BOM - Objetos navegador. Comunicación

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Nov

Carmelo José Jaén Díaz

- iPasen - cjaedia071@g.educaand.es

Índice

OBJETIVO

- Identificar el concepto de BOM con los objetos que lo componen y saber utilizarlo en el desarrollo de aplicaciones web.

GLOSARIO

BOM (Browser Object Model). Objeto en el que se representa el navegador, no solo el documento. Mediante el BOM se puede acceder a datos como el historial de navegación, dimensiones de la ventana, etcétera.

DOM. Plataforma e interfaz de un documento que permite a los scripts y programas acceder y modificar dinámicamente su contenido, estructura y estilo.

Tag. Término en inglés que significa “etiqueta” y hace referencia a una palabra clave que sirve para describir un documento.

INTRODUCCIÓN

En esta sección, vamos a ver cómo podemos comunicarnos con sub-ventanas, que abrimos empleando el método ***open()*** del objeto **window**.

Cada objeto **window** tiene unas propiedades, que nos permiten interactuar con `iframes` y con otras ventanas que nos van a resultar muy útiles para poder conocer el estado de la ventana, quién la creó, etc.

PROPIEDADES CON OTRAS VENTANAS

Esta serie de **propiedades** son:

- ***frames***: devuelve todos los iframe de la ventana.
- ***frameElement***: devuelve el frame en el que está insertada la ventana.
- ***length***: devuelve el número de frames que tiene la ventana.
- ***closed***: devuelve un booleano que indica si la ventana está cerrada.
    - No la ventana principal, sino otra que se haya creado y se pueda acceder a ella a través de su identificador.
- ***opener***: devuelve la referencia de la ventana que creó la ventana actual.
- ***parent***: devuelve la ventana padre de la actual.
- ***self***: devuelve la ventana actual

PROPIEDADES CON OTRAS VENTANAS

opener

Como hemos comentado cada objeto **window** tiene unas propiedades, entre las cuales se encuentra **opener**. Esta propiedad contiene la referencia a la ventana o marco, que ha abierto ese objeto **window** empleando el método ***open()***. Para la ventana principal el valor de **opener** será `null`.

Debido a que **opener** es una referencia válida a la ventana padre que abrió las otras, podemos emplearlo para iniciar la referencia a objetos de la ventana original (padre) desde la ventana hija.

Veamos un ejemplo de uso.

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

A continuación, mediante un ejemplo guiado se pone en práctica los métodos y propiedades citadas para facilitar su comprensión.

En primer lugar, vamos a crear la página web correspondiente a la ventana principal. Esta implementará:

- Dos botones, uno para abrir una ventana secundaria y otro para cerrar la misma.

```
<button onclick="abrirSubVentana()">Abrir sub ventana</button>
<button onclick="cerrarSubVentana()">Cerrar sub ventana</button>
//Etiqueta párrafo por estética rápida. Implementar con estilos CSS
<p>
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

- Un formulario donde se recibirá el dato de la ventana secundaria en uno de sus campos.

```
<form  action="">

   <label>Texto proveniente de la sub-ventana:</label>
   <input  type="text"  id="original">

</form>
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

En segundo lugar, vamos a crear la página web correspondiente a la ventana secundaria. Esta implementará:

- Un formulario donde se introducirá el dato a enviar a la ventana principal.

```
<form  id="formulario">

   <label  for="textocopiar">Introduzca texto a copiar en la ventana
   principal:</label>
   <input  type="text"  id="textocopiar"/>

</form>

//Etiqueta párrafo por estética rápida. Implementar con estilos CSS
<p>
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

- Un botón, para enviar el dato introducido en el campo del formulario a la ventana principal.

```
<button onclick="enviarAprincipal()">Enviar texto a la ventana principal</button>
```

Por último, implementaremos la lógica de los botones en un mismo fichero JavaScript que vincularemos a ambos documentos HTML.

- *abrirSubVentana()*
- *cerrarSubVentana()*
- *enviarAprincipal()*

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

- *abrirSubVentana()*

Abrimos una nueva ventana mediante el método *open()* del objeto **window** pasándole como argumentos la URL de la ventana secundaria y un tamaño en píxeles para la anchura y altura. Asignamos el método a una variable nuevaVentana.

```
function abrirSubVentana() {
    let nuevaVentana = window.open("secundaria.html", "sub",
"height=300,width=400");
}
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

- *cerrarSubVentana()*

  Comprobamos mediante un `if` si existe una nueva ventana creada, es decir si se ha pulsado el botón "Abrir sub ventana".

  - En caso afirmativo, la cerramos con el método *close()* del objeto **window**.
  - En caso contrario, no hacemos nada.

```
function cerrarSubVentana() {
    if (nuevaVentana) {
        nuevaVentana.close();
    }
}
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

- *enviarAprincipal()*

    - Primero: obtenemos la referencia a la ventana principal (ventana padre que abrió la ventana secundaria) mediante la propiedad **opener**.
    - Segundo: Recuperamos el valor del elemento con id = “original” del DOM mediante el método getElementById().
    - Tercero: como se trata de un dato introducido en un campo de texto de un formulario, utilizamos la propiedad **value** para devolver ese dato.
    - Cuarto: asignamos el dato introducido en el campo del formulario de la ventana secundaria a lo anteriormente citado de manera análoga.

```
function enviarAprincipal() {
    opener.document.getElementById("original").value =
document.getElementById('textocopiar').value;
}
```

COMUNICACIÓN ENTRE VENTANAS
Ejemplo guiado

La salida del código anterior puede verse en la imagen adjunta.