

# Objetos BOM

Location

History

Screen

Navigator

# Objeto location

- Nos da Información sobre la url actual y nos proporciona propiedades para trabajar sobre esta.

Propiedades	Métodos
hash host hostname href pathname port protocol search	reload() replace(cadenaURL)

# Objeto window.location

## Propiedades:

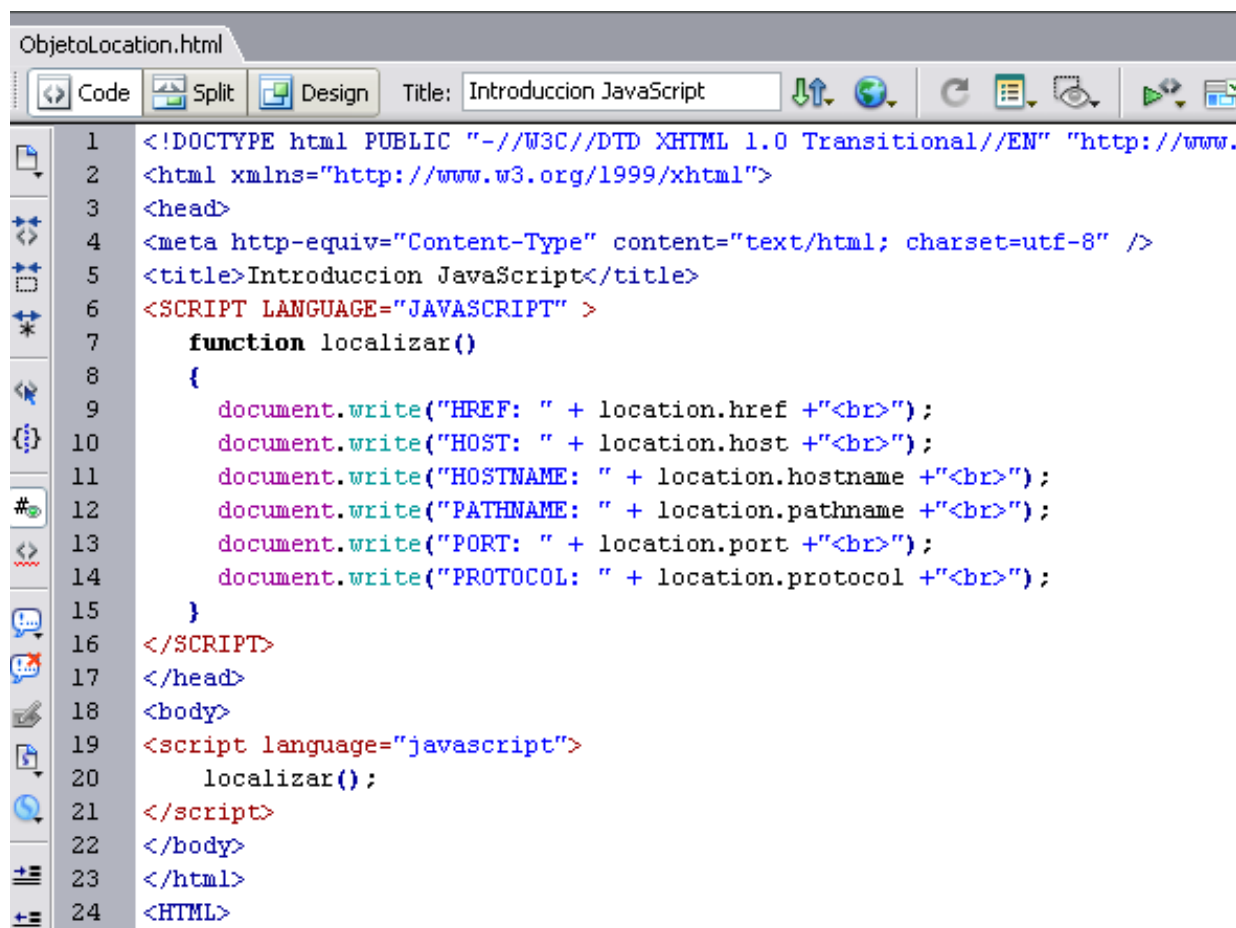
- **hash.** Devuelve lo que hay después del símbolo # incluido.
- **host.** Devuelve el nombre del servidor entre corchetes y el puerto separado por dos puntos.
- **hostname.** Devuelve el nombre del servidor.
- **href.** Devuelve el url completo.
- **pathname.** Devuelve directorio del recurso.
- **port.** Devuelve el puerto.
- **protocol.** Devuelve el protocolo.
- **search.** Devuelve lo que hay después del símbolo ? Incluido.



# Objeto window.location

- Métodos:
- **assign(url)**. *Sustituye el documento actual por el indicado en url, pero mantiene el actual en el historial.*
- **reload(forzar)**. *Refresca la página actual. Si forzar es true recarga de nuevo el recurso, si no lo toma de la caché.*
- **replace(url)**. *Carga el url indicado en la página actual, sin incluirlo en el historial de navegación*

# Objeto window.location



```
ObjetoLocation.html
Code Split Design Title: Introduccion JavaScript
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Introduccion JavaScript</title>
6 <SCRIPT LANGUAGE="JAVASCRIPT" >
7     function localizar()
8     {
9         document.write("HREF: " + location.href + "<br>");
10        document.write("HOST: " + location.host + "<br>");
11        document.write("HOSTNAME: " + location.hostname + "<br>");
12        document.write("PATHNAME: " + location.pathname + "<br>");
13        document.write("PORT: " + location.port + "<br>");
14        document.write("PROTOCOL: " + location.protocol + "<br>");
15    }
16 </SCRIPT>
17 </head>
18 <body>
19 <script language="javascript">
20     localizar();
21 </script>
22 </body>
23 </html>
24 <HTML>
```

# Ejercicio

- Crea una página web con dos botones, al clicar cualquiera de los muestra un mensaje “Hemos cambiado de ubicación esta página. En breves momentos será redireccionado...” junto a una cuenta atrás que muestre 5, 4, 3, 2, 1 (correspondiente a 5 segundos).
- Tras transcurrir 5 segundos y mostrarse la cuenta atrás, el usuario debe ser redireccionada a la url :http://xxx?
- Hacerlo con los dos métodos que tiene para ello.

# Objeto window.history

Nos da Información sobre las url vistas

Propiedades	Métodos
length	back() forward() go(url)

# Objeto window.history.

Hace referencia al historial de navegación de la ventana.

Métodos:

- forward()

Va a la página siguiente del historial de navegación.

- back()

Va a la página anterior del historial de navegación.

- go(*índice*)

Se mueve a la página indicada por *índice*.

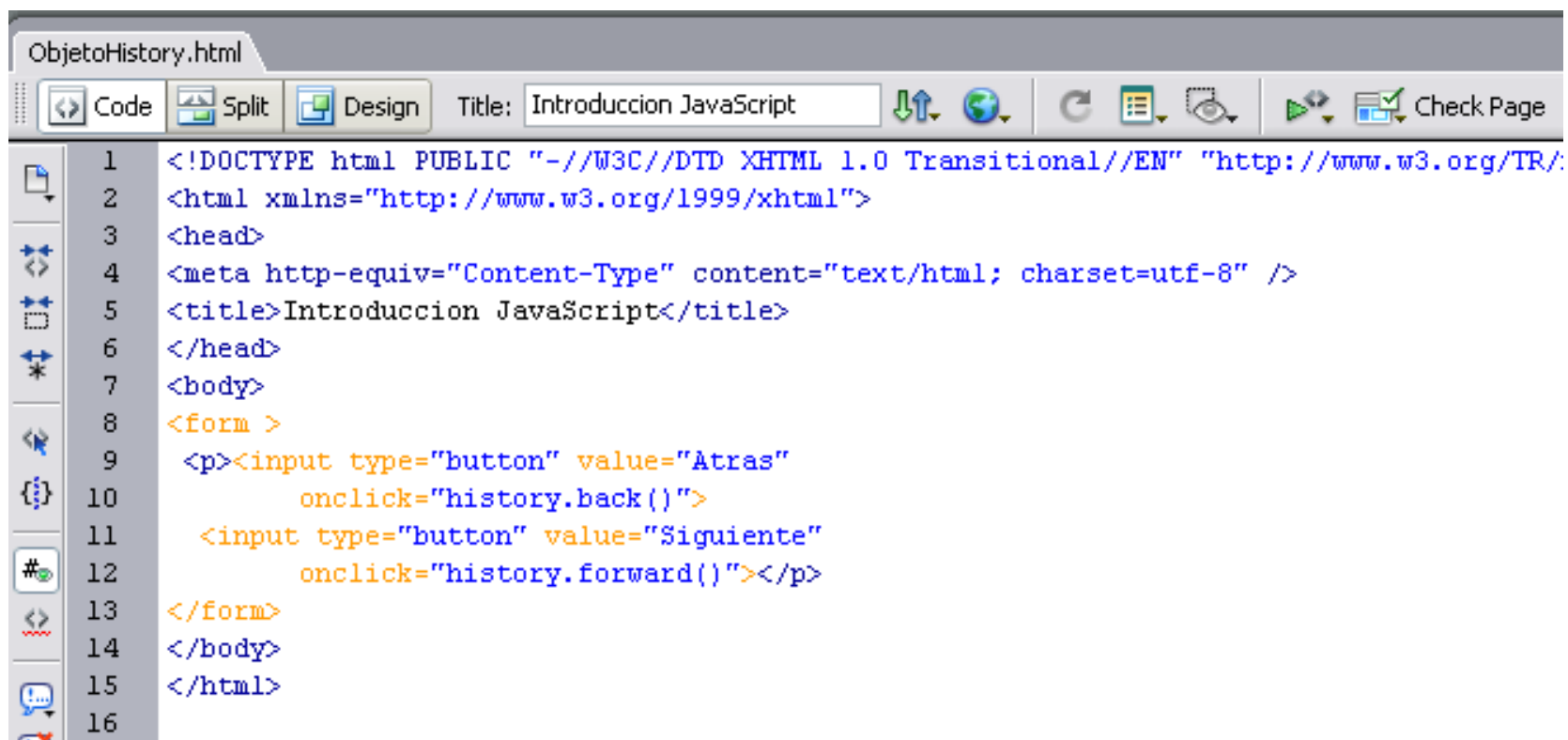
*Si índice es positivo se mueve hacia delante y si es negativo hacia atrás.*



# Probando

- Creamos tres páginas , en la primera ponemos un enlace a la segunda, en esta colocamos dos botones ANTERIOR-SIGUIENTE,( colocaremos el código correspondiente para ir atrás y adelante en el historial ) y un enlace a una tercera en la que saldrá una imagen y un botón para ir a la página 2.

# Objeto window.history



```
ObjetoHistory.html
Code Split Design Title: Introduccion JavaScript
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Introduccion JavaScript</title>
6 </head>
7 <body>
8 <form >
9 <p><input type="button" value="Atras"
10 <input type="button" value="Siguiete"
11 <input type="button" value="Siguiete"
12 <input type="button" value="Siguiete"
13 </form>
14 </body>
15 </html>
16
```

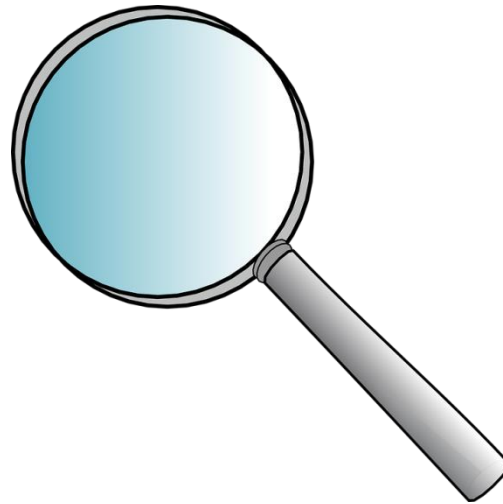
# Objeto screen

- Devuelve las características de la pantalla.

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

# Probamos

- Visualizar los valores anteriores del objeto screen.



# Objeto Screen

- 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.

# Ejercicio

Introduce por teclado el tamaño de la ventana que va aparecer.

Redimensiona la nueva ventana y que aparezca centrada en la pantalla principal.



# Objeto Navigator

Devuelve las características del navegador.

- Algunas propiedades comunes a Firefox, Internet Explorer y Chrome.
- **navigator.appCodeName**. Devuelve el nombre en código del navegador.
- **navigator.appName**. Devuelve el nombre oficial del navegador.
- **navigator.appVersion**. Devuelve la versión del navegador y de algunos de los elementos que lo forman.

# Objeto navigator (continuación).

- **navigator.cookieEnabled.** Devuelve un valor lógico indicando si están admitidas las cookies.
- **navigator.language.** Devuelve una cadena con el idioma del navegador.

En Firefox devuelve una cadena formada por el idioma y el dialecto

(por ejemplo “es-ES” o “es-LA”), en Chrome sólo el idioma (“es”).

En Internet Explorer, la propiedad es `navigator.systemLanguage` y devuelve sólo el idioma.



# Objeto navigator (continuación).

- **navigator.mimeTypes.** Devuelve un array con la lista de tipos mime instalados en el navegador.(extensiones multipropósito de correo de internet)
- **navigator.plugins.** Devuelve un array con la lista de plugins instalados en el navegador.

Aunque Internet Explorer la tiene, no devuelve nada.

- **navigator.userAgent.** Devuelve la cadena que el cliente envía al servidor.

Similar a navigator.appVersion, puede variar mucho de un navegador a otro.

- Los objetos `window.history` y `window.location` utilizan propiedades y métodos más o menos estándar.
- Las propiedades de los objetos `window.screen` y `window.navigator` están mucho menos estandarizadas.

# Recorrer propiedades



**Utilizando dos navegadores diferentes para la ejecución: chrome y firefox.**

**Recorrer las propiedades del objeto screen y del navigator, guardarlas en un array Pscreen y Pnavigator.**

**Visualizarlas. Indicar cuantas tiene cada objeto.**

**Crear a partir de estos arrays un primer array de propiedades coincidentes y otro con las que no coinciden.**

## Detectar el navegador.

En caso de que sea necesario utilizar alguna propiedad o método no estándar, se puede preguntar por ella •

- `//Esta propiedad sólo la tiene FireFox y devuelve el sistema operativo.`  
`if(navigator.oscpu){`  
`document.write("Este navegador tiene la propiedad`  
`navigator.oscpu <br>");}`  
`else {`  
`document.write("Este navegador no tiene la propiedad`  
`navigator.oscpu <br>");`  
`}`

