

Unidad Didáctica 2

Mejorar una página Web con JavaScript



Contenido

1. Introducción
2. Insertar JavaScript en documentos HTML
3. Aplicaciones

1. Introducción

JavaScript es un lenguaje de programación que se combina muy fácilmente con HTML, dotando de carácter dinámico a las páginas Web. Al igual que HTML, JavaScript es un lenguaje interpretado (no requiere compilación) cuyo código se inserta en el mismo documento de texto donde se escribe el código HTML.

Gracias a la utilización de JavaScript, se puede dar vida a nuestras páginas: ventanas de mensajes, elementos del documento que cambian de aspecto o aparecen y desaparecen, validaciones de formularios, atractivas animaciones etc. para explicar en detalle todas las posibilidades

Para explicar todos los aspectos de JavaScript sería necesario, como mínimo, un libro entero. No obstante, en este apartado se hará un breve recorrido a través de algunas posibilidades que ofrece este lenguaje.

2. Insertar JavaScript en documentos HTML

Hay varias formas de insertar código JavaScript en una página HTML. Estas son:

- **Dentro de la directiva `<SCRIPT>`:** Esta directiva se puede poner en cualquier lugar del código, pero lo normal es incluirla dentro de la etiqueta `<HEAD>`. Dentro de esta directiva, se declaran las funciones que se van a utilizar en la página. Algunos atributos que puede presentar esta directiva, son:
 - **language:** Aquí se indica el lenguaje que se escribe dentro de esta directiva. Si es JavaScript: `language = "JavaScript"`
 - **src:** Indica la URL donde se encuentre algún archivo con código fuente JavaScript. Se utiliza junto a `type` para indicar el tipo MIME del fichero, en esta caso: `src="text/javascript"`

- **Como respuesta a un evento:** Este método se usa cuando se quiere realizar una acción simple. Los eventos son notificaciones que se producen dependiendo del estado del documento y de alguna acción de usuario. Por ejemplo:

```
...  
<A href=http://www.unsitiodeejemplo.com  
onmouseover="window.status='Ir a Mi Sitio'; return true">Enlace</A>
```

...
onmouseover es un evento JavaScript que genera el texto *Ir a Mi Sitio* en la barra de estado del navegador (parte inferior) cuando se posiciona el ratón sobre el mismo (este texto sustituye a la habitual URL a la que apunta el enlace, que es lo que aparece normalmente en la barra de estado).

- **Enlace:** Aunque no es muy frecuente, se puede incluir código en el atributo *href* de los enlaces para que se ejecuten al seleccionarlo. Por ejemplo:

```
...  
<A href="javascript:window.history.foward()">
```

Al seleccionar este enlace, se invocaría la función “Adelante” del navegador.

3. Aplicaciones

En este apartado se verán algunas aplicaciones de JavaScript muy usadas en el desarrollo Web.

3.1. Personalizar el mensaje de la barra de estado

Por defecto, al posicionar el puntero del ratón sobre un enlace, en la barra de estado del navegador se mostrará la dirección a la que le lleva el mismo.

No obstante, mediante una rutina JavaScript se puede personalizar el texto que muestre la barra de estado cuando se posicione el cursor del ratón sobre

el enlace. Para ello, se debe incluir la siguiente línea de código dentro de la etiqueta <A>:

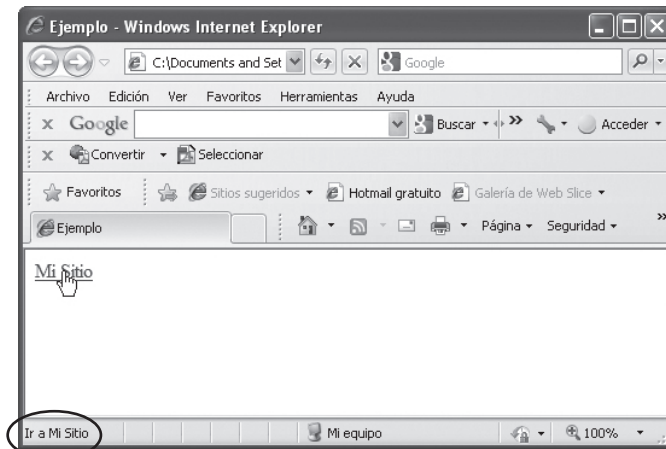
...onmouseover="window.status='Aquí viene el texto que quiere que aparezca en la barra de estado'; return true;..."

Por ejemplo:

```
..
<A href="www.unsitiodeejemplo.com" onmouseover="window.status='Ir a
Mi Sitio; return true;'">Mi Sitio</A>
```

...

El resultado en el navegador es el siguiente:



Una vez se posiciona el ratón sobre el enlace, el texto quedará fijo incluso si el puntero deja de estar posicionado de esa forma. Por esta razón, interesa usar otra función adicional (*onmouseout*) que establece otro texto cuando se el ratón se aleja del enlace. La sintaxis es idéntica a la función anterior:

...onmouseout="window.status='Aquí viene el texto que quiere que aparezca cuando el ratón se aleje del enlace';return true;"....

3.2. Abrir una ventana dimensionable

Es muy frecuente que, al acceder a una página, se abra una pequeña ventana de dimensiones predeterminadas y que ofrezca, por ejemplo, un resumen de lo que se va a encontrar en el sitio Web.

A continuación se muestra un ejemplo de este tipo de función, declarada dentro de la directiva `<HEAD>`:

```
...
<HEAD>
...
  <SCRIPT LANGUAGE="JavaScript">
    open('nombre de la ventana.html' 'Sizewindow',
        'width=300, height=350, scrollbars=no, toolbar=no')
  </SCRIPT>
</HEAD>
...
```

La función que hace esto se denomina *open*. Lo primero que es necesario especificar es el documento html que se cargará en la ventana y las dimensiones de la ventana se indicarán con *width* y *height*. En el caso de que no se desee incluir ni barras de herramientas ni barras de desplazamiento en la ventana, se signará a los parámetros *scrollbars* y *toolbar* el valor 'no'. En caso contrario, se escribiría 'yes'.

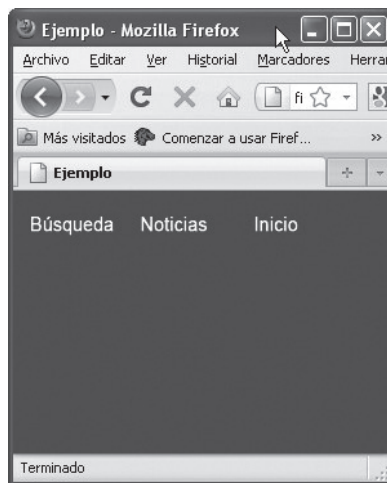
3.3. Creación de una barra de navegación mediante *rollovers*

Las imágenes de sustitución o *rollovers* es una de las aplicaciones más comunes del lenguaje JavaScript. Un *rollover* es un efecto de sustitución de una imagen por otra cuando se posiciona el ratón sobre la misma.

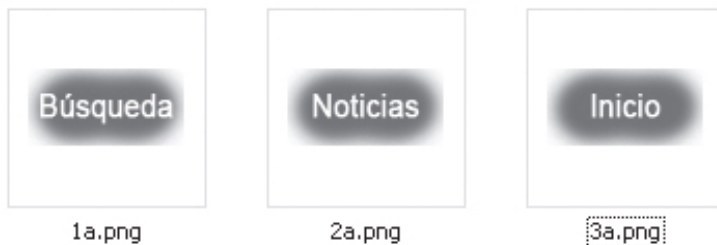
Una de las aplicaciones más frecuentes de los *rollovers* se da en el funcionamiento de las barras de navegación de los sitios Web.

Para indicar la imagen que aparece cuando se posiciona el ratón sobre el enlace se usa la función *onmouseover*, mientras que con *onmouseout* se declara la imagen que aparecerá cuando el ratón se aleje (imagen de sustitución). Ambas funciones contendrán la URL de la imagen correspondiente y deben de ubicarse dentro de la directiva ``.

A continuación se va a ver un ejemplo de construcción de una barra de navegación muy sencilla compuesta por tres enlaces. Cada enlace es una imagen png con texto blanco y fondo transparente:



De izquierda a derecha, las imágenes son: *1.gif*, *2.gif* y *3.gif*. cada imagen de la barra dispone de su propia imagen de sustitución. Estas son respectivamente (*1a.png*, *2a.png* y *3a.png*):

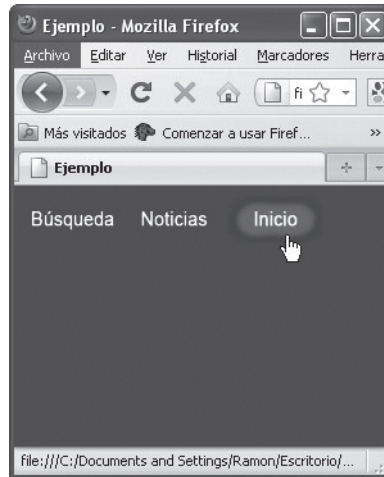


En total se disponen de seis imágenes, tres de la cuales conformarán el estado inicial de los enlaces, mientras que las otras tres serán las imágenes que sustituirán a las iniciales cuando el ratón se sitúe sobre cada una de ellas.

El código correspondiente a barra de navegación es:

```
...  
<A href="#">  
      
</A>  
<A href="#">  
      
</A>  
<A href="#">  
      
</A>  
...
```

Por ejemplo, cuando el ratón se sitúe sobre el enlace Inicio, se verá:



3.4. Validar la entrada de usuario

Es muy común encontrarse con formularios en los que se comprueba que los datos introducidos por el usuario sean correctos. Esta validación se puede efectuar cuando se envíe el formulario o cuando el puntero abandone el campo concreto a verificar.

Para el primer caso se utiliza el evento *onsubmit* en la etiqueta `<FORM>` u *onclick* en la directiva del botón de envío (`<INPUT type="submit"...>`).

El siguiente ejemplo se refiere al uso del evento *onsubmit* para verificar que un campo de texto llamado *nombre* tenga un número de caracteres mayor que 0 (una vez se envíe el formulario):

```
...
<FORM ... onsubmit=" if (this.nombre.value.length==0) { alert('El campo
nombre debe tener contenido'); return false; }; return true; }
...
```

if es una estructura condicional usada en multitud de lenguajes de programación que verifica una condición (la cual se dispone entre paréntesis). En

el caso de que la longitud del campo *nombre* sea 0, mostrará el mensaje: ‘*El campo nombre debe tener contenido*’.

Como se comentó antes, también se puede efectuar la validación del formulario cuando se abandone la edición del campo en cuestión. Para ello se utiliza *onblur*. En el siguiente ejemplo, la función (*verifica()*) es declarada previamente en la directiva *<HEAD>*, para posteriormente se “llamada” desde el campo *nombre*:

```
<HEAD>
...
<SCRIPT language="JavaScript">
    function verifica(objeto){
        if (!objeto.value.length ){
            alert ("Este campo no puede estar vacio.")
        }
    }
</SCRIPT>
</HEAD>

...
<INPUT name="Nombre" type="text" size="20"
onblur="verifica(this)">
...
```



Aplicación práctica

Se propone desarrollar el código completo de un documento que contenga un formulario con un único campo de texto de longitud 10 y un botón de envío. Antes de enviarse el formulario se deberá de verificar que dicho campo no esté vacío (al abandonar la edición del campo). Una vez hecho, verificar el código en el navegador y comprobar el resultado.

SOLUCIÓN

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <META charset="UTF-8"/>
    <TITLE>Formulario</TITLE>
    <SCRIPT language="JavaScript">
      function verify(obj){
        if (!obj.value.length){
          alert ("Este campo no puede estar vacío")
        }
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM action="ejemplo.php" method="POST">
      <FIELDSET>
        <LEGEND>Escriba su nombre</LEGEND>
        Nombre: <INPUT name="nombre"
          type="text" size="10"
          onblur="verify(this)"><BR /><BR />
        <INPUT type="submit" name="enviar"
          value="Enviar">
      </FIELDSET>
    </FORM>
  </BODY>
</HTML>
```

Si se intenta abandonar el campo *nombre* sin dejar nada escrito, aparecerá un mensaje:

Continúa en página siguiente >>

<< Viene de página anterior

