

ANEXO I - EL OBJETO WINDOW

Se trata del objeto **más alto en la jerarquía del navegador** (`navigator` es un objeto independiente de todos en la jerarquía), pues todos los componentes de una página web están situados dentro de una ventana. El objeto `window` hace referencia a la ventana actual. Veamos a continuación sus propiedades y sus métodos.

Propiedades

- ✓ `closed`. Válida a partir de Netscape 3 en adelante y MSIE 4 en adelante. Es un booleano que nos dice si la ventana está cerrada (`closed = true`) o no (`closed = false`).
- ✓ `defaultStatus`. Cadena que contiene el texto por defecto que aparece en la barra de estado (status bar) del navegador.
- ✓ `frames`. Es un array: cada elemento de este array (`frames[0]`, `frames[1]`, ...) es uno de los `frames` que contiene la ventana. Su orden se asigna según se definen en el documento HTML.
- ✓ `history`. Se trata de un array que representa las URLs visitadas por la ventana (están almacenadas en su historial).
- ✓ `length`. Variable que nos indica cuántos `frames` tiene la ventana actual.
- ✓ `location`. Cadena con la URL de la barra de dirección.
- ✓ `name`. Contiene el nombre de la ventana, o del frame actual.
- ✓ `opener`. Es una referencia al objeto `window` que lo abrió, si la ventana fue abierta usando el método `open()` que veremos cuando estudiemos los métodos.
- ✓ `parent`. Referencia al objeto `window` que contiene el `frameset`.
- ✓ `self`. Es un nombre alternativo del `window` actual.
- ✓ `status`. `String` con el mensaje que tiene la barra de estado.
- ✓ `top`. Nombre alternativo de la ventana del nivel superior.
- ✓ `window`. Igual que `self`: nombre alternativo del objeto `window` actual.

Métodos

- ✓ `alert(mensaje)`. Muestra el mensaje 'mensaje' en un cuadro de diálogo
- ✓ `blur()`. Elimina el foco del objeto `window` actual. A partir de NS 3, IE 4.
- ✓ `clearInterval(id)`. Elimina el intervalo referenciado por 'id' (ver el método `setInterval()`, también del objeto `window`). A partir de NS 4, IE 4.
- ✓ `clearTimeout(nombre)`. Cancela el intervalo referenciado por 'nombre' (ver el método `setTimeout()`, también del objeto `window`).
- ✓ `close()`. Cierra el objeto `window` actual.
- ✓ `confirm(mensaje)`. Muestra un cuadro de diálogo con el mensaje 'mensaje' y dos botones, uno de aceptar y otro de cancelar. Devuelve true si se pulsa aceptar y devuelve false si se pulsa cancelar.
- ✓ `focus()`. Captura el foco del ratón sobre el objeto `window` actual. A partir de NS 3, IE 4.
- ✓ `moveBy(x,y)`. Mueve el objeto `window` actual el número de pixels especificados por (x,y). A partir de NS 4.
- ✓ `moveTo(x,y)`. Mueve el objeto `window` actual a las coordenadas (x,y). A partir de NS 4.
- ✓ `open(URL,nombre,características)`. Abre la URL que le pasemos como primer parámetro en una ventana de nombre 'nombre'. Si esta ventana no existe, abrirá una ventana nueva en la que mostrará el contenido con las características especificadas. Las características que podemos elegir para la ventana que queramos abrir son las siguientes:
 - ➔ `toolbar = [yes|no|1|0]`. Nos dice si la ventana tendrá barra de herramientas (yes,1) o no la tendrá (no,0).
 - ➔ `location = [yes|no|1|0]`. Nos dice si la ventana tendrá campo de localización o no.
 - ➔ `directories = [yes|no|1|0]`. Nos dice si la nueva ventana tendrá botones de dirección o no.
 - ➔ `status = [yes|no|1|0]`. Nos dice si la nueva ventana tendrá barra de estado o no.
 - ➔ `menubar = [yes|no|1|0]`. Nos dice si la nueva ventana tendrá barra de menús o no.
 - ➔ `scrollbars = [yes|no|1|0]`. Nos dice si la nueva ventana tendrá barras de desplazamiento o no.

- `resizable = [yes|no|1|0]`. Nos dice si la nueva ventana podrá ser cambiada de tamaño (con el ratón) o no.
- `width = px`. Nos dice el ancho de la ventana en pixels.
- `height = px`. Nos dice el alto de la ventana en pixels.
- `outerWidth = px`. Nos dice el ancho ***total*** de la ventana en pixels. A partir de NS 4.
- `outerHeight = px`. Nos dice el alto ***total*** de la ventana en pixels. A partir de NS 4
- `left = px`. Nos dice la distancia en pixels desde el lado izquierdo de la pantalla a la que se debe colocar la ventana.
- `top = px`. Nos dice la distancia en pixels desde el lado superior de la pantalla a la que se debe colocar la ventana.
- ✓ `prompt(mensaje, respuesta_por_defecto)`. Muestra un cuadro de diálogo que contiene una caja de texto en la cual podremos escribir una respuesta a lo que nos pregunte en '*mensaje*'. El parámetro '*respuesta_por_defecto*' es opcional, y mostrará la respuesta por defecto indicada al abrirse el cuadro de diálogo. El método retorna una cadena de caracteres con la respuesta introducida.
- ✓ `scroll(x,y)`. Desplaza el objeto `window` actual a las coordenadas especificadas por (x,y). A partir de NS3, IE4.
- ✓ `scrollBy(x,y)`. Desplaza el objeto `window` actual el número de pixels especificado por (x,y). A partir de NS4.
- ✓ `scrollTo(x,y)`. Desplaza el objeto `window` actual a las coordenadas especificadas por (x,y). A partir de NS4.
- ✓ `setInterval(expresion, tiempo)`. Evalua 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 `clearInterval()`. A partir de NS4, IE4.
- ✓ `setTimeout(expresion, tiempo)`. Evalua 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 `clearTimeout()`. A partir de NS4, IE4.

Me dejo en el tintero otras propiedades y métodos como `innerHeight`, `innerWidth`, `outerHeight`, `outerWidth`, `pageXOffset`, `pageYOffset`, `personalbar`, `scrollbars`, `back()`, `find(["cadena"], [caso, bkwd])`, `forward()`, `home()`, `print()`, `stop()`... todas ellas disponibles a partir de NS 4 y cuya explicación remito como ejercicio al lector interesado en saber más sobre el objeto `window`.

```
<HTML>
  <HEAD>
    <title>Ejemplo de JavaScript</title>
    <script LANGUAGE="JavaScript">
      <!--
        function moverVentana(){
          mi_ventana.moveBy(5,5);
          i++;
          if (i<20)
            setTimeout('moverVentana()',100);
          else
            mi_ventana.close();
        }
      //-->
    </script>
  </HEAD>
  <BODY>
    <script LANGUAGE="JavaScript">
      <!--
        var opciones="left=100,top=100,width=250,height=150", i= 0;

        mi_ventana = window.open("", "", opciones);
        mi_ventana.document.write("Una prueba de abrir ventanas");
        mi_ventana.moveTo(400,100);
        moverVentana();
      //-->
    </script>
  </BODY>
</HTML>
```

ANEXO II - EL OBJETO LOCATION

Este objeto contiene la URL actual así como algunos datos de interés respecto a esta URL. Su finalidad principal es, por una parte, modificar el objeto `location` para cambiar a una nueva URL, y extraer los componentes de dicha URL de forma separada para poder trabajar con ellos de forma individual si es el caso. Recordemos que la sintaxis de una URL era:

```
protocolo://maquina host[:puerto]/camino al recurso
```

Propiedades

- ✓ `hash`. Cadena que contiene el nombre del enlace, dentro de la URL.
- ✓ `host`. Cadena que contiene el nombre del servidor y el número del puerto, dentro de la URL.
- ✓ `hostname`. Cadena que contiene el nombre de dominio del servidor (o la dirección IP), dentro de la URL.
- ✓ `href`. Cadena que contiene la URL completa.
- ✓ `pathname`. Cadena que contiene el camino al recurso, dentro de la URL.
- ✓ `port`. Cadena que contiene el número de puerto del servidor, dentro de la URL.
- ✓ `protocol`. Cadena que contiene el protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
- ✓ `search`. Cadena que contiene la información pasada en una llamada a un script, dentro de la URL.

Métodos

- ✓ `reload()`. Vuelve a cargar la URL especificada en la propiedad `href` del objeto `location`.
- ✓ `replace(cadenaURL)`. Reemplaza el historial actual mientras carga la URL especificada en `cadenaURL`.

```
<HTML>
  <HEAD>
    <title>Ejemplo de JavaScript</title>
  </HEAD>
  <BODY>
    <script LANGUAGE="JavaScript">
      <!--
        document.write("Location <b>href</b>: " + location.href + "<br>");
        document.write("Location <b>host</b>: " + location.host + "<br>");
        document.write("Location <b>hostname</b>: " + location.hostname + "<br>");
        document.write("Location <b>pathname</b>: " + location.pathname + "<br>");
        document.write("Location <b>port</b>: " + location.port + "<br>");
        document.write("Location <b>protocol</b>: " + location.protocol + "<br>");
      //-->
    </script>
  </BODY>
</HTML>
```

ANEXO III - EL OBJETO NAVIGATOR

Este objeto simplemente nos da información relativa al navegador que esté utilizando el usuario.

Propiedades

- ✓ `appName`. Cadena que contiene el nombre del código del cliente.
- ✓ `appName`. Cadena que contiene el nombre del cliente.
- ✓ `appVersion`. Cadena que contiene información sobre la versión del cliente.
- ✓ `language`. Cadena de dos caracteres que contiene información sobre el idioma de la versión del cliente.
- ✓ `mimeType`. Array que contiene todos los tipos MIME soportados por el cliente. A partir de NS 3.
- ✓ `platform`. Cadena con la plataforma sobre la que se está ejecutando el programa cliente.
- ✓ `plugins`. Array que contiene todos los plug-ins soportados por el cliente. A partir de NS 3.
- ✓ `userAgent`. Cadena que contiene la cabecera completa del agente enviada en una petición HTTP. Contiene la información de las propiedades `appName` y `appVersion`.

Métodos

- ✓ `javaEnabled()`. Devuelve `true` si el cliente permite la utilización de Java, en caso contrario, devuelve false.

```
<HTML>
  <HEAD>
    <title>Ejemplo de JavaScript</title>
  </HEAD>
  <BODY>
    <script LANGUAGE="JavaScript">
      <!--
        document.write("Navigator <b>appName</b>: " + navigator.appCodeName + "<br>");
        document.write("Navigator <b>appName</b>: " + navigator.appName + "<br>");
        document.write("Navigator <b>appVersion</b>: " + navigator.appVersion + "<br>");
        document.write("Navigator <b>language</b>: " + navigator.language + "<br>");
        document.write("Navigator <b>platform</b>: " + navigator.platform + "<br>");
        document.write("Navigator <b>userAgent</b>: " + navigator.userAgent + "<br>");

      //-->
    </script>
  </BODY>
</HTML>
```

ANEXO IV - EL OBJETO STRING

El objeto String se utiliza para manipular una cadena almacenada de texto.
Los objetos String se crean con new String()

Sintaxis

```
var txt = new String("cadena");
```

o simplemente:

```
var txt = "cadena";
```

Para obtener más información consulte:

http://www.w3schools.com/js/js_obj_string.asp

Propiedades del objeto String

Propiedad	Descripción
<code>constructor</code>	Devuelve la función que ha creado el prototipo del objeto String
<code>length</code>	Devuelve la longitud de una cadena
<code>prototype</code>	Te permite añadir propiedades y métodos a un objeto

Métodos del objeto String

Método	Descripción
<code>charAt()</code>	Devuelve el carácter en el índice especificado
<code>charCodeAt()</code>	Devuelve el carácter Unicode del índice especificado
<code>concat()</code>	Une dos o más cadenas y devuelve una copia de las cadenas unidas
<code>fromCharCode()</code>	Convierte valores Unicode a caracteres
<code>indexOf()</code>	Devuelve la posición de la primera aparición de un valor especificado en una cadena
<code>lastIndexOf()</code>	Devuelve la posición de la última aparición de un valor especificado en una cadena
<code>match()</code>	Busca una coincidencia entre una expresión regular y una cadena, y devuelve las coincidencias
<code>replace()</code>	Busca una coincidencia entre una subcadena (o expresión regular) y una cadena, y sustituye a la subcadena encontrada con una nueva subcadena
<code>search()</code>	Busca una coincidencia entre una expresión regular y una cadena, y devuelve la posición de la coincidencia
<code>slice()</code>	Extrae una parte de una cadena y devuelve una nueva cadena
<code>split()</code>	Divide una cadena dentro de un array de subcadenas
<code>substr()</code>	Extrae los caracteres de una cadena, empezando en la posición de inicio especificado, y el número especificado de caracteres
<code>substring()</code>	Extrae los caracteres de una cadena, entre dos índices especificados
<code>toLowerCase()</code>	Convierte una cadena a minúsculas
<code>toUpperCase()</code>	Convierte una cadena a mayúsculas
<code>valueOf()</code>	Devuelve el valor primitivo de un objeto String

Métodos envoltantes de String HTML

Los métodos envoltantes HTML devuelven la cadena envuelta en una etiqueta HTML apropiada

Method	Description
<code>anchor()</code>	Crea un <code>anchor</code> (ancla)
<code>big()</code>	Visualiza una cadena usando una fuente grande
<code>blink()</code>	Visualiza una cadena parpadeante
<code>bold()</code>	Visualiza una cadena en negrita

<code>fixed()</code>	Visualiza una cadena utilizando una fuente de paso fijo
<code>fontcolor()</code>	Visualiza una cadena usando el color especificado
<code>fontsize()</code>	Visualiza una cadena usando el tamaño especificado
<code>italics()</code>	Visualiza una cadena en cursiva
<code>link()</code>	Visualiza una cadena como un vínculo
<code>small()</code>	Visualiza una cadena en letra pequeña
<code>strike()</code>	Visualiza una cadena con un tachado
<code>sub()</code>	Visualiza una cadena como texto subíndice
<code>sup()</code>	Visualiza una cadena como texto superíndice