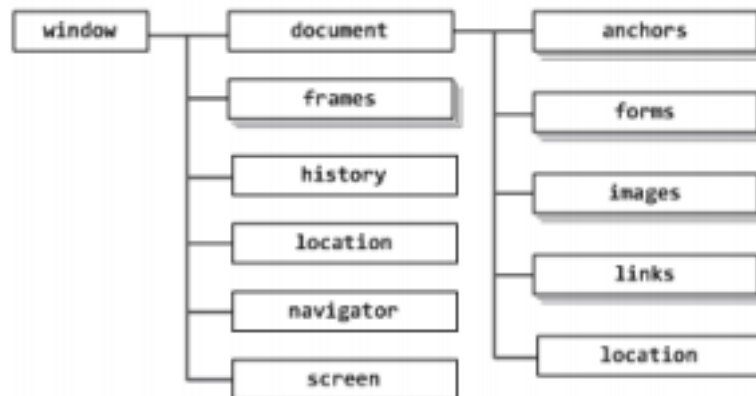


## BOM II



### El objeto document

El objeto **document** es el único que **pertenece** tanto al **DOM** como al **BOM**. Desde el punto de vista del BOM, el objeto **document** proporciona **información** sobre la **propia página HTML**.

Algunas de las **propiedades** más importantes definidas por el objeto **document** son:

| Propiedad    | Descripción   |
|--------------|---|
| lastModified | La fecha de la última modificación de la página   |
| referrer     | La URL desde la que se accedió a la página (es decir, la página anterior en el array history) |
| title        | El texto de la etiqueta <title>   |
| URL          | La URL de la página actual del navegador  |

La propiedad **title** es de **lectura** y **escritura**, por lo que además de obtener su valor, se puede establecer de forma directa:

```
// modificar el título de la página
document.title = "Nuevo titulo";
```

Además de propiedades, el objeto document contiene varios arrays con información sobre algunos elementos de la página:

| Array   | Descripción  |
|---------|--|
| anchors | Contiene todas las "anclas" de la página (los enlaces de tipo <a name="nombre_ancla"></a>) |
| applets | Contiene todos los applets de la página  |
| embeds  | Contiene todos los objetos embebidos en la página mediante la etiqueta <embed>             |
| forms   | Contiene todos los formularios de la página  |
| images  | Contiene todas las imágenes de la página   |
| links   | Contiene todos los enlaces de la página (los elementos de tipo <a href="enlace.html"></a>) |

Los elementos de cada **array** del objeto document se pueden **acceder** mediante su **índice numérico** o mediante el **nombre** del **elemento** en la **página** HTML. Si se considera por ejemplo la siguiente página HTML:

```
<html>
<head><title>Pagina de ejemplo</title></head>
<body>
<p>Primer parrafo de la pagina</p>
<a href="otra_pagina.html">Un enlace</a>

<form method="post" name="consultas">
<input type="text" name="id" />
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

Para **acceder** a los elementos de la página se pueden emplear las **funciones DOM** o los **objetos** de **BOM**:

- Párrafo: `document.getElementsByTagName("p")`
- Enlace: `document.links[0]`
- Imagen: `document.images[0]` o `document.images["logotipo"]` •

Formulario: `document.forms[0]` o `document.forms["consultas"]`

Una vez obtenida la referencia al elemento, se puede acceder al valor de sus **atributos HTML** utilizando las **propiedades** de **DOM**. De esta forma, el método del formulario se obtiene mediante `document.forms["consultas"].method` y la ruta de la imagen es `document.images[0].src`.

Ejemplo:

```
<!DOCTYPE html>
<html>
<body>
<embed src="helloworld.swf">
<embed src="helloworld.swf">
<button onclick="myFunction()">Haz clic</button>
<p id="demo"></p>
<script>
function myFunction() {
  var x = document.embeds.length;
  var y = document.embeds[0].src;
  document.getElementById("demo").innerHTML = x + ' - ' + y
; }
</script>
</body>
</html>
```

## El objeto location

El objeto **location** es uno de los objetos más útiles del BOM. Debido a la falta de estandarización, location es una **propiedad** tanto del **objeto window** como del **objeto document**.

El objeto location representa la **URL** de la **página HTML** que se muestra en la ventana del navegador y proporciona varias **propiedades** útiles para el manejo de la URL:

| Propiedad | Descripción   |
|-----------|---|
| hash      | El contenido de la URL que se encuentra después del signo # (para los enlaces de las anclas)<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>hash = #seccion |

|          |   |
|----------|---|
| host     | El nombre del servidor<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>on host = www.ejemplo.com   |
| hostname | La mayoría de las veces coincide con host, aunque en ocasiones, se eliminan las www del principio<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>on hostname = www.ejemplo.com  |
| href     | La URL completa de la página actual<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>URL =<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion  |
| pathname | Todo el contenido que se encuentra después del host<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>pathname = /ruta1/ruta2/pagina.html  |
| port     | Si se especifica en la URL, el puerto accedido<br>http://www.ejemplo.com:8080/ruta1/ruta2/pagina.html#seccion<br>port = 8080 La mayoría de URL no proporcionan un puerto, por lo que su contenido es vacío<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>on port = (vacío) |
| protocol | El protocolo empleado por la URL, es decir, todo lo que se encuentra antes de las dos barras inclinadas //<br>http://www.ejemplo.com/ruta1/ruta2/pagina.html#seccion<br>on protocol = http:   |
| search   | Todo el contenido que se encuentra tras el símbolo ?, es decir, la consulta o "query string"<br>http://www.ejemplo.com/pagina.php?variable1=valor1&variable2=valor2<br>search = ?variable1=valor1&variable2=valor2  |

De todas las propiedades, la más utilizada es **location.href**, que permite obtener o establecer la **dirección de la página que se muestra en la ventana** del navegador.

Además de las propiedades de la tabla anterior, el objeto location contiene numerosos métodos y funciones. Algunos de los métodos más útiles son los siguientes:

**// Método assign()**

**location.assign("http://www.ejemplo.com");**

**// Equivalente a location.href = "http://www.ejemplo.com"**

**// Método replace()**

**location.replace("http://www.ejemplo.com");**

```
// Similar a assign(), salvo que se borra la página actual del array  
history del navegador
```

```
// Método reload()
```

```
location.reload(true);
```

```
/* Recarga la página. Si el argumento es true, se carga la página desde  
el servidor. Si es false, se carga desde la cache del navegador */
```

## El objeto navigator

El objeto navigator es uno de los primeros objetos que incluyó el BOM y permite obtener **información** sobre el **propio navegador**.

Aunque es uno de los objetos menos estandarizados, algunas de sus propiedades son comunes en casi todos los navegadores. **A continuación se muestran algunas de esas propiedades**

| Propiedad      | Descripción  |
|----------------|--|
| oscpu          | (Sólo Firefox) Cadena que representa el sistema operativo o la CPU                           |
| platform       | Cadena que representa la plataforma sobre la que se ejecuta el navegador                     |
| plugins        | Array con la lista de plugins instalados en el navegador                                     |
| Preference()   | (Sólo Firefox) Método empleado para establecer preferencias en el navegador                  |
| product        | Cadena que representa el nombre del producto   |
| systemLanguage | (Sólo Internet Explorer) Cadena que representa el idioma del sistema operativo               |
| userAgent      | Cadena que representa la cadena que el navegador emplea para identificarse en los servidores |
| userLanguage   | (Sólo Explorer) Cadena que representa el idioma del sistema operativo                        |

El objeto navigator se emplea habitualmente para detectar el tipo y/o versión del navegador en las aplicaciones cuyo código difiere para cada navegador. Además, se emplea para detectar si el navegador tiene habilitadas las cookies y Java y también para comprobar los plugins disponibles en el navegador.

## El objeto 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**.

Las siguientes propiedades están disponibles en el objeto screen:

| Propiedad                 | Descripción   |
|---------------------------|---|
| availHeight               | Altura de pantalla disponible para las ventanas           |
| availWidth                | Anchura de pantalla disponible para las ventanas          |
| colorDepth/<br>pixelDepth | Profundidad de color de la pantalla (32 bits normalmente) |
| height                    | Altura total de la pantalla en píxel                      |
| width                     | Anchura total de la pantalla en píxel                     |

La **altura/anchura de pantalla disponible para las ventanas es menor que la altura/anchura total de la pantalla**, ya que se tiene en cuenta el tamaño de los elementos del sistema operativo como por ejemplo la barra de tareas y los bordes de las ventanas del navegador.

Además de la elaboración de estadísticas de los equipos de los usuarios, las **propiedades** del objeto **screen** se utilizan por ejemplo para determinar cómo y cuánto se puede **redimensionar** una **ventana** y para **colocar** una **ventana centrada** en la **pantalla** del usuario.

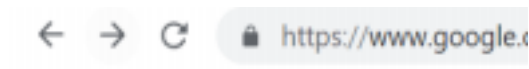
El siguiente ejemplo **redimensiona** una nueva ventana al **tamaño máximo** posible según la **pantalla** del usuario:

```
window.moveTo(0, 0);  
window.resizeTo(screen.availWidth, screen.availHeight);
```

## El objeto history

El objeto history es contiene la historia de las diferentes **páginas visitadas** por el **navegador**.

Algunos de los métodos más útiles del objeto history son los siguientes:



```
// Método back()
```

```
history.back();
```

```
// Equivalente a location.href = la URL visitada anteriormente n la  
historia del navegador
```

```
// Método forward()
```

```
history.forward();
```

```
// Equivalente a location.href = la URL visitada posteriormente en la  
history del navegador
```

```
// Método go(x)
```

```
history.go(1);
```

```
// Equivalente a history.forward()
```

```
history.go(-1);
```

```
// Equivalente a history.back()
```

```
<html>
<head>
<script>
function goBack() {
  window.history.back()
}
function goB() {
  window.history.go(-2);
}

function goForward() {
  window.history.forward()
}
</script>
</head>
<body>

<input type="button" value="Back" onclick="goBack()">
<input type="button" value="Forward"
onclick="goForward()"> <input type="button"
value="Back_2" onclick="goB()">


</body>
</html>
```