

Unidad 2. Modelo de objetos predefinidos en JavaScript.

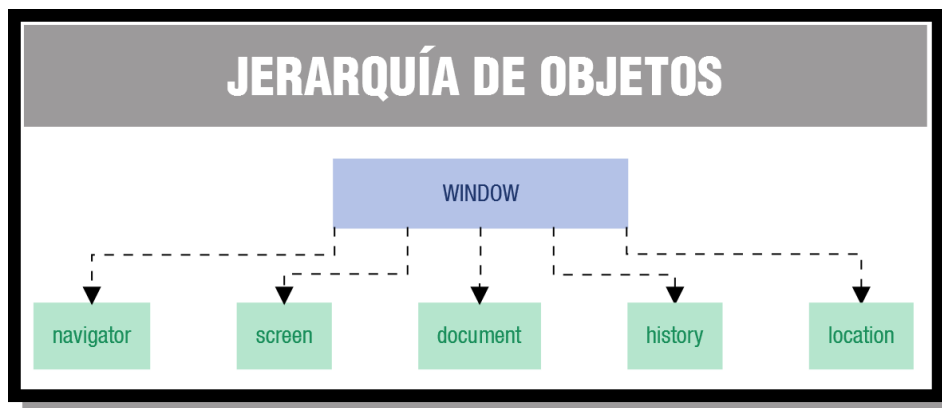
Boletín de actividades

La unidad cuenta con 3 apartados principales:

1. El modelo de objetos del navegador.
2. Los marcos.
3. Los objetos nativos de JavaScript.

1. El modelo de objetos del navegador.

El BOM (Browser Object Model o Modelo de Objetos del Navegador) es un conjunto de objetos creados por el navegador y que se pueden consultar y modificar desde JavaScript. Este esquema muestra los objetos que forman parte del BOM:



Hay que tener en cuenta que no existe una especificación común para el BOM y su implementación depende del navegador que se esté utilizando. Puede haber por tanto, métodos, propiedades o comportamientos que varíen de un navegador a otro.

Objeto Window

Window representa la ventana del navegador. Cada pestaña también se representa como un objeto de este tipo y si una página contiene frames cada uno de ellos también cuenta con su propio objeto Window.

Fuentes a tener en cuenta:

- [W3Schools](#)
- [MDN](#)
- [codexejemplar](#)

Actividad 1. Consultar propiedades.

Realiza una aplicación que consulte los siguientes valores de la ventana actual del navegador:

- La distancia a la que se encuentra la ventana con respecto a la pantalla desde el lado izquierdo y superior. La consulta también debe ser válida para versiones de IE anteriores a la 9. (Propiedades: `screenX` y `screenY`)
- La altura y anchura de la ventana sin incluir las barras del navegador. La consulta también debe ser válida para versiones de IE anteriores a la 9. (Propiedades: `innerHeight` e `innerWidth`)
- La altura y anchura de la ventana incluyendo las barras del navegador. (Propiedades: `outerHeight` y `outerWidth`)

Actividad 2. Abrir y cerrar.

Desarrolla una aplicación que al pulsar un botón abra una nueva ventana con las siguientes propiedades:

- 600px de ancho y 450px de alto (Propiedades: `width` y `height`)
- 50px de distancia al borde superior y 50px al borde izquierdo (Propiedades: `top` y `left`)
- El menú del navegador sea visible (propiedad `menubar`)
- Se pueda redimensionar (propiedad `resizable`)
- Se muestre la barra de direcciones (propiedad `location`)
- Aparezcan las barras de desplazamiento (propiedad `scrollbars`)
- Sea visible la barra de estado (propiedad `status`)
- Se muestre la barra de herramientas (propiedad `toolbar`)

A continuación, incluye un botón en la nueva ventana que al pulsarlo haga que ésta se cierre.

Actividad 3. Mover y redimensionar.

Los métodos principales para mover y redimensionar ventanas son: `moveBy`, `moveTo`, `resizeBy` y `resizeTo`.

Crea una aplicación que cuente con botones para realizar estas acciones:

- Mover la ventana 50px a la derecha.
- Mover la ventana a las coordenadas 50,50 de la pantalla.
- Aumentar el ancho de la ventana 50 píxeles.
- Establecer el tamaño de la ventana a 600px x 450px.

Nota: En algunos navegadores los métodos de mover y redimensionar ventanas sólo funcionan cuando se aplican sobre una ventana abierta previamente por medio de JavaScript.

Actividad 4. Temporizadores.

El objeto `Window` posee varios métodos relacionados con el tiempo. Su manejo es fundamental en la programación de juegos, animaciones y otras aplicaciones. Hay dos métodos principales que nos permiten ejecutar un determinado código cuando pasa un cierto tiempo: `setTimeout` y `setInterval`.

Para probar estas funciones desarrolla estas dos tareas:

- a) Crea una aplicación que lance una nueva ventana emergente (`alert`) 5 segundos después de abrir la aplicación.
- b) Crea una nueva aplicación que abra una nueva ventana y ponga en marcha una cuenta atrás de 10 segundos (por consola). Al llegar a 0 la nueva ventana debe cerrarse.

Puedes consultar los [ejemplos de codexemplar](#).

Objeto Location

Location contiene información referente a la URL actual.

Fuentes a tener en cuenta:

- [W3Schools](#)
- [MDN](#)
- [codexexempla](#)

Actividad 1. Consultar propiedades.

Ejecuta el siguiente ejemplo e indica qué información se obtiene de cada una de las propiedades mostradas por consola.

```
// Se ha creado un elemento <a> y se ha usado la propiedad href
para el propósito de este ejemplo.
// Una alternativa mas correcta es navegar a la URL y usar
document.location o window.location
var url = document.createElement('a');
url.href = 'https://developer.mozilla.org:8080/en-US/search?
q=URL#search-results-close-container';
console.log(url.href);
console.log(url.protocol);
console.log(url.host);
console.log(url.hostname);
console.log(url.port);
console.log(url.pathname);
console.log(url.search);
console.log(url.hash);
console.log(url.origin);
```

Actividad 2. Recarga y redirección.

Desarrolla una aplicación que:

1. Asigne una nueva dirección a `location` provocando una redirección (`assign`).
2. Recargue el documento actual, tomando los datos del servidor y no de la caché del navegador (`reload`).
3. Asigne una nueva dirección a `location` provocando una redirección (`replace`),

¿Qué diferencia al método `replace` del método `assign`? ¿cuál es más recomendable utilizar?

Objeto Navigator

Navigator contiene información sobre el propio navegador.

Fuentes a tener en cuenta:

- [W3Schools](#)
- [MDN](#)
- [codexexempla](#)

Actividad 1. Consultar propiedades.

Desarrolla una aplicación que consulte el valor de las propiedades `appName`, `appVersion`, `platform`, `plugins` y `userAgent` del objeto Navigator.

Usa también el método `javaEnabled()` para comprobar si el navegador tiene Java habilitado.

Abre la aplicación con diferentes navegadores y observa qué información se obtiene con cada propiedad.

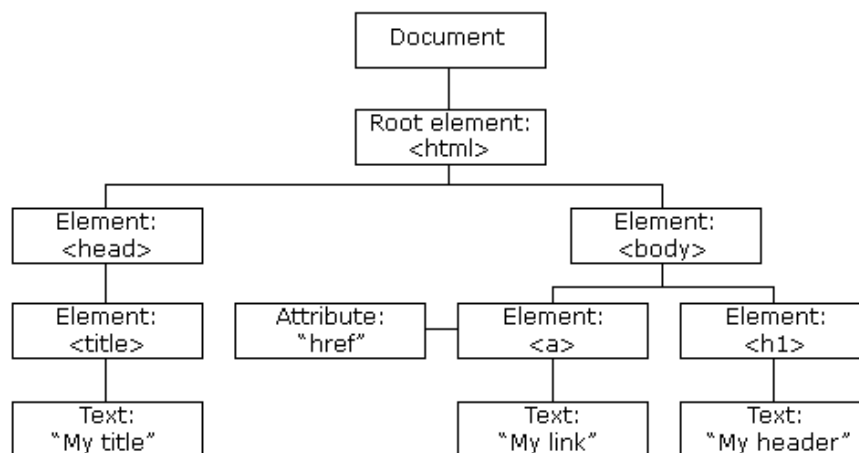
Objeto Document

El objeto `Document` representa el documento HTML cargado en el navegador.

El DOM permite a los desarrolladores ver una página HTML de otra manera, como un conjunto de objetos que JavaScript puede consultar y modificar. Por su importancia el DOM se estudiará en detalle en una unidad posterior.

Fuentes a tener en cuenta:

- [W3Schools](#)
- [MDN](#)
- [codexemplar](#)



Actividad 1. Recuperar elementos.

Los principales métodos para recuperar elementos de un documento HTML son: `getElementById`, `getElementsByTagName`, `getElementsByClassName` y `querySelectorAll`.

Crea un nuevo documento HTML con 4 párrafos (elementos `p`). Recupera el segundo párrafo utilizando los 4 métodos.

Actividad 2. Cambiar contenido.

Las principales propiedades para cambiar el contenido de elementos HTML son: `innerHTML` y `textContent`.

Cambia el contenido del segundo párrafo por “Éste es el nuevo contenido del segundo párrafo” y el contenido del tercero por “Éste es el nuevo contenido del ``tercer párrafo``.”

¿Cuál es la diferencia entre `innerHTML` y `textContent`?

Actividad 3. Añadir y eliminar elementos.

Los principales métodos para añadir y eliminar elementos HTML son: `createElement` y `remove`.

Elimina el cuarto párrafo del documento. Crea un nuevo párrafo y añádelo en último lugar. Crea un nuevo párrafo y añádelo en tercer lugar.

Actividad 4. Otras propiedades y métodos.

- a) Cambia el contenido de un párrafo por el contenido del título del documento.
- b) Añade una imagen a un documento y cámbiala desde JavaScript.
- c) Incluye un campo de texto de un formulario en un documento y cambia su valor por “Modificado desde JavaScript”.
- d) Utiliza el método `write` para escribir el contenido “Nuevo contenido escrito con write”. ¿qué ocurre con el contenido previo de la página?
- e) Realiza las siguientes operaciones sobre un párrafo:
 - Cambia a rojo el color del texto.
 - Añádele un fondo amarillo.
 - Cambia el tamaño del texto a 40 píxeles.
 - Ocúltalo.

2. Marcos

Para crear marcos en HTML5 se utiliza la etiqueta `iframe`.

Nota: La etiqueta `frame` no está soportada por HTML5. Esta etiqueta se usaba dentro de la etiqueta `frameset` y no directamente en el body como `iframe`. Puedes ver un ejemplo de uso [aquí](#).

Referencias:

- Apartado 2.1. *Jerarquías* y 2.2. *Comunicación entre marcos*.
- Artículos de `frame` e `iframe` en W3Schools y MDN.

Actividad 1. Uso de marcos.

¿Por qué actualmente se desaconseja el uso de marcos en una página web?

Actividad 2. Creación de marcos.

Crea una página web en HTML5 con dos marcos para mostrar dos páginas web externas (también deben crearse). La primera debe contener un elemento `p` con identificador “pagina1” y texto “Soy la Página 1 cargada en el marco 1”. La segunda debe contener un elemento `p` con identificador “pagina2” y texto “Soy la Página 2 cargada en el marco 2”.

Actividad 3. Comunicación entre marcos.

Habiendo realizado la actividad anterior, incorpora un botón en la página 1 que cambie el contenido del párrafo de la página 2 a “Contenido cambiado desde el marco 1”.

Posteriormente, incorpora un botón en la página 2 que cambie el contenido del párrafo de la página 1 a “Contenido cambiado desde el marco 2”.

3. Objetos nativos de JavaScript

Objeto String

Actividad 1. Propiedades y métodos.

Utiliza las propiedades y métodos del objeto `String` para realizar las siguientes tareas sobre la cadena "I have become comfortably brilliant" (nota que hay 4 espacios en blanco entre las últimas dos palabras). Sigue el índice de [W3Schools](https://www.w3schools.com/js/js_string_methods.asp):

1. Indica el número total de caracteres incluyendo los espacios en blanco.
2. Obtén el carácter que ocupa la octava posición.
3. Obtén el código Unicode del primer carácter.
4. Concatena la cadena con la cadena " and exciting". ¿Se modifica la primera cadena?
5. Comprueba si la cadena termina con los caracteres "brilliant".
6. Convierte el valor Unicode 65 a su carácter equivalente.
7. Comprueba si la cadena contiene los caracteres "comfortably".
8. Indica la posición que ocupa el primer carácter "a" de la cadena.
9. Indica la posición que ocupa el último carácter "a" de la cadena.
10. Compara la cadena con la cadena "You have become comfortably brilliant". ¿Cuál iría en primer lugar?
11. Obtén todas las coincidencias de la cadena con la expresión regular "/com/g".
12. Obtén una nueva cadena con 3 repeticiones de la cadena actual.
13. Reemplaza los caracteres "brilliant" por "exciting".
14. Busca los caracteres "brit" en la cadena.
15. Obtén de la cadena los caracteres del primero al quinto.
16. Obtén un array con todas las palabras de la cadena.
17. Comprueba si la cadena comienza con los caracteres "I have".
18. Obtén siete caracteres de la cadena a partir del segundo carácter.
19. Obtén todos los caracteres de la cadena a partir del cuarto carácter.
20. Convierte todos los caracteres de la cadena a mayúsculas.
21. Convierte todos los caracteres de la cadena a minúsculas.
22. Deja la cadena con un único espacio en blanco entre palabras.

Actividad 2. Cifrado.

Crea una aplicación que pida al usuario un mensaje y una clave numérica y a continuación use el alfabeto Unicode para codificar el mensaje desplazando cada carácter un número de veces igual al número de corresponda en la clave. Mira el siguiente ejemplo:

Mensaje: Este es el mensaje
Clave: 12345
Mensaje cifrado: Fuwi%fu#iq!ohrxblh

El mensaje cifrado se forma de esta manera:

- $E + 1 = F$
- $s + 2 = u$
- $t + 3 = w$
- $e + 4 = i$
- $(\text{espacio en blanco}) + 5 = \%$
- $e + 1 = f$
- $s + 2 = u$
- etc.

Nota: Utiliza las siguientes propiedades y funciones del objeto `String`: `charCodeAt`, `charAt`, `fromCharCode`, `length`.

Actividad 3. Validación de credenciales.

Crea una aplicación que pida al usuario un nombre y una contraseña y valide los datos recibidos. Ni la forma de pedir los datos (`prompt`) ni la validación son los métodos más ideales pero nos permitirá practicar con las funciones y propiedades del objeto `String`.

- El nombre de usuario sólo puede estar formado por letras mayúsculas y números y un máximo de 10 caracteres, de otro modo indicaremos el error y pediremos de nuevo el nombre.
- Validaremos en la contraseña que tenga al menos un elemento de los siguientes:
 - Una letra mayúscula.
 - Una letra minúscula.
 - Un número.
 - Un carácter que no sea ni letra ni número.

Si la contraseña no cumple lo indicado, pediremos al usuario que la indique de nuevo. Cuando lo cumpla, indicaremos al usuario que se ha almacenado correctamente la información.

Objeto Math

Actividad 1. Propiedades y métodos.

Utiliza las propiedades y métodos del objeto `Math` para realizar las siguientes tareas. Sigue el índice de [W3Schools](#):

1. Obtén el resultado de multiplicar el número `PI` por el número `E`.
2. Obtén el valor absoluto de `-9,87`.
3. Redondea `1,2` al entero más alto.
4. Redondea `1,8` al entero más bajo.
5. Redondea `2.51` al entero más próximo.
6. Obtén el mayor número de los siguientes `6, 7, 4, 10, 10.1`
7. Obtén el menor número de los siguientes `6, 7, 4, 3.9, 10`
8. Obtén un valor aleatorio entre `100` y `200` (Puedes consultar [MDN](#))
9. Obtén el valor de `2` elevado a `16`.
10. Obtén el valor de la raíz cuadrada de `16`.

Actividad 3. Círculo.

Implementa una aplicación que calcule la circunferencia y el área de un círculo a partir de su radio.

Actividad 2. Lotería primitiva.

Crea una aplicación que muestre 20 combinaciones para jugar a la lotería primitiva. Las combinaciones son de 6 números del 1 al 49, ten en cuenta que los números no se pueden repetir.

Objeto Number

Actividad 1. Propiedades y métodos.

Utiliza las propiedades y métodos del objeto `Number` para realizar las siguientes tareas sobre el número 123,123. Sigue el índice de [W3Schools](https://www.w3schools.com/js/js_numbers.asp):

1. Comprueba si es un número.
2. Comprueba si el número es un entero.
3. Convierte el número a una cadena.
4. Formatea el número para que solo tenga un número decimal.

Objeto Date.

Actividad 1. Propiedades y métodos.

Obtén la fecha actual. A continuación muestra los siguientes datos sobre la fecha:

1. El día del mes.
2. El día de la semana.
3. El año.
4. La hora.
5. Los milisegundos.
6. Los minutos.
7. El mes del año.
8. Los segundos.
9. Una cadena legible de la fecha.
10. Una cadena legible de la fecha en formato local.
11. Una cadena legible de la hora en formato local.
12. Una cadena legible de la fecha y la hora en formato local.

Actividad 2. Tiempo para el mundial.

Crea una aplicación que muestre por consola los años y meses que quedan desde hoy para que empiece el siguiente mundial de fútbol.