

JavaScript - Jerarquía de objetos - window

Objeto WINDOW

El objeto **WINDOW** representa una ventana o marco del navegador. Es el elemento raíz de la estructura de objetos de cada página HTML.

Tiene definidos los eventos: onblur, ondragdrop, onerror, onfocus, onload, onmove, onresize, onunload

Propiedades del objeto WINDOW:

Arrays de elementos contenidos en la ventana actual

`frames` Array de marcos (frames) de la ventana actual.

Otros atributos

<code>closed</code>	Vale true si esa ventana está cerrada.
<code>defaultStatus</code>	Mensaje por defecto que se muestra en la barra de estado de esa ventana.
<code>document</code>	Objeto documento asociado a la ventana.
<code>history</code>	Objeto historial de URLs visitadas.
<code>innerHeight</code>	Altura del área de contenido de la ventana (en pixels).
<code>innerWidth</code>	Ancho del área de contenido de la ventana (en pixels).
<code>length</code>	Número de marcos en la ventana (equivalente a <code>frames.length</code>).
<code>location</code>	Objeto location asociado a la ventana. Contiene información sobre el URL actual en la barra de direcciones.
<code>locationbar</code>	Barra de dirección (URL) del navegador. Contiene a su vez la propiedad 'visible' que indica si se muestra o no la barra en la ventana actual.
<code>menubar</code>	Barra de menú del navegador. Contiene la propiedad 'visible' que indica si se muestra o no la barra en la ventana actual.
<code>name</code>	Nombre de la ventana.
<code>opener</code>	Referencia al objeto Window que abrió una nueva ventana. Por ejemplo, para saber el nombre de dicha ventana: <code>nombre=window.opener.name;</code>
<code>outerHeight</code>	Altura exterior de la página en pixels.
<code>outerWidth</code>	Anchura exterior de la página en pixels.
<code>parent</code>	Representa el nombre de la ventana o marco que contiene a la ventana actual.
<code>self</code>	Representa la ventana actual.
<code>status</code>	Cadena con el mensaje de la barra de estado.
<code>statusbar</code>	Barra de estado de la ventana.

toolbar	Barra de herramientas del navegador.
top	Representa la ventana de nivel superior (raíz) en una estructura de marcos.
window	Representa la ventana actual (como self).

Resumen de métodos del objeto WINDOW:

Métodos

`alert()` Muestra una ventana de mensajes. Ejemplo:

```
If (numero>10) alert ("Fuera de rango");
```

`back()` Carga la URL anterior de la lista del objeto HISTORY correspondiente a la ventana de nivel superior:

```
window.back( );
```

`blur()` Desactiva una página.

`clearInterval(variable)` Cancela un temporizador iniciado con `setInterval`.

`clearTimeout(variable)` Cancela un temporizador iniciado con `setTimeout`

`close()` Cierra la ventana especificada (dependiendo de la seguridad implementada en cada navegador puede que sólo funcione con ventanas abiertas con el método `open`):

```
otraventana = window.open("")
otraventana.close ()
```

`confirm()` Genera un cuadro de diálogo con los botones Aceptar y Cancelar.

`find()` Busca una cadena de texto en el contenido de la ventana especificada:

```
find (cadena, mayusculas, siguiente)
```

Si 'mayusculas' es true, se lleva a cabo una búsqueda que tiene en cuenta mayúsculas y minúsculas. Si 'siguiente' es true, se busca la siguiente coincidencia. Si no se especifica 'cadena', el navegador lanza la ventana de búsqueda. La función devuelve true si se encontró la cadena.

`focus()`

Asigna el foco a un objeto.

`forward()`

Carga la siguiente URL de la lista del objeto HISTORY.

`home()`

Carga la URL definida como 'HOME' en el navegador.

`moveTo(x,y)`

Mueve la ventana a las coordenadas especificadas:

(deben estar permitidas las
ventanas emergentes)

```
nombreDeVentana.window.moveTo(25,10);
```

`moveBy(x,y)`

Mueve la ventana actual el número de pixels especificados por (x,y).

(deben estar permitidas las
ventanas emergentes)

`open()`

Abre una nueva ventana del navegador [*].

(deben estar permitidas las
ventanas emergentes)

`print()`

Imprime el contenido de la ventana

`prompt()`

Genera un cuadro de diálogo con un cuadro de texto para que el usuario introduzca valores.

`resizeBy(x,y)`

Redimensiona la ventana actual el número de pixels especificado por (x,y):

(deben estar permitidas las
ventanas emergentes)

```
nombreDeVentana.window.resizeBy (500, 500);
```

`resizeTo(x,y)`

Redimensiona la ventana a los nuevos valores de ancho y alto

(deben estar permitidas las
ventanas emergentes)

```
nombreDeVentana.window.resizeTo (400, 400);
```

`scrollBy(x,y)`

Desplaza la ventana actual el número de pixels especificado por (x,y)

(deben estar permitidas las
ventanas emergentes)

`scrollTo(x,y)`

Desplaza la ventana actual a las coordenadas especificadas por (x,y).

(deben estar permitidas las
ventanas emergentes)

`setInterval()`

Establece un temporizador cíclico que se encarga de evaluar una expresión regularmente [*].



`setTimeout()`

Establece un temporizador que evalúa una expresión una vez transcurrido el tiempo asignado [*].

`stop()`

Detiene el proceso de carga de la página actual.

El método **open** se utiliza para abrir nuevas ventanas del navegador. Los parámetros son:

```
open( URL, nombre_de_la_ventana, opciones);
```

Las opciones se especifican mediante una secuencia de pares 'parametro=valor' (comillas dobles o simples) separados por comas (sin espacios en medio):

Algunas opciones del método open()

height	Altura de la ventana (en pixels).
menubar	Si es 'yes', crea una barra de menú en la nueva ventana.
resizable	Si es 'yes', el usuario podrá redimensionar la ventana.
scrollbars	Si es 'yes', aparecerán barras de desplazamiento en la nueva ventana (cuando el documento supere las dimensiones de la misma).
status	Si es 'yes', la nueva ventana tendrá barra de estado.
toolbar	Si es 'yes', la ventana tendrá barra de herramientas.
width	Ancho de la ventana (en pixels).

Por ejemplo, para abrir una página en una nueva ventana:

```
miVentana = window.open( "", "" );  
window.open( "pagina2.html", "ventana2" );
```

Una nueva ventana con barra de menu, pero sin barra de estado, ni entrada para direcciones:

```
window.open( "pagina2.html", "ventana2", "status=no, menubar=yes" );
```

Si no se especifica URL, la ventana se abre sin cargar ninguna página. Si la ventana ya existe, no se borra la página que está cargada en ese momento.

Los métodos **setInterval** y **setTimeout** permiten establecer temporizadores en la ventana. Para establecer temporizadores cíclicos se utiliza el método setInterval. Por ejemplo, para establecer un aviso cada 5 segundos:

```
window.setInterval ( "alert ( 'Han pasado 5 segundos' )", 5000 );
```

Se puede establecer que se active el intervalo asignándole a una variable que servirá para parar el intervalo.

```
variable= window.setInterval ( "alert ( 'Han pasado 5 segundos' )", 5000 );  
window.clearInterval(variable);
```