Curso de Javascript

Unidad Didáctica 09: Herencia

http://cursosdedesarrollo.com/

Ayuntamiento de Vitoria-Gasteiz
Vitoria-Gasteizko Udala

Índice de contenidos

http://cursosdedesarrollo.com/

Introducción

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

Introducción

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

Introducción

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

Introducción

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.

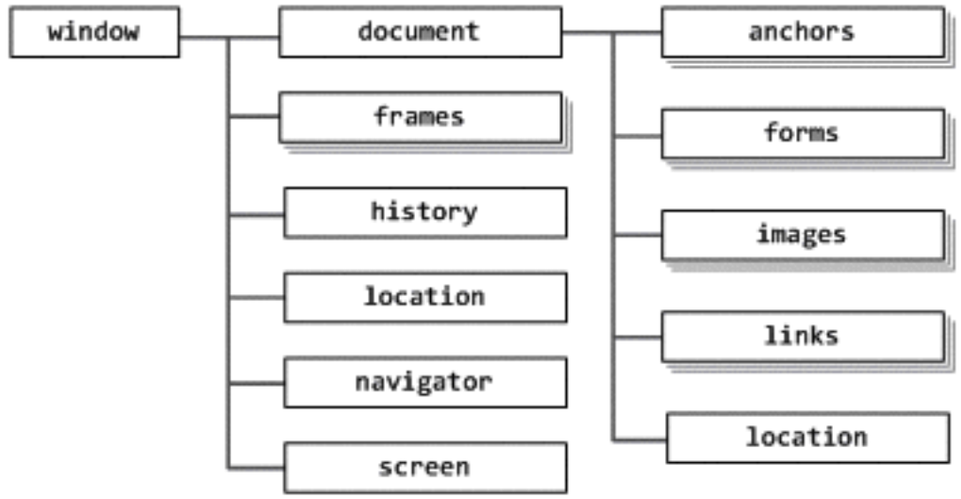
http://cursosdedesarrollo.com/

Introducción

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

http://cursosdedesarrollo.com/

Introducción



http://cursosdedesarrollo.com/

Introducción

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

http://cursosdedesarrollo.com/

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

http://cursosdedesarrollo.com/

Objeto window

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.forms[0] === forms[0]

window.document === document

http://cursosdedesarrollo.com/

Objeto window

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.

http://cursosdedesarrollo.com/

Objeto window

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

http://cursosdedesarrollo.com/

Objeto window

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);

http://cursosdedesarrollo.com/

Objeto Window

Además de desplazar y redimensionar la ventana del navegador, es posible averiguar la posición y tamaño actual de la ventana. Sin embargo, la ausencia de un estándar para BOM provoca que cada navegador implemente su propio método:

- Internet Explorer proporciona las propiedades window.screenLeft y window.screenTop para obtener las coordenadas de la posición de la ventana. No es posible obtener el tamaño de la ventana completa, sino solamente del área visible de la página (es decir, sin barra de estado ni menús). Las propiedades que proporcionan estas dimensiones son document.body.offsetWidth y document.body.offsetHeight.

- Los navegadores de la familia Mozilla, Safari y Opera proporcionan las propiedades window.screenX y window.screenY para obtener la posición de la ventana. El tamaño de la zona visible de la ventana se obtiene mediante window.innerWidth y window.innerHeight, mientras que el tamaño total de la ventana se obtiene mediante window.outerWidth y window.outerHeight.

Control de Tiempos

Al contrario que otros lenguajes de programación, JavaScript no incorpora un método wait() que detenga la ejecución del programa durante un tiempo determinado

http://cursosdedesarrollo.com/

Control de Tiempos

Sin embargo, JavaScript proporciona los métodos setTimeout() y setInterval() que se pueden emplear para realizar tareas similares

http://cursosdedesarrollo.com/

Control de Tiempos

El método setTimeout() permite ejecutar una función al transcurrir un determinado periodo de tiempo:

```
setTimeout("console.log('Han transcurrido 3 segundos desde que me programaron')", 3000);
```

http://cursosdedesarrollo.com/

Control de Tiempos

El método setTimeout() requiere dos argumentos. El primero es el código que se va a ejecutar o una referencia a la función que se debe ejecutar

El segundo argumento es el tiempo, en milisegundos, que se espera hasta que comienza la ejecución del código

http://cursosdedesarrollo.com/

Control de Tiempos

El ejemplo anterior se puede rehacer utilizando una función:

```
function muestraMensaje() {
console.log("Han transcurrido 3 segundos desde que me programaron");
}

setTimeout(muestraMensaje, 3000);
```

http://cursosdedesarrollo.com/

Control de Tiempos

Como es habitual, cuando se indica la referencia a la función no se incluyen los paréntesis, ya que de otro modo, se ejecuta la función en el mismo instante en que se establece el intervalo de ejecución

Control de Tiempos

Cuando se establece una cuenta atrás, la función setTimeout() devuelve el identificador de esa nueva cuenta atrás

Empleando ese identificador y la función clearTimeout() es posible impedir que se ejecute el código pendiente

http://cursosdedesarrollo.com/

Control de Tiempos

```
function muestraMensaje() {

console.log("Han transcurrido 3 segundos desde que me programaron");

}

var id = setTimeout(muestraMensaje, 3000);

// Antes de que transcurran 3 segundos, se decide eliminar la ejecución pendiente

clearTimeout(id);
```

http://cursosdedesarrollo.com/

Control de Tiempos

Además de programar la ejecución futura de una función, JavaScript también permite establecer la ejecución periódica y repetitiva de una función

http://cursosdedesarrollo.com/

Control de Tiempos

El método necesario es setInterval() y su funcionamiento es idéntico al mostrado para setTimeout():

```
function muestraMensaje() {

console.log("Este mensaje se muestra cada segundo");

}

setInterval(muestraMensaje, 1000);
```

http://cursosdedesarrollo.com/

Control de Tiempos

De forma análoga a clearTimeout(), también existe un método que permite eliminar una repetición periódica y que en este caso se denomina clearInterval():

```
function muestraMensaje() {
    console.log("Este mensaje se muestra cada segundo");
}

var id = setInterval(muestraMensaje, 1000);

// Despues de ejecutarse un determinado número de veces, se elimina el intervalo

clearInterval(id);
```

http://cursosdedesarrollo.com/

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 document

Algunas de las propiedades más importantes definidas por el objeto document son:

- lastModified La fecha de la última modificación de la página
- referrer La URL desde la que se accedió a la página (es decir, la página anterior en el array history)
- title El texto de la etiqueta <title>
- URL La URL de la página actual del navegador

Objeto document

Las propiedades title y URL son de lectura y escritura, por lo que además de obtener su valor, se puede establecer de forma directa:

```
// modificar el título de la página

document.title = "Nuevo titulo";

// llevar al usuario a otra página diferente

document.URL = "http://nueva_pagina";
```


Objeto document

Además de propiedades, el objeto document contiene varios arrays con información sobre algunos elementos de la página:

- anchors Contiene todas las "anclas" de la página (los enlaces de tipo)
- applets Contiene todos los applets de la página
- embeds Contiene todos los objetos embebidos en la página mediante la etiqueta <embed>
- forms Contiene todos los formularios de la página
- images Contiene todas las imágenes de la página
- links Contiene todos los enlaces de la página (los elementos de tipo)

http://cursosdedesarrollo.com/

Objeto document

Los elementos de cada array del objeto document se pueden acceder mediante su índice numérico o mediante el nombre del elemento en la página HTML

Objeto document

<html>

<head><title>Pagina de ejemplo</title></head>

<body>

Primer parrafo de la pagina

Un enlace

<form method="post" name="consultas">

<input type="text" name="id" />

<input type="submit" value="Enviar">

</form>

</body>

</html>

http://cursosdedesarrollo.com/

Objeto document

Para acceder a los elementos de la página se pueden emplear las funciones DOM o los objetos de BOM:

- Párrafo: document.getElementsByTagName("p")
- Enlace: document.links[0]
- Imagen: document.images[0] o document.images["logotipo"]
- Formulario: document.forms[0] o document.forms["consultas"]

http://cursosdedesarrollo.com/

Objeto document

Una vez obtenida la referencia al elemento, se puede acceder al valor de sus atributos HTML utilizando las propiedades de DOM

De esta forma, el método del formulario se obtiene mediante document.forms["consultas"].method y la ruta de la imagen es document.images[0].src

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

Objeto location

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

http://cursosdedesarrollo.com/

Objeto location

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

http://cursosdedesarrollo.com/

Objeto location

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

http://cursosdedesarrollo.com/

Objeto location

Además de las propiedades de la tabla anterior, el objeto location contiene numerosos métodos y funciones

Objeto location

// 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 */

http://cursosdedesarrollo.com/

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

http://cursosdedesarrollo.com/

Objeto navigator

Aunque es uno de los objetos menos estandarizados, algunas de sus propiedades son comunes en casi todos los navegadores

Objeto navigator

- appCodeName Cadena que representa el nombre del navegador (normalmente es Mozilla)
- appNameCadena 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
- browserLanguageCadena 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")

http://cursosdedesarrollo.com/

Objeto navigator

- javaEnabled Boolean que indica si Java está habilitado
- language Cadena que representa el idioma del navegador
- mimeTypesArray 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
- plugins Array con la lista de plugins instalados en el navegador
- preference() (Sólo Firefox) Método empleado para establecer preferencias en el navegador

http://cursosdedesarrollo.com/

Objeto navigator

- product Cadena que representa el nombre del producto (normalmente, es Gecko)
- productSub Cadena que representa información adicional sobre el producto (normalmente, la versión del motor Gecko)
- securityPolicy Sólo Firefox
- systemLanguage (Sólo Internet Explorer) Cadena que representa el idioma del sistema operativo
- userAgent Cadena que representa la cadena que el navegador emplea para identificarse en los servidores
- userLanguage (Sólo Explorer) Cadena que representa el idioma del sistema operativo
- userProfile (Sólo Explorer) Objeto que permite acceder al perfil del usuario

http://cursosdedesarrollo.com/

Objeto navigator

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

Objeto screen

Las siguientes propiedades están disponibles en el objeto screen:

- availHeightAltura de pantalla disponible para las ventanas
- availWidth Anchura de pantalla disponible para las ventanas
- colorDepthProfundidad de color de la pantalla (32 bits normalmente)
- height Altura total de la pantalla en píxel
- width Anchura total de la pantalla en píxel

Objeto screen

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

Objeto screen

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 cuanto se puede redimensionar una ventana y para colocar una ventana centrada en la pantalla del usuario

http://cursosdedesarrollo.com/

Objeto screen

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);
```

http://cursosdedesarrollo.com/

Conclusiones

Hemos conocido los principales objetos disponibles en Javascript

http://cursosdedesarrollo.com/

Datos de Contacto

http://www.cursosdedesarrollo.com
info@cursosdedesarrollo.com

http://cursosdedesarrollo.com/

Licencia

David Vaquero Santiago