

Objetos del cliente Objetos del BOM

Tema 5 parte 3.1

Introducción objetos BOM

Objeto WINDOW

- El Browser Object Model (BOM) permite representar un documento html desde el punto de vista del navegador.

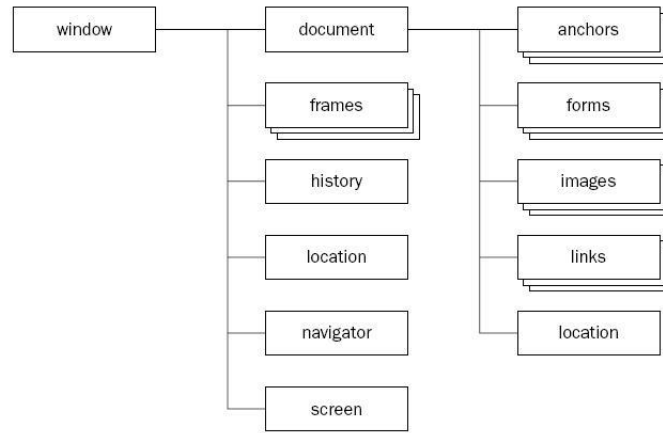


- Permite **interactuar con las ventanas** del navegador mediante JavaScript.
- Crear, mover, redimensionar y cerrar ventanas de navegador.
- Obtener **información** sobre el propio navegador.
- **Propiedades** de la página actual y de la pantalla del usuario.
- **Obtener colecciones** de elementos (formularios, imágenes, frames, etc.).

Objetos del cliente

- Cuando un cliente carga un documento HTML, crea los objetos correspondientes al documento, a su contenido y a la información útil.
- Cada documento HTML crea los objetos siguientes:

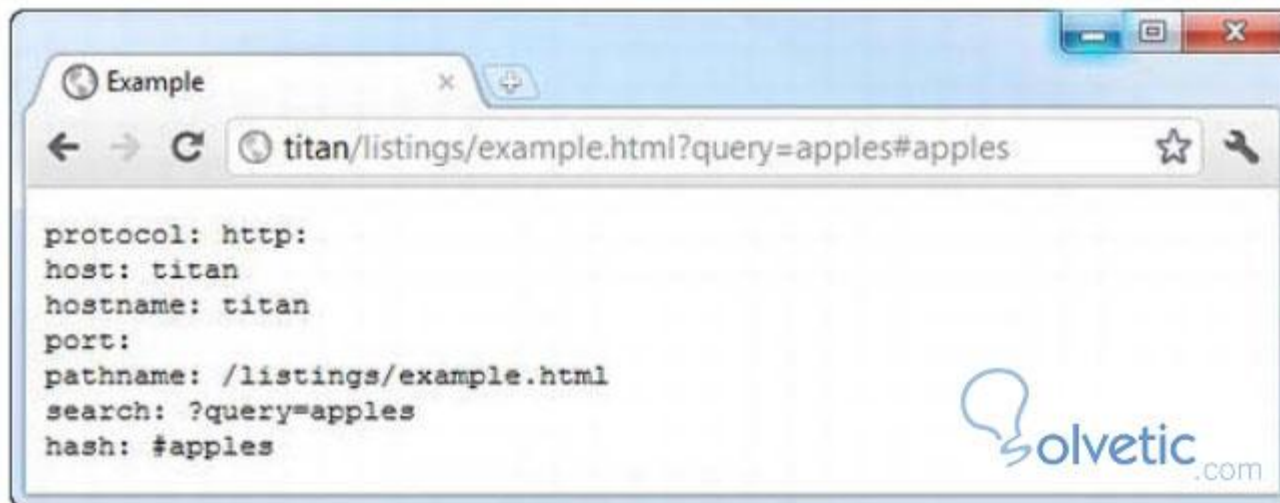
Objeto Window



- **window.** Representa la raíz del árbol, y posee descendientes cuando el documento contiene subventanas.

Objeto location

- **location**. Contiene las propiedades de la dirección URL del documento



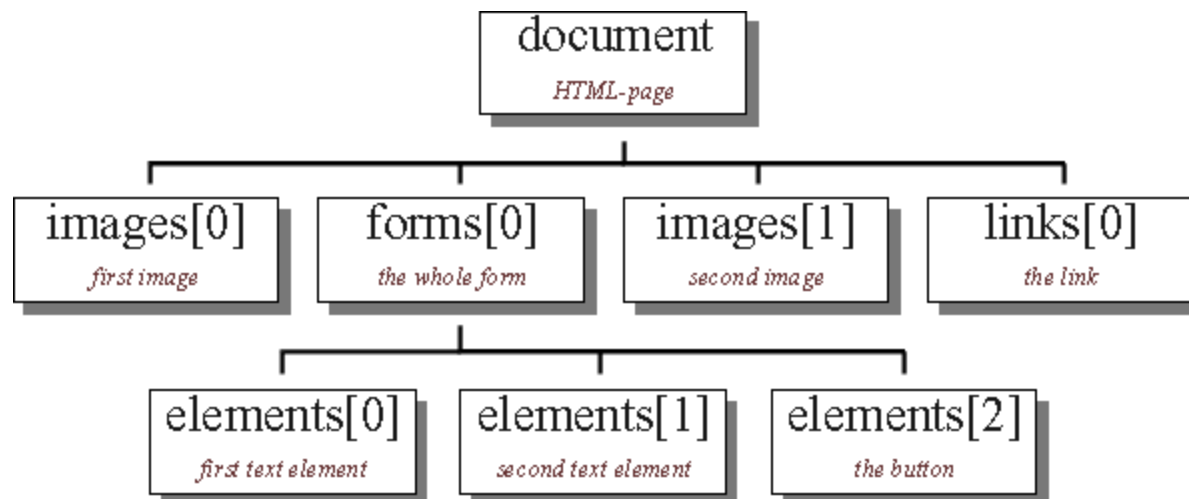


Objeto history

- **history.** Contiene la información relativa a las direcciones URL visitadas en la sesión actual.

Objeto document

- **document.** Contiene información sobre el documento actual como el título, los formularios, colecciones...





Objeto Screen



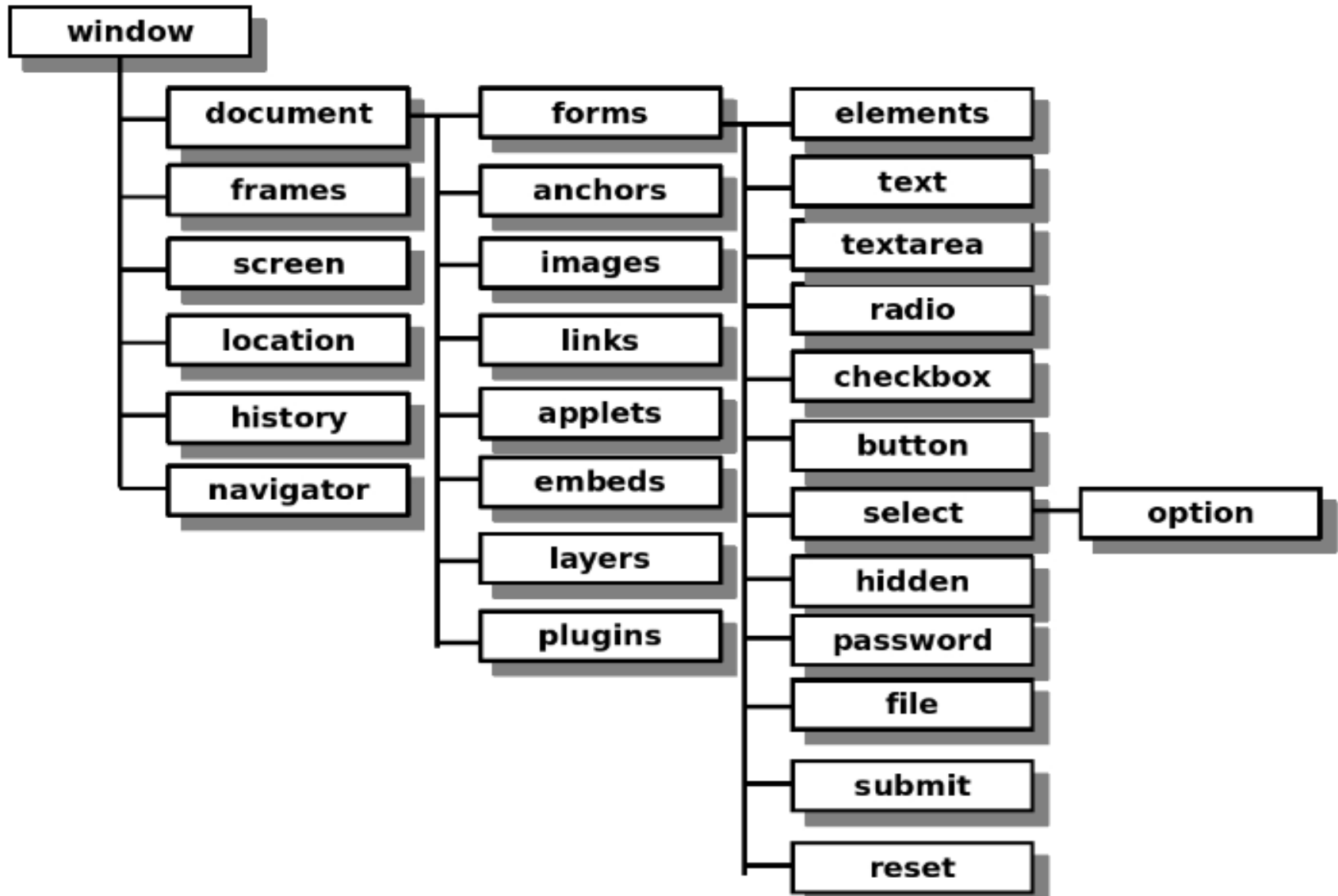
- **Screen:** El objeto screen se utiliza para obtener información sobre la pantalla del usuario. Uno de los datos más importantes que proporciona el objeto screen es la **resolución** del monitor en el que se están visualizando las páginas. Los diseñadores de páginas web necesitan conocer las resoluciones más utilizadas por los usuarios para adaptar sus diseños a esas resoluciones.

Objeto navigator

- **Navigator:** Permite obtener información sobre el propio navegador.



Objetos del cliente



Objetos del cliente-Window

- **Objeto window**
 - Representa la ventana del navegador



Propiedades	Métodos
closed, defaultStatus, frames, history, length, location, name, parent, opener, self, status, top	alert(mensajes), open(), close(), confirm(mensaje), prompt(), moveBy(x,y), moveTo(x,y), print(), setTimeout(), resizeBy(x,y), resizeTo(ancho,alto), scroll(x,y), scrollBy(x,y), scrollTo(x,y), clearInterval(), setInterval(), setTimeout()

Objetos del cliente-Window-Método Open

- Window—método open

Open(URL, nombre, características): Abre la url que le pasamos como primer parámetro en la ventana cuyo nombre se indica en el segundo parámetro. Si no existe esa ventana, abre otra nueva con las características que se le pasen por parámetro:

Características	Valores	Nos indica
tollbar	Yes, no, 1, 0	Si tendrá barra de herramientas
Location	Yes, no, 1, 0	Si tendrá campo de localización
Directories	Yes, no, 1, 0	Si tendrá botones de direcciones
Status	Yes, no, 1, 0	Si tendrá barra de estado
Menubar	Yes, no, 1, 0	Si tendrá barra de menús
Scrollbars	Yes, no, 1,0	Si tendrá barra de desplazamiento
Resizable	Yes, no, 1, 0	Si se podrá cambiar de tamaño con el ratón
Width	Nº pixels	Ancho de la ventana
Height	Nº pixels	Alto de la ventana
Left	Nº pixels	Distancia a la izquierda
Top	Nº pixels	Distancia hacia arriba

Objetos del cliente-Window-Método Open

Apertura de una ventana

- Abrir una ventana a través de JavaScript es tan simple como pasar a una instrucción una dirección URL para que la localice y la cargue:

nuevaVentana=window.open("dirección URL");



Objetos del cliente-Window-Método Open

```
variable=window.open("dirección URL",  
"Ventana de destino","parámetros de  
apertura" );
```

En ventana destino:

_blank ->se carga en una nueva pestaña

_self→reemplaza la página actual

Name: nombre de la ventana

Objetos del cliente-Window-Método close



Cerrar ventanas

Para cerrar una ventana se utiliza el método **close()** de las mismas:

variable_de_ventana.close();

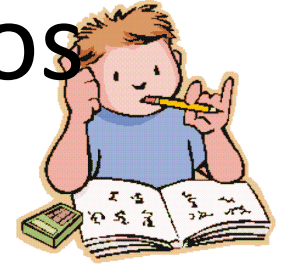
Si la ventana a cerrar es la propia, entonces tendremos que usar:

window.close();

close();

- Ha de tenerse cuidado al cerrar una ventana dentro de un elemento de un formulario ya que si es la propia ventana la que queremos cerrar habremos de especificar **window.close();** y no solo **close();**
- Esto es así porque dentro de los formularios el objeto por defecto es *document* no *window*, por ello, *close()* solamente cerraría el documento, no la ventana.

Objetos del cliente-Ejercicios



Probando:

- 1-Abrir una nueva ventana con barra de herramientas, sin posibilidad de escribir una dirección y que no sea redimensionable. En ella vamos a cargar la página "xxxxx" y la vamos a llamar "segundaPag"; la altura será de 200 pixeles.
- 2-abra una nueva ventana con esta dirección www.torrelavega.es, se tendrá que abrir en una nueva pestaña. Asociar características.
- 3-Retocar el ejercicio anterior para dar la opción de cerrar la ventana desde la ventana principal (usar confirm).

Objetos del cliente-WINDOW-métodos de manipulación

moveBy(x,y)

Desplaza la ventana desde su punto actual según el número de píxeles definidos en los parámetros. Para x los valores positivos mueven la ventana a la derecha y los negativos a la izquierda; para y, los positivos la mueven hacia abajo y los negativos hacia arriba.

moveTo(x,y)

Desplaza la ventana hasta las coordenadas indicadas en los parámetros, tomando como referencia el punto 0,0 de la pantalla del usuario.

resizeBy(x,y)

Modifica el tamaño de la ventana en la cantidad —positiva o negativa— de píxeles definidos por los parámetros, donde x es la anchura e y la altura

resizeTo(x,y)

Modifica el tamaño de la ventana hasta el definido en los parámetros, donde x es la anchura e y la altura. No acepta valores inferiores a 100.

Objetos del cliente-window-métodos

setInterval-setTimeout

Estos dos métodos permiten ejecutar una función pasado un cierto intervalo de tiempo.

La **única diferencia** entre ellas es que "setInterval" se ejecutará una y otra vez en intervalos de x segundos,

en cambio, setTimeout se ejecutará una sola vez pasados x segundos.

Utilización Temporizadores

- **setTimeout**(*código_JavaScript, milisegundos*)

Retarda la ejecución del *código_JavaScript* el tiempo indicado por *milisegundos*.
Devuelve un identificador que representa el retardo.
Se puede cancelar la ejecución mediante el método

- **clearTimeout**(*identificador*)

Identificador es el valor devuelto por `setTimeout`.

- **setInterval**(*código_JavaScript, milisegundos*)

Ejecuta *código_JavaScript* cada vez con el intervalo puesto en *milisegundos*.
Al igual que `setTimeout`, devuelve un identificador que se puede utilizar para cancelar el retardo mediante

- **clearInterval**(*identificador*).



EJEMPLO PRACTICO

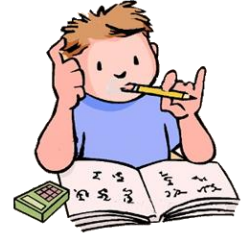
- ```
var contar =0;
function contar(){
 contar++;
}
tempo = setInterval("contar()", 1000);
function parar(){
 clearInterval(tempo);
 alert("Has tardado "+contar+" segundos");
}
```
- En la página podríamos colocar un botón que llame a esta función parar
- Al pulsar el botón se detiene el contador y aparece un mensaje indicando el tiempo que tardó en pulsar el botón, por supuesto es un valor aproximado.



# OBJETOS DEL CLIENTE , MAS PROPIEDADES DEL OBJ WINDOW

|                    |                                                                                                                                                                      |   |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|                    |                                                                                                                                                                      |   |
| <b>innerHeight</b> | Devuelve la altura en píxeles del área de contenidos (viewport) del navegador incluyendo scrollbar si existe.                                                        |   |
| <b>innerWidth</b>  | Devuelve la anchura en píxeles del área de contenidos (viewport) del navegador incluyendo scrollbar si existe.                                                       |   |
| <b>outerHeight</b> | Devuelve la altura en píxeles de la ventana completa hasta el límite exterior de sus bordes (quedando dentro de esta medida barras de menú, barras de estado, etc.). |   |
| <b>outerWidth</b>  | Devuelve el ancho en píxeles de la ventana completa hasta el límite exterior de sus bordes (quedando dentro de esta medida barras de menú, barras de estado, etc.).  | 2 |
|                    |                                                                                                                                                                      |   |

# Ejercicio 1

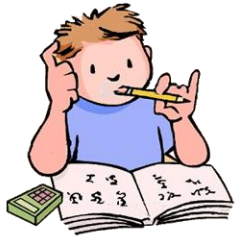


Script que abra una ventana con ancho de 300px y alto 200px.

La ventana se desplazará de forma aleatoria durante un nº finito de veces.

(probar con `moveTo()` y con `moveBy()`)

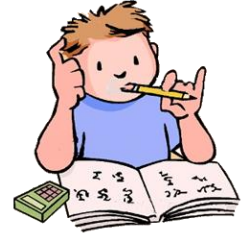




## Ejercicio2

- **Realizar un script que abra una nueva ventana de pequeño tamaño y que esta Se caiga hasta abajo , durante la caida en la ventana pondrá: “ me caigo” y cuando llegue abajo se cerrará y aparecerá en el documento “llegO”.**

# Ejercicio 3



- Realizar un script que nos pida confirmación para abrir una nueva ventana, si es que si nos activa una barra de progreso y cuando se llene abrimos la ventana.

# Ejercicio 4

Realizar un cronómetro , a toda pantalla :

Formato:

00:00:00:00

Con las siguientes opciones:

Start

Stop

Resume

Reset