Unidad 3.- Utilización de los objetos predefinidos del lenguaje:

a) Utilización de objetos. Objetos nativos del lenguaje.

b) Interacción con el navegador. Objetos predefinidos asociados.

c) Generación de texto y elementos HTML desde código.

d) Aplicaciones prácticas de los marcos.

e) Gestión de la apariencia de la ventana.

f) Creación de nuevas ventanas. Comunicación entre ventanas.

a) Utilización de objetos. Objetos nativos del lenguaje.

Vamos a empezar viendo el objeto **Date** que representa a las fechas:

Tenemos los siguientes constructores:

- **new()**
- **new(***milisegundos***)** // formato Unix, milisegundos desde el día 1/1/70 a las 0:0:00
- **new(***cadena-fecha***)** // formato mes-día-año
- **new(***año***,** *mes***,** *día* [**,** *hora***,** *minuto***,** *segundo***,** milisegundo]**)**

Métodos

- **now()** devuelve el número de milisegundos transcurridos desde el 1/1/70 a las 0:0:00.
- **parse(***cadena***)** devuelve los milisegundos de la fecha, formato Unix.
- **UTC(***año***,** *mes***,** *día* [**,***hora* [**,***minuto* [**,***segundo* [**,** *milisegundo*]]]]**)** devuelve los milisegundos transcurridos entre la fecha y el 1/1/70 a las horas según el formato horario universal coordinado (UTC) o (GMT).
- **getDate()**: devuelve el día del mes de la fecha (1-31).
- **getDay()**: devuelve el día de la semana (0-6).
- **getMonth()**: devuelve el número del mes (0-11)
- **getFullYear()**: devuelve el año.
- **getHours()**: devuelve la hora.
- **getMinutes()**: devuelve los minutos
- **getSeconds()**: devuelve los segundos
- **getMiliseconds()**: devuelve los milisegundos
- **getTime()**: devuelve el número de milisegundos entre el 1/1/70 y la fecha.
- **getTimezoneOffset()**: diferencia horaria expresada en minutos, entre hora actual y hora utm.
- **getUTCDate()**: día del mes de la fecha universal.
- **getUTCDay()**: devuelve el día de la semana de la fecha universal (0-6).
- **getUTCMonth()**: devuelve el mes de la fecha universal (0-11)
- **getUTCFullYear()**: devuelve el año de la fecha universal
- **getUTCHours()**: devuelve la hora de la fecha universal
- **getUTCMinutes()**: devuelve los minutos de la fecha universal
- **getUTCSeconds()**: devuelve los segundos de la fecha universal.
- **getUTCMIliseconds()**: devuelve los milisegundos de la fecha universal.
- **setDate(***día-mes***)**: establece el día del mes de la fecha.
- **setMonth(***número-mes***)**: establece el mes
- **setFullYear(***año***)**: establece el año.
- **setHours(***hora***)**: establece la hora.
- **setMinutes(***minutos***)**: establece los minutos
- **setSeconds(***segundo***)** establece los segundos
- **setMiliseconds(***milisegundos***)** establece los milisegundos
- **setTime(***milisegundos***)**: valor numérico de la hora, milisegundos desde el 1-1-1970.
- **setUTCDate(***día***)**: establece el día del mes de la fecha universal.
- **setUTCMonth(***número-mes***)**: establece el mes de la fecha universal
- **setUTCFullYear(***año***)**: establece el año de la fecha universal
- **setUTCHours(***hora***)**: establece la hora de la fecha universal
- **setUTCMinutes(***minutos***)**: establece los minutos de la fecha universal
- **setUTCSeconds(***segundos***)**: establece los segundos de la fecha universal

- **setUTCMIliseconds(***milisegundos***)**: establece los milisegundo de la fecha universal.
- **toDateString()**: devuelve una cadena con la fecha.
- **toLocaleString([***cadena-código-país***])**:: devuelve una cadena con la fecha, usando reglas locales del país o del país indicado.
- **toLocaleDateString([***cadena-código-país***])**: devuelve una cadena con la fecha en formato del país o del país indicado.
- **toTimeString()**: devuelve una cadena con la hora
- **toLocaleTimeString([***cadena-código-país***])**: devuelve una cadena con la hora en formato del país o del país indicado.
- **toString()**: devuelve una cadena con la fecha y hora.
- **toUTCString()**: devuelve una cadena con la fecha y hora UTC.
- **toISOString()**: devuelve una cadena con la fecha en formato ISO.
- **toJSON()**: devuelve una cadena con la fecha en formato JSON.
- **valueOf()**: devuelve el número de milisegundos transcurridos desde el 1-1-70 a la fecha.

Math para operaciones matemáticas.

→ Propiedades

- **E**: constante e valor aproximado 2.718.
- **LN2**: logaritmo natural de 2, aproximadamente 0.693.
- **LN10**: logaritmo natural de 10, aproximadamente 2.302.
- **LOG2E**: logaritmo en base 2 de e, aproximadamente 1.442.
- **LOG10E**: logaritmo en base 10 de e, aproximadamente 0.434.
- **PI**: constante pi, aproximadamente 3.14159.
- **SQRT1_2**: raíz cuadrada de ½, aproximadamente 0.707.
- **SQRT2**: raíz cuadrada de 2, aproximadamente 1.414.

→ Métodos

- **abs(***número***)**: devuelve el valor absoluto del número.
- **ceil(***número***)**: devuelve un número entero, correspondiente a realizar el redondeo por exceso del número.
- **floor(***número***)**: devuelve un número entero, correspondiente a realizar el redondeo por defecto del número.
- **round(***numero***)**: devuelve el número entero correspondiente a redondear el número indicado.
- **exp(***número***)**: devuelve el valor de e elevado al número indicado.
- **pow(***base***,** *exponente***)**: devuelve el valor correspondiente a elevar la base al exponente.
- **sqrt(***número***)**: devuelve la raíz cuadrada del número.
- **log(***número***)**: devuelve el logaritmo natural (base E) del número
- **max(***lista-números***)**: devuelve el valor mayor de la lista.
- **min(***lista-números***)**: devuelve el valor menor de la lista.
- **random()**: devuelve un número pseudo-aleatorio comprendido entre 0 y 1.
- **cos(***numero***)**: devuelve el coseno del número.
- **sin(***número***)**: devuelve el seno del número.
- **tan(***número***)**: devuelve la tangente del número.
- **acos(***número***)**: devuelve el arco coseno del número en radianes.
- **asin(***número***)**: devuelve el arco seno del número en radianes.
- **atan(***número***)**: devuelve el arco tangente del número en radianes.

- **atan2(*y*, *x*)**: devuelve el ángulo (en radianes) que forman el eje X y el punto indicado.

b) Interacción con el navegador. Objetos predefinidos asociados.

Vamos a empezar viendo el objeto **navigator** (este objeto simplemente nos da información relativa al navegador que esté utilizando el usuario) que poseen los siguientes elementos:

- **Propiedades**
  - **appCodeName**: cadena que contiene el nombre del código del cliente.
  - **appName**: cadena que contiene el nombre del cliente.
  - **appVersion**: cadena que contiene información sobre la versión del cliente.
  - **language**: cadena de dos caracteres que contiene información sobre el idioma de la versión del cliente.
  - **mimeTypes**: array que contiene todos los tipos MIME soportados por el cliente. A partir de NS 3.
  - **platform**: cadena con la plataforma sobre la que se está ejecutando el programa cliente.
  - **plugins**: array que contiene todos los plug-ins soportados por el cliente. A partir de NS 3.
  - **userAgent**: cadena que contiene la cabecera completa del agente enviada en una petición HTTP. Contiene la información de las propiedades appCodeName y appVersion.
  - **cookieEnabled**: indica mediante un valor lógico si están activadas las cookies
  - **onLine**: indica si el navegador está conectado a Internet.
- **Métodos**
  - **javaEnabled()**. Devuelve true si el cliente permite la utilización de Java, en caso contrario, devuelve false.

Si codificamos:

```
001   console.log(navigator.appCodeName);
002   console.log(navigator.appName);
003   console.log(navigator.appVersion);
004   console.log(navigator.language);
005   console.log(navigator.mimeTypes);
006   console.log(navigator.platform);
007   console.log(navigator.plugins);
008   console.log(navigator.userAgent);
```

Si lo ejecutamos en **Mozilla Firefox** 15.0 obtenemos:

```
Mozilla
Netscape
5.0 (Windows)
es-ES
[object MimeTypeArray]
Win32
[object PluginArray]
Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1
```

Si lo ejecutamos en **Internet Explorer** 9

```
Mozilla
Microsoft Internet Explorer
5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
undefined
```

[object]
Win32
[object]
Mozilla/5.0 (compatible; *MSIE* 9.0; Windows NT 6.1; Trident/5.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)

Si lo ejecutamos en **Google Chrome 22.0**
Mozilla
Netscape
5.0 (Windows NT 6.1) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4
es
MimeTypeArray
Win32
PluginArray
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.4 (KHTML, like Gecko) *Chrome*/22.0.1229.94 Safari/537.4

Si lo ejecutamos en **Opera 9.8**
Mozilla
Opera
9.80 (Windows NT 6.1; U; es-ES)
es-ES
MimeTypes
Win32
PluginArray [Plugin , Plugin , Plugin , Plugin , Plugin]
Opera/9.80 (Windows NT 6.1; U; es-ES) Presto/2.10.289 Version/12.02

Si lo ejecutamos en **Safari 5.1.7**
Mozilla
Netscape
5.0 (Windows NT 6.1) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2
es-ES
MimeTypeArray
Win32
PluginArray
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2

Actualmente

Si lo ejecutamos en **Mozilla Firefox versión 32.0.3** obtenemos:
Netscape
5.0 (Windows)
es-ES
MimeTypeArray { 0=MimeType, 1=MimeType, 2=MimeType, más...}
Win32
PluginArray { 0=Plugin, 1=Plugin, 2=Plugin, más...}
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:32.0) Gecko/20100101 Firefox/32.0

Si lo ejecutamos en **Internet Explorer versión 11.0.9**
Netscape
5.0 (Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; rv:11.0) like Gecko
es-ES
[object MimeTypeArray]{0: MimeType {...}, 1: MimeType {...}, constructor: MimeTypeArray {...}, length: 2}
Win32
[object PluginArray]{0: Plugin {...}, constructor: PluginArray {...}, length: 1}
Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; rv:11.0) like Gecko

Si lo ejecutamos en **Google Chrome versión 37.0.20**

Netscape

5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36
es
MimeTypeArray
Win32
PluginArray
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36

Si lo ejecutamos en **Opera versión 24**
Netscape
5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.122 Safari/537.36 OPR/24.0.1558.64 (Edition Campaign 21)
es
MimeTypeArray
Win32
PluginArray
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.122 Safari/537.36 OPR/24.0.1558.64 (Edition Campaign 21)

Si lo ejecutamos en **Safari versión 5.1.7**
Netscape
5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2
es-ES
[object MimeTypeArray]
Win32
[object PluginArray]
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2

Si lo ejecutamos en **Google Chrome 45.0**
appCodeName: Mozilla
appName: Netscape
appVersion: 5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.99 Safari/537.36
language: es
mimeTypes: [object MimeTypeArray]
platform: Win32
plugins: [object PluginArray]
userAgent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.99 Safari/537.36

Si lo ejecutamos en **Mozilla Firefox 40.0.3**
appCodeName: Mozilla
appName: Netscape
appVersion: 5.0 (Windows)
language: es-ES
mimeTypes: [object MimeTypeArray]
platform: Win32
plugins: [object PluginArray]
userAgent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0

Si lo ejecutamos en **Opera 32.0**
appCodeName: Mozilla
appName: Netscape
appVersion: 5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36 OPR/32.0.1948.25
language: es
mimeTypes: [object MimeTypeArray]
platform: Win32

plugins: [object PluginArray]
userAgent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36 OPR/32.0.1948.25

Si lo ejecutamos en **Internet Explorer 11.0**
appCodeName: Mozilla
appName: Netscape
appVersion: 5.0 (Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; rv:11.0) like Gecko
language: es-ES
mimeTypes: [object MimeTypeArray]
platform: Win32
plugins: [object PluginArray]
userAgent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; rv:11.0) like Gecko

Si lo ejecutamos en **Safari versión 5.1.7**
appCodeName: Mozilla
appName: Netscape
appVersion: 5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2
language: es-ES
mimeTypes: [object MimeTypeArray]
platform: Win32
plugins: [object PluginArray]
userAgent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/534.57.2 (KHTML, like Gecko) Version/5.1.7 Safari/534.57.2

Para determinar el navegador Web que utilizamos podemos codificar:

```
001    if (navigator.userAgent.indexOf("Firefox") != -1) {
002        alert ("Navegador Mozilla Firefox");
003    } else if (navigator.userAgent.indexOf("MSIE") != -1) {
004        alert ("Navegador Internet Explorer");
005    } else if (navigator.userAgent.indexOf("Chrome") != -1) {
006        alert ("Navegador Google Chrome");
007    } else if (navigator.userAgent.indexOf("Safari") != -1) {
008        alert ("Navegador Apple Safari");
009    } else if (navigator.userAgent.indexOf("Opera") != -1) {
010        alert ("Navegador Opera");
011    } else if (navigator.userAgent.indexOf("OPR") != -1) {
012        alert ("Navegador Opera");
013    } else if (navigator.userAgent.indexOf(".NET CLR ") != -1) {
014        alert ("Navegador Internet Explorer");
015    }
```

El objeto **screen** devuelve información acerca de la pantalla

Propiedades

- **availHeight**: altura disponible
- **availWidth**: ancho disponibles
- **height**: altura total
- **width**: ancho total

Métodos

- **colorDepth()**: profundidad de bits de la paleta de colores
- **pixelDepth()**: resolución de color

c) Generación de texto y elementos HTML desde código.

El objeto **document** es el que tiene el contenido de toda la página que se está visualizando. Esto incluye el texto, imágenes, enlaces, formularios, ... Gracias a este objeto

vamos a poder añadir dinámicamente contenido a la página, o hacer cambios, según nos convenga.

Propiedades

- **alinkColor**: esta propiedad tiene almacenado el color de los enlaces activos
- **anchors**: se trata de un array con los enlaces internos existentes en el documento
- **applets**: es un array con los applets existentes en el documento
- **bgColor**: propiedad que almacena el color de fondo del documento
- **cookie**: es una cadena con los valores de las cookies del documento actual
- **domain**: guarda el nombre del servidor que ha servido el documento
- **embeds**: es un array con todos los EMBED del documento
- **fgColor**: en esta propiedad tenemos el color del primer plano
- **forms**: se trata de un array con todos los formularios del documento. Los formularios tienen a su vez elementos (cajas de texto, botones, etc) que tienen sus propias propiedades y métodos, y serán tratados en el siguiente capítulo.
- **images**: array con todas las imágenes del documento
- **lastModified**: es una cadena con la fecha de la última modificación del documento
- **linkColor**: propiedad que almacena el color de los enlaces
- **links**: es un array con los enlaces externos
- **location**: cadena con la URL del documento actual
- **referrer**: cadena con la URL del documento que llamó al actual, en caso de usar un enlace.
- **title**: cadena con el título del documento actual
- **vlinkColor**: propiedad en la que se guarda el color de los enlaces visitados

Métodos

- **write(***textos***)**: escribe texto en el documento, no dinámicamente, excepto opera.
- **writeln(***textos***)**: escribe texto en el documento, y además lo finaliza con un salto de línea
- **clear()**: limpia la ventana del documento.
- **open()**: abre la escritura sobre un documento.
- **close()**: cierra la escritura sobre el documento actual.

d) Aplicaciones prácticas de los marcos.

Para realizar la definición de marcos utilizaremos las etiquetas frameset y frame de la siguiente manera, la utilización de marcos está desaconsejada en HTML 5 (usar capas):

```
<frameset cols="lista-columnas" rows="lista-filas">
    <frame name="nombre-marco" src="ul-página-cargar" />
        ....
    <frame name="nombre-marco" src="ul-página-cargar" />
</frameset>
```

Para hacer referencia a los marcos mediante javascript vamos a poder utilizar:

window.parent.*nombre-marco*
parent.*nombre-marco*

window.parent.frames[*nombre-marco***]**
parent.frames[*nombre-marco***]**
window.parent.frames[*número-marco***]**
parent.frames[*número-marco***]**

En estos dos últimos casos deberemos tener en cuenta que los marcos se enumeran empezando por cero y se asignan los valores a partir de la primera aparición de la etiqueta <frame>. Vamos a ver algunos ejemplos de su utilización.

Cuando tenemos una página web con en marcos, como las siguientes:

```
<!doctype html>
<html>
   <head>
      <meta charset="utf-8">
      <title>  ejemplo de marcos con javascript </title>
   </head>
   <frameset cols="30%,70%">
        <frame name="izquierda" src="ejemplo_3_35_menu.html">
        <frameset rows="40%,60%">
           <frame name="arriba" src="">
           <frame name="abajo" src="">
        </frameset>
   </frameset>
</html>
```

siendo la página

```
<!doctype html>
<html lang="es">
   <head>
      <meta charset="utf-8">
      <title> Marcos  </title>
      <script type="text/javascript">
        function primero(pagina) {
            parent.arriba.location.href=pagina;
        }
        function segundo(pagina) {
            parent.abajo.location.href=pagina;
        }
     </script>
   </head>
   <body>
    <ul>
    <li><a href="javascript:primero('ejemplo_3_35_carnivoros.html')">grandes carnivoros</a> </li>
    <li><a href="javascript:segundo('ejemplo_3_35_carnivoros2.html')">pequeños carnivoros</a> </li>
    </ul>
   </body>
</html>
```

O bien

```
<!doctype html>
<html lang="es">
   <head>
      <meta charset="utf-8">
      <title>  Marcos </title>
      <script>
        function cargar(pagina, marco) {
            parent.frames[marco].location.href=pagina;
         }
     </script>
   </head>
   <body>
        <ul>
            <li><a href="javascript:cargar('ejemplo_3_35_carnivoros.html', 'uno')">grandes carnivoros</a> </li>
```

```
015             <li><a
    href="javascript:cargar('ejemplo_3_35_carnivoros2.html','dos')">peque&ntilde;os
    carnivoros</a></li>
016         </ul>
017     </body>
018 </html>
```

O bien

```
001 <!doctype html>
002 <html lang="es">
003     <head>
004         <meta charset="utf-8">
005         <title> Marcos  </title>
006         <script>
007           function cargar(pagina, marco) {
008               parent.frames[marco].location.href=pagina;
009            }
010       </script>
011     </head>
012     <body>
013         <ul>
014             <li><a href="javascript:cargar('ejemplo_3_35_carnivoros.html',
    1)">grandes carnivoros</a> </li>
015             <li><a
    href="javascript:cargar('ejemplo_3_35_carnivoros2.html',2)">peque&ntilde;os
    carnivoros</a></li>
016         </ul>
017     </body>
018 </html>
```

e) Gestión de la apariencia de la ventana.

A continuación listaré la mayoría de las propiedades que tenemos a mano con el objeto **window** que no son pocas:

Propiedades

- **length**: fija o devuelve el número de marcos de la ventana actual.
- **name**: fija o devuelve el nombre de la ventana que nos servirá para referirnos a ella en el código. No lo confunda con el título de la página que aparece en la parte superior izquierda.
- **defaultStatus**: nos permite fijar u obtener el mensaje por defecto que se muestra en la barra de estado. Esta propiedad no es compatible en Firefox por defecto.
- **status**: nos permite modificar el texto de la barra de estado. Esta propiedad no es compatible en Firefox por defecto.
- **outerHeight**: establece o devuelve el tamaño de la altura, en pixeles, del espacio de toda la ventana en vertical, respectivamente. Este valor incluye las barras de desplazamiento, de herramientas, etc...
- **outerWidth**: establece o devuelve el tamaño de la anchura, en pixeles, del espacio de toda la ventana en horizontal, respectivamente. Este valor incluye las barras de desplazamiento, de herramientas, etc...
- **innerHeight**: obtiene la altura, en pixeles, del espacio donde se visualiza propiamente la página. Este valor tiene en cuenta las barras de desplazamiento, si hubiera pero excluye todas las demás (menú, herramientas, estado, etc…)
- **innerWidth**: obtiene la anchura, en pixeles, del espacio donde se visualiza propiamente la página. Este valor tiene en cuenta las barras de

desplazamiento, si hubiera pero excluye todas las demás (menú, herramientas, estado, etc…)

- **opener**: hace referencia a la ventana que abrió la actual (ventana emergente o pop-up). Si la ventana no fue abierta por otra, obtendremos un valor undefined.
- **closed**: nos indica si la ventana está cerrada o no a través de un valor booleano.
- **parent**: se refiere a la ventana donde está situado el marco (<FRAMESET>) en el que estamos trabajando. En caso de no haber marcos en la página, equivale a la propia ventana.
- **top**: devuelve la ventana que se encuentra un nivel por encima del marco actual. Si no hay marcos en la página, devuelve la propia ventana.
- **self**: se refiere a la ventana actual.
- **menubar**: nos devuelve un objeto que presenta la barra de menú del navegador.
- **toolbar**: nos devuelve un objeto que representa la barra de herramientas del navegador.
- **statusbar**: nos devuelve un objeto que representa la barra de estado del navegador.
- **scrollbars**: nos devuelve el objeto que simboliza las barras de desplazamiento.
- **location**: nos devuelve el objeto location del navegador.
- **history**: nos devuelve el objeto history del navegador.
- **document**: nos devuelve el objeto document del navegador.

Métodos

Entre los métodos del objeto window encontraremos muchas funcionalidades de utilidad para aportar más dinamismo a nuestras páginas.

- **alert(***mensaje***)**: muestra un mensaje en un cuadro de diálogo con el botón Aceptar.
- **confirm(***mensaje***)**: muestra un mensaje en un cuadro de diálogo pero, a diferencia de alert, muestra los botones Aceptar y Cancelar. Además, nos permite saber sobre qué botón se ha hecho clic, ya que devuelve un valor true para Aceptar y false para Cancelar.
- **prompt(***mensaje***,** *valor_por_defecto***)**: este método es una extensión de confirm, ya que incluye una caja de texto para recoger un valor escrito por el usuario, el cual además es devuelto, opcionalmente podemos indicar un valor que será mostrado por defecto en la caja de texto cuando se muestre el mensaje. Si el usuario hace clic sobre Cancelar obtendremos un valor null.
- **focus()**: establece el foco de un objeto. Se dice que un objeto tiene el foco cuando lo estamos utilizando (por ejemplo, escribiendo en un cuadro de texto) y lo pierde cuando dejamos de usarlo. Con estos métodos podemos forzar estas situaciones.
- **blur()**: retira el foco de un objeto
- **close()**: cierra la ventana actual.
- **setInterval(***expresión***,** *milisegundos***)**: evalúa una expresión continuamente después de que hayan pasado el numero de milisegundos especificados. Este método nos devuelve un identificador, que podrá ser utilizado para detener su ejecución.
- **setTimeout(***expresión***,** *milisegundos***)**: evalúa una expresión una única vez después de esperar a que pasen el numero de milisegundos indicados. Este

método también nos devuelve un identificador para poder cancelar esta ejecución antes de que ocurra.

- **clearInterval(***identificador***)**: cancelan la ejecución de setInterval a través del identificador pasado como parámetro.
- **clearTimeout(***identificador***)**: cancelan la ejecución de setTimetout a través del identificador pasado como parámetro.
- **print()**: manda imprimir la página que estemos viendo.
- **scrollBy(***x***, *y*)**: realiza un desplazamiento horizontal y/o vertical de tantos pixeles como marquen los parámetros.
- **scrollTo(***x***, *y*)**: realiza un desplazamiento horizontal y /o vertical hasta una posición concreta.
- **moveBy(***x***, *y*)**: desplaza la ventana el número de pixeles indicados. La x indica el desplazamiento horizontal y la y el vertical. No se puede aplicar sobre la propia ventana se debe crear una nueva ventana y aplicarlo.
- **moveTo(***x***, *y*)**: mueve la ventana a la una posición concreta. No se puede aplicar sobre la propia ventana se debe crear una nueva ventana y aplicarlo.
- **resizeBy(***x***, *y*)**: redimensiona la ventana tantos pixeles como indiquen los parámetros. La x se refiere a la anchura y la y a la altura. No se puede aplicar sobre la propia ventana se debe crear una nueva ventana y aplicarlo.
- **resizeTo(***x***, *y*)**: establece una anchura y altura concretas a la ventana. No se puede aplicar sobre la propia ventana se debe crear una nueva ventana y aplicarlo.

El objeto **location** contiene la url actual, así como algunos datos de interés respecto a esta url.

Propiedades

- **hash**: nombre del enlace dentro de la url.
- **host**: nombre del host o servidor y el número de puerto.
- **hostname**: nombre del host, nombre del dominio del servidor (o la dirección IP)
- **href**: url completa de la página actual.
- **pathname**: ruta de la url.
- **port**: número del puerto, si es el 80 no se indica
- **protocol**: protocolo
- **search**: parte de la consulta incluyendo ?.

Métodos

- **assign(***url***)**: carga la página indica
- **replace(***url*): carga la página indicada
- **reload(**[*del-servidor*]**)**: recarga la página del servidor, con true, o de la cache, con false, en función del valor del parámetro.

f) Creación de nuevas ventanas. Comunicación entre ventanas.

Con el objeto **window**

- **open(***URL***, *nombre*, *parámetros*, *reemplaza_historial***)**: crea una nueva ventana (normalmente un pop-up) en la que se carga el URL especificado. Esta ventana recibe también un nombre para identificarla, se puede sustituir por _blank (nueva), _parent (marco padre), _self(actual), _top(eliminar marcos página queda) y, opcionalmente, una serie de parámetros que se pasan como una cadena de pares “propiedad=valor”

separados por comas, siendo yes o 1 valores positivos, y no o 0 los negativos. Pasemos a ver los cuales podemos incluir:

- toolbar: indica si la ventana tendrá barra de herramientas o no.
- menubar: muestra u oculta la barra de menús de la nueva ventana.
- location: muestrar u ocultar la barra de direcciones en la nueva ventana.
- directories: indica si la nueva ventana tendrá barras de dirección o no.
- scrollbars: indica si la nueva ventana tendrá barras de desplazamiento o no.
- status: nos permite controlar la visibilidad de la barra de estado.
- resizable: permite establecer si la nueva ventana podrá ser cambiada de tamaña (con el ratón) o no.
- fullscreen: indica si la ventana se verá a pantalla completa o no.
- width: establece el ancho que tendrá la ventana en pixeles.
- height: establece el alto que tendrá la ventana en pixeles.
- top: indica la distancia, en pixeles, a la que estará la ventana respecto al lado superior de la pantalla.
- letf: indica la distancia, en pixeles, a la que estará la ventana respecto al lado izquierdo de la pantalla.

Este método devuelve un identificador para la ventana que hemos abierto (para poder hacer referencia a ella después) o un valor null si la ventana no pudo abrirse (por ejemplo a causa del un bloque de ventanas emergentes). Se puede decir que el método open crea una instancia del objeto window. A través de ese nombre vamos a poder comunicar la ventana inicial con la que nos acabamos de crear.

var nueva=window.open("nueva.html");

Desde la ventana nueva vamos a hacer referencia a la ventana inicial con **opener** o bien con **window.opener**. no funciona en opera y chrome.

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset="utf-8" />
        <title>Apariencia ventana </title>
        <script>
        var nueva;
        function abrir() {
            nueva = window.open("ejemplo_3_40_nueva.html","nueva2");
            }
            function cambiar() {
            nueva.location.href="ejemplo_3_40_nueva2.html";
            }
            function yo(){
               alert("pagina inicial");
              }
        </script>
    </head>
    <body>
        <div> P&aacute;gina inicial
            <a href="javascript:abrir()" > nueva </a> <br>
            <a href="javascript:cambiar()"> cambiar </a>
        </div>
    </body>
</html>
```

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset="utf-8" />
        <title> Creada</title>
    </head>
    <body>
        <div> Página nueva recien creada </div>
```

```
009         </body>
010     </html>
```

```
001     <!doctype html>
002     <html lang="es">
003         <head>
004             <meta charset="utf-8" />
005             <title> Hija</title>
006             <script type="text/javascript">
007                 function padre() {
008                     opener.yo();
009                 }
010             </script>
011         </head>
012         <body>
013             <div style="width:300px,height:300px;background-color:red">
014                 P&aacute;gina modificada <br>
015                 <a href="javascript:padre()"> padre </a>
016             </div>
017         </body>
018     </html>
```