

# Lenguajes de Marcas

**Contenido:**

Introducción	2
Elementos del Formulario	3
Fieldset y Legend	7
El tipo File	9



## Unidad 7– Formularios HTML

Los formularios interactivos permiten a los autores de páginas Web poner elementos interactivos en sus páginas, por ejemplo, para recibir mensajes de sus lectores, de forma similar a las cartas de respuestas que se encuentra en algunas revistas.

El lector escribe la información rellenando campos o haciendo clic sobre botones, y lue-

go presiona un botón de envío para enviarla a una dirección URL que se suele dirigir a una dirección de correo electrónico o a un script dinámico Web como PHP, ASP o CGI.

## 1.- Introducción

Los formularios más sencillos se pueden crear utilizando solamente dos etiquetas: `<form>` y `<input>`. Si se considera el formulario que muestra la siguiente imagen:



El código HTML necesario para definir el formulario anterior se muestra a continuación:

```
<form action="http://www.librosweb.es/maneja_formulario.php" method="post">
```

```
Escribe tu nombre:
```

```
<input type="text" name="nombre" value="" />
```

```
<br/>
```

```
<input type="submit" value="Enviar" />
```

```
</form>
```

La etiqueta `<form>` encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables) y la etiqueta `<input>` permite definir varios tipos diferentes de elementos (botones y cuadros de texto).

Los atributos de la etiqueta `<form>` son los siguientes:

**action** = "url" - Indica la URL que se encarga de procesar los datos del formulario

**method** = "POST o GET" - Método HTTP empleado al enviar el formulario

**enctype** = "application/x-www-form-urlencoded o multipart/form-data" - Tipo de codificación empleada al enviar el formulario al servidor (sólo se indica de forma explícita en los formularios que permiten adjuntar archivos)

**accept** = "tipo\_de\_contenido" - Lista separada por comas de todos los tipos de archivos aceptados por el servidor (sólo para los formularios que permiten adjuntar archivos)

**Otros:** accept-charset, onsubmit, onreset

La mayoría de formularios utilizan sólo los atributos `action` y `method`. El atributo `action` indica la URL de la aplicación del servidor que se encarga de procesar los datos introducidos por los usuarios. Esta aplicación también se encarga de generar la respuesta que muestra el navegador.

El atributo `method` establece la forma en la que se envían los datos del formulario al servidor. Este atributo hace referencia al método HTTP, por lo que no es algo propio de HTML. Los dos valores que se utilizan en los formularios son `GET` y `POST`. De esta forma, casi todos los formularios incluyen el atributo `method="get"` o el atributo `method="post"`.

Al margen de otras diferencias técnicas, el método `POST` permite el envío de mucha más información que el método `GET`. En general, el método `GET` admite como máximo el envío de unos 500 bytes de información. La otra gran limitación del método `GET` es que no permite el envío de archivos adjuntos con el formulario. Además, los datos enviados mediante `GET` se ven en la barra de direcciones del navegador (se añaden al final de la URL de la página), mientras que los datos enviados mediante `POST` no se pueden ver tan fácilmente.

Si no sabes que método elegir para un formulario, existe una regla general que dice que el método `GET` se debe utilizar en los formularios que no modifican la información (por ejemplo en un formulario de búsqueda). Por su parte, el método `POST` se debería utilizar cuando el formulario modifica la información original (insertar, modificar o borrar alguna información).

El ejemplo más común de formulario con método `GET` es el de los buscadores. Si realizas una búsqueda con tu buscador favorito, verás que las palabras que has introducido en tu búsqueda aparecen como parte de la URL de la página de resultados.

Del resto de atributos de la etiqueta `<form>`, el único que se utiliza ocasionalmente es `enctype`. Como se explica más adelante, este atributo es imprescindible en los formularios que permiten adjuntar archivos.



## 2.- Elementos de formulario

Los elementos de formulario como botones y cuadros de texto también se denominan "campos de formulario" y "controles de formulario". La mayoría de controles se crean con la etiqueta `<input>`, por lo que su definición formal y su lista de atributos es muy extensa: en los formularios que permiten adjuntar archivos.

### 2.1 La etiqueta INPUT

La forma general de la etiqueta INPUT es la siguiente:

```
<INPUT TYPE="tipo" NAME="nombre" VALUE="valor" SIZE="tamaño" MAXLENGTH="max_longitud" CHECKED>
```

Los atributos de INPUT son:

**TYPE** determina el tipo de entrada que se va a utilizar y sus valores pueden ser uno de los siguientes:

- `text`, muestra una caja donde introducir texto. Es el tipo por defecto de INPUT.
- `password`, muestra una caja donde se tecleará texto cuyo eco será sustituido por asteriscos.
- `checkbox`, un botón que sólo puede tener dos estados, {on, off}.
- `radio`, un botón que sólo puede tener dos estados, {on, off}, pero que a diferencia de los checkbox, puede agruparse con otros botones de tipo radio con el mismo nombre y obtener así un comportamiento "uno de muchos".
- `submit`, un botón que produce que el formulario actual sea enviado al URL especificado en el atributo ACTION del formulario.
- `reset`, un botón que produce que todos los elementos del formulario pasen a su estado por defecto.
- `hidden`, es una entrada que no se muestra en el browser y que sirve para pasar información desde el formulario al servidor que no se quiere que el usuario vea.

**NAME** es el nombre simbólico al que se le va asociar el valor introducido en ese elemento por el usuario. Debe estar presente en todos los elementos excepto en submit y reset.

**VALUE** tiene varios significados dependiendo del elemento que se trate. Para campos texto o password especificará el valor por defecto. En botones de tipo radio o checkbox especificará el valor que tendrá el botón cuando sea presionado (el defecto es on). Para los botones submit y reset el valor se puede utilizar para especificar la etiqueta del botón. En los campos ocultos especifica el valor que se pasa al servidor.

**SIZE** es el tamaño físico del campo de entrada en elementos texto o password (por defecto es 20).

**MAXLENGTH** es el número máximo de caracteres que se permiten en un elemento de tipo texto o password (ilimitado por defecto).

**CHECKED** especifica si ese botón, radio o checkbox estará activado por defecto.

## 2.1 La etiqueta TEXTAREA

La introducción de un texto, con un número de filas y columnas determinado se puede conseguir con la utilización de la etiqueta TEXTAREA cuya forma general es la siguiente:

```
<TEXTAREA NAME="nombre" ROWS="filas" COLS="columnas">  
texto por defecto  
</TEXTAREA>
```

Con ello, se presenta una caja donde el usuario puede introducir varias líneas de texto. La caja de texto tendrá barras de desplazamiento y unas dimensiones de ROWSCOLS. El texto por defecto (optativo) aparecerá entre las etiquetas de apertura y cierre del área de texto.

## 2.3 La etiqueta SELECT

Este elemento se utiliza para presentar al usuario varias opciones y que elija entre ellas una o más. La forma general es:

```
<SELECT NAME="nombre" SIZE="num" MULTIPLE>  
<OPTION VALUE="valor_asociado_A" SELECTED>Opción A  
<OPTION VALUE="valor_asociado_B" >Opción B  
<OPTION VALUE="valor_asociado_C" >Opción C  
</SELECT>
```

Los atributos de SELECT permiten especificar el nombre lógico al que se asociará la opción elegida de entre las posibles. Sólo serán visibles las que se especifiquen en SIZE. Si es 1, el elemento selección tendrá la apariencia de un menú desplegable (cuidado con el número de opciones, pues podría desbordar la pantalla). Normalmente, el elemento selección permite seleccionar sólo una de sus opciones, pero con MULTIPLE podremos hacer que la selección afecte a varios de sus elementos.

Las opciones se especifican con la etiqueta OPTION que puede aparecer con un atributo VALUE que indica el valor que se asociará al elemento de selección en caso de que esa opción sea la elegida. Este atributo puede obviarse, en cuyo caso el valor devuelto será la descripción de la opción misma. Se puede especificar con SELECTED la opción seleccionada por defecto.

## 3.- Fieldset y Legend

Los atributos FIELDSET y LEGEND se utilizan en conjunto y sirven respectivamente para definir y etiquetar grupos lógicos de elementos de formularios. Realmente no afectan a la operativa del formulario, pero sirven para agrupar elementos en diferentes áreas, de modo que se clarifique la entrada de datos del usuario. Al formar varios grupos de elementos se puede crear una estructura mucho más fácil de asimilar por el usuario, sobre todo si se trata de formularios que tengan muchos elementos.

### Etiqueta FIELDSET

La etiqueta FIELDSET sirve para agrupar los elementos. Se utiliza con su respectiva etiqueta de cierre y lo que hace es crear un recuadro que rodea a los elementos de formulario colocados dentro de ella.

Por ejemplo, se podría usar de esta manera:

```
<fieldset>  
Elemento de formulario: <input type="text" name="elemento1"> <br>  
Otro elemento: <input type="text" name="otro">  
</fieldset>
```

Simplemente creará un cuadrado que agrupará los dos elementos del formulario incluidos dentro del FIELDSET. Podemos ver el resultado en una página aparte.

### Etiqueta LEGEND

LEGEND sirve para nombrar o etiquetar un grupo creado con FIELDSET. Añade simplemente una nota aclaratoria sobre qué tipo de información se está agrupando en el recuadro. Tampoco sirve para nada en especial, de no ser porque queda bonita y porque puede servir para ayudar al usuario y mejorar la interfaz y la claridad de los formularios.

La etiqueta LEGEND se coloca después de la etiqueta FIELDSET. Tiene su propia etiqueta de cierre. Entre <LEGEND> y </LEGEND> colocamos el texto con el que queremos marcar el recuadro definido con FIELDSET.

A la etiqueta LEGEND se le puede poner el atributo align para indicar el lugar donde debe aparecer la leyenda. Por ejemplo podríamos indicar align=right para que apareciera en la parte de la derecha, en lugar de la izquierda, que es donde aparece por defecto.

Veamos ahora un ejemplo sencillo de utilización de las etiquetas FIELDSET y LEGEND en conjunto.

```
<form>  
<fieldset>  
<legend align="right">Datos personales</legend>  
Nombre: <input type="text" name="nombre"> <br>  
Edad: <input type="text" name="edad" size="2"> <br>
```

```
Dirección: <input type="text" name="direccion">
</fieldset> <br>
<fieldset>
<legend align="right">Datos de tu ordenador</legend>
Modelo de ordenador: <input type="text" name="modelo"> <br>
Sistema que te da el problema:
<select>
<option value=cpu>CPU
<option value=impresora>Impresora
</select>
</fieldset> <br>
<fieldset>
<legend align="right">Descripción del problema</legend>
<textarea cols="55" rows="8" name="descripcion"></textarea>
</fieldset>
</form>
```

El ejemplo en marcha tendría este aspecto.

Nombre: <input type="text"/> Edad: <input type="text"/> Dirección: <input type="text"/>	Datos personales
Modelo de ordenador: <input type="text"/> Sistema que te da el problema: CPU <input type="button" value="v"/>	Datos de tu ordenador
<input type="text"/>	Descripción del problema

Podremos comprobar como aparecen tres bloques en el formulario, producidos por tres etiquetas FIELDSET, con varios campos de formulario incluidos en cada una. Además, cada uno de los FIELDSET tiene dentro un LEGEND que sirve para nombrar con una leyenda cada uno de los tres bloques.

### Etiqueta LABEL

Aunque no forma parte del objetivo de este artículo, queremos nombrar también otra etiqueta llamada LABEL que sí tiene una utilidad especial en la creación de formularios, además de la estética. Sirve para poner texto al lado de los elementos de formulario y que tal texto esté asociado al propio elemento. Ese texto, que pondremos con el tag LABEL, se asocia a un elemento concreto con el atributo FOR, colocando como valor del atributo el identificador del campo que se está asociando.

```
<label for="edad">Edad</label> <input type="text" name="edad" id="edad">
```

Como vemos, hemos creado un LABEL y hemos colocado en el atributo FOR el nombre del campo de formulario que estamos asociando a ese texto. El resultado es que el texto colocado dentro de LABEL es un elemento interactivo, al que podemos hacer clic y sería como si hiciésemos clic en el propio campo asociado al LABEL.

## 4.- El tipo FILE

El control de tipo file nos permite enviar un archivo al servidor. Nuevamente el HTML solo indica al navegador que debe enviar el archivo al servidor pero debe haber en el servidor un programa que lo almacene en una carpeta del servidor.

Veamos la sintaxis para disponer un control de tipo file:

```
<input type="file" name="archivo">
```

Nuevamente utilizamos el elemento HTML input para definir este tipo de control. En la propiedad type inicializamos con el valor file. Inicializar la propiedad name también es importante ya que mediante este nombre se lo recupera en el servidor.

Otra cosa muy importante a tener en cuenta cuando hacemos upload de archivos al servidor es inicializar la propiedad enctype del elemento form:

```
<form method="post" action="registrardatos.php" enctype="multipart/form-data">
```

Con esto indicamos al navegador que el formulario almacena uno o más archivos que deben ser enviados al servidor. Confeccionemos una página que solicite el ingreso de un nombre y la foto de la persona: El control de tipo file nos permite enviar un archivo al servidor.

Confeccionemos una página que solicite la inserción de un nombre y la foto de la persona:

```
<html>
<head>
<title>Prueba de formulario</title>
</head>
<body>
<form action="registrardatos.php" method="post" enctype="multipart/form-data">
Ingrese su nombre:
<input type="text" name="nombre" size="30"><br>
Seleccione la foto:
<input type="file" name="foto">
<br>
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

Ingrese su nombre:

Seleccione la foto:  No se ha seleccionado ningún archivo.

IES MIGUEL ROMERO ESTEO  
C/MARTIN CARRIÓN S/N  
29006 MALAGA

Teléfono: 951298668

Fax: 951298670

*Estos apuntes forman parte del curso “Lenguajes de Marcas y Sistemas de Gestión de la Información” que forma parte del currículum del ciclo superior de “Administración de Sistemas Informáticos en Red”.*

*El temario completo se encuentra en la web del instituto Miguel Romero Esteo.*



**[www.romeroesteo.es](http://www.romeroesteo.es)**