Nombre del ancla de la URL
<i>host</i>	Nombre del servidor
<i>hostname</i>	La mayoría de las veces coincide con host, aunque en ocasiones, se eliminan las www del principio
<i>href</i>	La URL completa de la página actual
<i>pathname</i>	Camino al recurso de la URL
<i>port</i>	Si se especifica en la URL, el puerto accedido
<i>protocol</i>	El protocolo empleado por la URL
<i>search</i>	Todo el contenido que se encuentra tras el símbolo ?, es decir, la consulta o " <i>query string</i> "

Ejemplo 3

```
<html>
<head>
<title>Exemple 3</title>
</head>
<body>
<script type="text/javascript">
document.write("<p>Location <b>hash</b>: " + location.hash + "</p>");
document.write("<p>Location <b>host</b>: " + location.host + "</p>");
document.write("<p>Location <b>hostname</b>: " + location.hostname + "</p>");
document.write("<p>Location <b>href</b>: " + location.href + "</p>");
document.write("<p>Location <b>pathname</b>: " + location.pathname + "</p>");
document.write("<p>Location <b>port</b>: " + location.port + "</p>");
document.write("<p>Location <b>protocol</b>: " + location.protocol + "</p>");
</script>
</body>
</html>
```


history

- Este objeto guarda las referencias de los lugares visitados. Es una propiedad de *window*. Se utiliza, sobre todo, para movernos hacia delante o hacia atrás en la lista Webs visitadas.

Atributo	Descripción
<i>length</i>	Contiene el número de entradas del historial

Métodos	Descripción
<i>back()</i>	Vuelve a cargar la URL del documento anterior dentro del historial
<i>forward()</i>	Vuelve a cargar la URL del documento siguiente dentro del historial
<i>go(pos)</i>	Vuelve a cargar la URL del documento especificado por <i>pos</i> dentro del historial

document (I)

- El objeto *document* representa el contenido de toda la página que se está visualizando. Esto incluye el texto, imágenes, enlaces, formularios, etc.
- *document* es el único objeto que pertenece al BOM y a al DOM (Document Object Model)

Atributo	Descripción
<i>alinkColor</i>	Color de los enlaces activos
<i>anchors</i>	Vector con los enlaces internos existentes en el documento
<i>applets</i>	Vector con los <i>applets</i> existentes en el documento
<i>bgColor</i>	Color de fondo del documento
<i>fgColor</i>	Color del primer plano
<i>forms</i>	Vector con todos los formularios del documento
<i>images</i>	Vector con todas las imágenes del documento
<i>lastModified</i>	Cadena con la fecha de la última modificación del documento
<i>linkColor</i>	Color de los enlaces

document (II)

Atributo	Descripción
<i>links</i>	Array con los enlaces externos
<i>location</i>	Objeto <i>location</i>
<i>referrer</i>	Cadena con la URL del documento que llamó al actual
<i>title</i>	Cadena con el título del documento actual
<i>vlinkColor</i>	Color de los enlaces visitados
URL	El URL de la página actual

Métodos	Descripción
<i>clear()</i>	Limpia la ventana del documento
<i>open()</i>	Abre la escritura sobre el documento
<i>close()</i>	Cierra la escritura sobre el documento
<i>write()</i>	Escribe texto en el documento
<i>writeln()</i>	Escribe texto en el documento y añade un salto de línea

Ejemplo 4 (I)

```
<html>
<head>
<title>Exemple 4. Propietats de l'objecte document</title>
<script type="text/javascript">
var i=1;
var titulooriginal = document.title;
function cambiarTitulo()
{
    document.title = titulooriginal.substring(0,i);
    i++;
    if (i>titulooriginal.length) i=0;
    setTimeout('cambiarTitulo()',100);
}
</script>
</head>
```

Ejemplo 4 (II)

```
<body text='blue' link='red' bgcolor='yellow' vlink='grey'>
<script type="text/javascript">
with(document){
    write("Document <b>linkColor </b>: " + linkColor + "<br>");
    write("Document <b>bgColor</b>: " + bgColor + "<br>");
    write("Document <b>fgcolor</b>: " + fgColor + "<br>");

    fgColor="red";
    bgColor="cccccc";
}
cambiarTitulo();
</script>
</html>
```

navigator

- Da información relativa al navegador que esté utilizando el usuario.

Atributo	Descripción
<i>appCodeName</i>	Cadena que contiene el nombre del código del cliente
<i>appName</i>	Cadena que contiene el nombre del cliente (navegador).
<i>appVersion</i>	Cadena que contiene información sobre la versión del cliente.
<i>language</i>	Cadena de dos caracteres que contiene información sobre el idioma de la versión del cliente
<i>platform</i>	Cadena con la plataforma sobre la que se está ejecutando el programa cliente
<i>plugins</i>	Array que contiene todos los <i>plug-ins</i> soportados por el cliente
<i>userAgent</i>	Cadena que contiene la cabecera completa del agente, enviada en una petición HTTP.

Método	Descripción
<i>javaEnabled()</i>	Devuelve true si el cliente permite la utilización de Java, en caso contrario, devuelve false.

Ejemplo 5

```
<html>
<body>
<script type="text/javascript">
with(document){
    write("<h2>Propiedades de navigator</h2>");
    write("<p><b>appCodeName</b>: " + navigator.appCodeName + "<br>");
    write("<b>appName</b>: " + navigator.appName + "<br>");
    write("<b>appVersion</b>: " + navigator.appVersion + "<br>");
    write("<b>language</b>: " + navigator.language + "<br>");
    write("<b>platform</b>: " + navigator.platform + "<br>");
    write("<b>userAgent</b>: " + navigator.userAgent + "<br>");
    write("<b>javaEnabled</b>: " + navigator.javaEnabled() + "</p>");
}
</script>
</body>
</html>
```

screen

- Permite conocer la configuración de la pantalla del usuario.
- Todos sus atributos son de sólo lectura.

Atributo	Descripción
<i>height</i>	Altura de la resolución de la pantalla en píxeles
<i>width</i>	Anchura de la resolución de la pantalla en píxeles
<i>availHeight</i>	Altura de la resolución de la pantalla en píxeles que pueden utilizar las ventanas
<i>availWidth</i>	Anchura de la resolución de la pantalla en píxeles que pueden utilizar las ventanas
<i>pixelDepth</i>	Número de bits por pixel para representar los colores

Ejemplo 6

```
<body>
<script type="text/javascript">
if(screen){
    var texto1="Configuración de la pantalla "+
        screen.width+"x"+screen.height;
    var texto2=screen.pixelDepth+" bits -->
        "+Math.pow(2,screen.pixelDepth)+" colores.";
    var texto3="Configuración de la pantalla máxima"+
        screen.availWidth+"x"+screen.availHeight;
}else
    texto1="No se tu configuración de pantalla";
document.write("<p>"+texto1+"<br>"+texto2+"<br>"+texto3+"</p>");
moveTo(0,0);
resizeTo(screen.availWidth, screen.availHeight);
</script>
</body>
```