

Unidad Didáctica 7

Formularios

Contenido

1. Introducción
2. La directiva **<form>** y sus atributos
3. Elementos de formulario. La directiva <input>
4. Otros elementos
5. Formato

1. Introducción

Los formularios son un método para recoger información introducida en la Web por el visitante de la página. Una vez introducida la información, ésta es enviada al servidor a través de aplicaciones externas a HTML (como PHP, ASP, JSP etc.), las cuales no son el propósito de este manual. No obstante, la creación de los elementos básicos que pueden formar parte de un formulario, sí que es un tema importante en el aprendizaje HTML.

2. La directiva `<FORM>` y sus atributos

2.1. Atributos básicos

Todo formulario está contenido por la etiqueta `<FORM>`, dentro de la cual se definen los diferentes elementos que puedan constituir el formulario (cajas de texto, listas desplegables etc.).

Esta directiva posee dos atributos de obligada aparición: *action* y *method*. El primero se utiliza para definir la acción que debe de ejecutarse con los datos introducidos en el formulario y por lo general contiene la URL del documento encargado de recibir los datos y generar alguna respuesta (similar a un enlace). Respecto al atributo *method*, se refiere a la forma en la que el formulario envía (a la URL de *action*) los datos y puede ser: "GET" y "POST" (este último es el más usual).

A continuación se pueden ver dos ejemplos de uso de `<FORM>`:

```
...  
<FORM action="busqueda.php" method="POST">  
...
```



Definición

El envío de datos por el método GET, consiste en definir estos datos en la propia URL que “llama” documento que los recibirá. Estos datos se definen después de la URL y separada de la misma por el símbolo ‘?’. En el caso de enviar más de un dato, éstos serán separados con el carácter ‘&’. Por ejemplo, una dirección Web (www.undominio.com/recibir.php) va a recibir los datos de un formulario enviado por GET. Estos datos son el nombre y apellidos del usuario, por lo que la URL que envía los datos tiene la forma:

```
www.undominioejemplo.com/recibir.php?nombre=Juan&apellido=Martinez
```

El método POST es el más usado ya que envía los datos al margen de la URL y no son visibles en la barra de direcciones del navegador.

2.2. Otros atributos

El atributo *enctype* describe el método a utilizar para codificar el contenido del formulario. Su valor por defecto es:

```
enctype="application/x-www-form-urlencoded"
```

El atributo *name* permite identificar al formulario con un nombre. Para visualizar los resultados del envío del formulario en una ventana distinta, se utiliza el atributo *target* (*target*=_blank; igual que en <A>, ver Capítulo 7).

El atributo *accept-charset*, especifica la lista de codificaciones de caracteres para los datos introducidos que son aceptadas por el servidor que procesa el formulario. El valor es una lista de valores de codificaciones de caracteres separadas por espacios y/o comas que son los tipos MIME a los que pertenecen los ficheros enviados en el formulario. El valor por defecto de este atributo es la cadena reservada “UNKNOWN” (“desconocido”).

3. Elementos de formulario. La directiva `<INPUT>`

Un formulario está compuesto por uno o más elementos a través de los cuales el usuario puede introducir los datos a enviar. La forma de introducir estos datos dependerá del tipo de elemento de formulario en cuestión, los cuales son definidos con la directiva `<INPUT>`, cuya tipología es indicada con el atributo `type`. A continuación se enumeran los diferentes valores que puede contener este atributo.



Definición

`disabled` es un atributo de la directiva `<INPUT>` (aunque también es soportado por `<TEXTAREA>`) que indica que un elemento concreto está desactivado. Por otro lado, el atributo `readonly` impide modificar el elemento.

3.1. TEXT

El elemento `"text"` es, básicamente, una caja de texto donde el usuario puede escribir lo que desee. Es el valor por defecto del atributo `type` (omitirlo es lo mismo que escribir `type="text"`). El atributo `name` contiene el nombre de la variable que almacenará lo que se introduzca en este elemento de formulario. El atributo `size` es el tamaño máximo de caracteres que se visualizarán (tamaño visual del campo de texto) y `maxlength` establece la longitud máxima de caracteres que se pueden introducir.

Otro atributo es `value` (de la directiva `<INPUT>`) que permite establecer un valor inicial en el elemento de formulario. Por ejemplo:

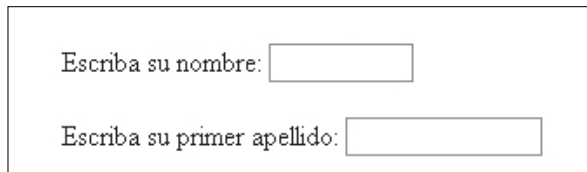
```
...
<INPUT type="text" name="nombre" size="10" maxlength="15"
value="Juan"/>
...
```

En esta línea la caja de texto se visualizaría con un valor inicial (“escrito”) de “Juan”, el cual puede ser modificado antes del envío del formulario.

A continuación se muestra el código correspondiente a un formulario básico con dos cajas de texto

```
...  
<FORM action="busqueda.php" method="POST">  
  Escriba su nombre:  
  <INPUT type="text" name="nombre" size="10" maxlength="15"/>  
  <BR /><BR />Escriba su primer apellido:  
  <INPUT type="text" name="apellido" size="15" maxlength="20"/>  
</FORM>  
...
```

El aspecto que tendrá en el navegador es:



Visualización de un formulario web con dos campos de texto. El primer campo está etiquetado como "Escriba su nombre:" y el segundo como "Escriba su primer apellido:". Ambos campos son rectángulos blancos con bordes grises.

3.2. PASSWORD

Esta entrada de formulario es muy similar a la anterior y comparte los mismos atributos. La diferencia está en que los datos que se escriben se “camuflan” con algún símbolo (el cual depende del navegador). Esto es útil para campos donde se introduzcan contraseñas u otros datos confidenciales que se deseen ocultar a los ojos de los demás. Por ejemplo:

```
...  
<FORM action="busqueda.php" method="POST">  
  Escriba su clave:
```

```

    <INPUT type="password" name="clave" size="10"
    maxlength="15"/>
</FORM>
...

```

El aspecto que tendrá en el navegador es:



3.3. Hidden

Este tipo consiste en un “almacén interno de datos”, cuyo valor no puede ser modificado ni introducido por el usuario, ya que es un elemento no visible. El valor que se envía con el formulario se debe de especificar, desde el código, con el atributo *value* y permanece constante hasta que no se modifique este atributo. Por ejemplo:

```

...
<FORM action="busqueda.php" method="POST">
    <INPUT type="hidden" name="oculto" value="Arena"/>
</FORM>
...

```

3.4. Checkbox

Este elemento consiste en una casilla de selección, la cual puede estar seleccionada o no. En este caso, *value* contiene lo que se enviará, pero sólo en el caso de que dicha casilla esté seleccionada. Si se incluye el atributo *checked*, inicialmente la casilla estará marcada. Por ejemplo:

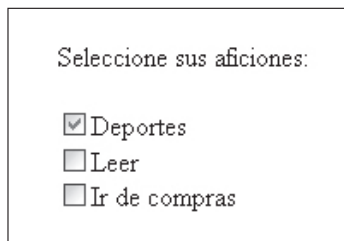
```

<FORM action="busqueda.php" method="POST">

```

```
Seleccione sus aficiones:<BR /><BR />
<INPUT type="checkbox" name="deportes" checked>Deportes<BR />
<INPUT type="checkbox" name="lectura">Leer<BR />
<INPUT type="checkbox" name="compras">Ir de compras
</FORM>
```

El resultado en el navegador es:



Seleccione sus aficiones:

- Deportes
- Leer
- Ir de compras

3.5. RADIO

Es similar a la anterior, pero asigna una misma variable a varios elementos. Esto significa que, cuando existen varios elementos de radio que tengan el mismo valor del atributo *name*, solo uno de ellos puede estar seleccionado. Los atributos *checked* y *value* funcionan de la misma forma que en el caso anterior. Por ejemplo:

```
...
<FORM action="busqueda.php" method="POST">
  &#191Cree que saldremos de la crisis?:<BR /><BR />
  <INPUT type="radio" name="crisis" value="1" checked/>Si<BR />
  <INPUT type="radio" name="crisis" value="2"/>No<BR />
  <INPUT type="radio" name="crisis" value="3"/>NS/NC
</FORM>
..
```

En el navegador se mostrará:

¿Cree que saldremos de la crisis?:

Sí

No

NS/NC

3.6. RESET

Este elemento no almacena ningún tipo de información. Se trata de un elemento especial, similar a un botón, que es capaz de (al “pulsarlo”) borrar todo lo escrito en un formulario. El atributo *value* determina el texto que aparecerá sobre el botón (por defecto es “Reset”). Por ejemplo:

```
...
<INPUT type="reset" value="Borrar todo"/>
...
```

Borrar todo

3.7. Submit

Se trata un elemento fundamental en cualquier formulario ya que, al seleccionar este botón, el formulario se enviará. El atributo *value* funciona igual que en el elemento *reset* (por defecto es “Submit”). Por ejemplo:

```
...  
<INPUT type="submit" value="Enviar"/>  
...
```



3.8. Image

Este es un tipo muy cercano a los mapas de imágenes (ver Capítulo anterior). Se trata de una imagen, cuya ubicación se indica con el atributo *src*, que actúa igual que el botón *submit*. Por otro lado también envía un parámetro (identificado por el atributo *name*) que corresponde a las coordenadas sobre las que hizo clic el usuario. Este componente es muy útil cuando se tiene una barra de herramientas compuesta por varias imágenes y se desea conocer la opción elegida. Por ejemplo:

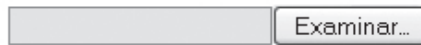
```
...  
<INPUT type="image" src="imágenes/img1.jpg" name="coord"/>  
...
```

3.9. File

Este elemento permite el envío de ficheros por HTTP y consiste en un campo de texto adjunto a un botón *Examinar*. En el campo de texto se escribe la ruta donde está el fichero a enviar. El botón *Examinar* ayuda a seleccionar, desde el explorador del S.O., la ubicación exacta del archivo en cuestión. En los formularios donde se utilicen estos elementos es necesario incluir en la directiva *<FORM>* el valor del atributo *enctype* como *multipart/form-data*. A continuación se muestra un ejemplo:

```
...  
<FORM action="busqueda.php" method="POST" enctype="multipart/form-  
data">
```

```
<INPUT type="file" name="fichero"/>
</FORM>
...
```



3.10. Elementos nuevos de HTML 5

HTML 5 aporta nuevos elementos tipo *INPUT*, que ya son reconocidos por algunos navegadores. A continuación se van a ver algunos de ellos.

3.10.1 RANGE

RANGE es un elemento de formulario que consiste en una barra sobre la que se puede desplazar un cursor. Cuando se envíe el formulario, el valor que se mandará corresponderá a la posición del cursor, ya que la barra tiene una escala de valores comprendida entre un valor mínimo (se indica con el atributo *min*) y uno máximo (especificado con el atributo *max*). Por ejemplo, observe el código de siguiente elemento de formulario y su aspecto en el navegador:

```
...
<INPUT name="marcador" type="range" min="0" max="100"/>
...
```



Imagine que dispone de un formulario con el elemento de la imagen y sitúa el cursor (haciendo clic sobre el cursor y desplazarlo sin soltar el ratón) justo en la mitad de la barra. Al enviar el formulario, se enviará una variable llama-

da *marcador* con valor 50 (correspondiente a la mitad comprendida entre 0 y 100).

3.10.2 Elementos de validación

Los elementos de formulario del tipo validación adquieren un estado concreto en el momento de que se introduzca una información no válida. Este estado debe de ser definido previamente mediante CSS. Por ejemplo, en nuestro caso queremos que el fondo de los elementos de validación adquiera color rojo mientras no detecte un valor correcto en la casilla correspondiente. Para ello, definimos el siguiente estilo CSS, utilizando la pseudo-clase *invalid*. No se preocupe si no entiende esto muy bien, ya lo comprenderá mejor cuando veamos el Módulo 3, referente a las Hojas de estilo CSS.

```
...  
:invalid { background-color: red; }  
...
```

A continuación vamos a ver algunos de estos nuevos elementos de validación.

3.10.2.1 color

Este elemento consiste en una caja de texto que adquirirá un estado incorrecto, mientras no se escriba el nombre (nomenclatura en inglés) o el valor (hexadecimal, ver Módulo 3) de un color válido. Por ejemplo:

```
...  
<INPUT type="color" />  
...
```



3.10.2.2 *number*

number consiste en otra caja de texto, la cual adquirirá un estado inválido cuando se introduzca cualquier carácter distinto a un número decimal. Por ejemplo:

<input type="text" value="123"/>	<input type="text" value="abc"/>
Número Válido	Número inválido

...

```
<INPUT type="number" />
```

...

3.10.2.3 *email*

Este elemento es muy similar a los anteriores, con la diferencia de que presentará el estado inválido hasta que no se escriba una dirección de correo electrónico, la cual debe de tener el siguiente formato:

nombre@dominio.extensión

Por ejemplo:

<input type="text" value="info@dom.es"/>	<input type="text" value="abcscdef"/>
email Válido	email inválido

4. Otros elementos

En este apartado se van a ver algunos elementos de formulario que disponen de directivas propias.

4.1. Textarea

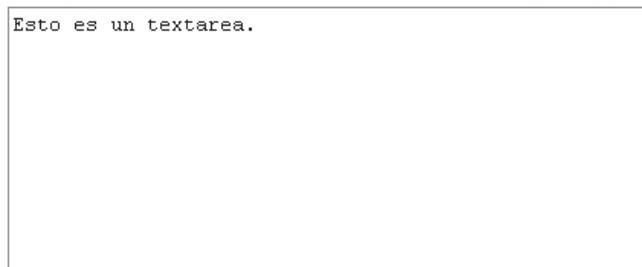
Este elemento es similar a *text* pero consiste en un cuadro de texto con varias filas y columnas. Se define por la etiqueta `<TEXTAREA>` y dispone de cierre: `</TEXTAREA>`. En caso de querer mostrar algún contenido inicial en el área de texto, éste debe ser indicado entre la apertura y cierre (a diferencia del atributo *value*).

Las dimensiones del cuadro de texto se definen con los atributos *rows* (filas) y *cols* (anchura de cada fila). Esta directiva también dispone de atributo *name*.

A continuación se muestra un ejemplo:

```
...  
<TEXTAREA name="cuadro" cols="50" rows="10">Esto es un textarea.</  
TEXTAREA>  
...
```

Es aspecto en el navegador es:



A screenshot of a web browser displaying a text area. The text area is a rectangular box with a thin border. Inside the box, the text "Esto es un textarea." is visible at the top left. The rest of the box is empty, showing the white background of the browser.



Aplicación práctica

A modo de repaso, se propone escribir el código HTML del siguiente formulario:

Introduzca el nombre de usuario:

Introduzca su clave:

Seleccione el sexo:
 Hombre Mujer

Explique en este cuadro sus expectativas profesionales.

Aclaraciones:

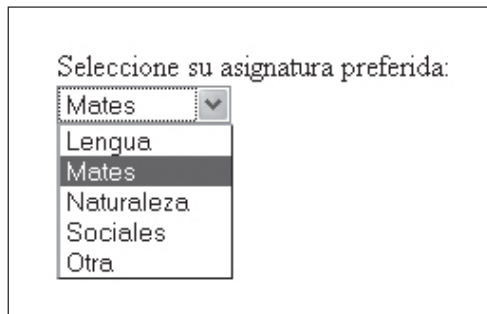
- Los elementos del formulario se llamarán: form1, form2,...
- El formulario se enviará, por *POST*, a: *recibir.php*.
- No existen campos *hidden*.
- Establezca el tamaño de las cajas y áreas de texto de forma aproximada.
- Recuerde que, para establecer espacios en blanco, en HTML se usa: * *.

Continúa en página siguiente >>

ción. Para establecer qué elemento de la lista estará seleccionado por defecto, se añade el atributo *selected* en la directiva `<OPTION>` correspondiente. Esta directiva admite el atributo *disabled*. Por ejemplo:

```
...
Seleccione su asignatura preferida: <BR />
<SELECT name="option">
  <OPTION value="a">Lengua</OPTION>
  <OPTION value="b" selected>Mates</OPTION>
  <OPTION value="c">Naturaleza</OPTION>
  <OPTION value="d">Sociales</OPTION>
  <OPTION value="e">Otra</OPTION>
</SELECT>
...
```

En el navegador se verá (al desplegar la lista):



4.3. Button

Este elemento agrupa todas las funciones asociadas a los botones. Se declara con `<BUTTON>` y Puede tener los siguientes atributos:

- *name*: Mismo cometido que en los demás elementos.
- *value*: Valor asociado al botón.
- *type*: Tipo de botón. Puede ser:

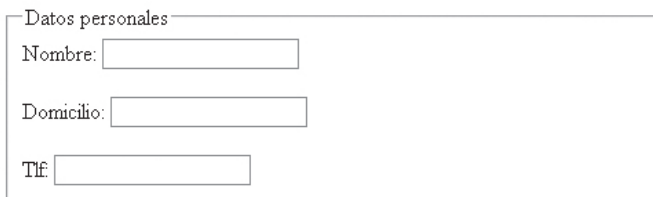
- *reset*: Igual que el INPUT correspondiente.
- *submit*: Igual que el INPUT correspondiente.
- *button*: Esta opción no añade ninguna funcionalidad específica, pero el formulario se envía al pulsarlo.

4.4. Fieldset y Legend

La directiva <FIELDSET> permite agrupar contenido en un formulario, mientras que la directiva <LEGEND> (que está dentro de <FIELDSET> permite nombrar esta agrupación. Esto se verá más claro en el siguiente ejemplo:

```
...
<FORM action="ejemplo.php" method="POST">
  <FIELDSET>
    <LEGEND>Datos personales</LEGEND>
    Nombre: <INPUT type="text" name="pnombre"/>
    <BR /><BR />
    Domicilio: <INPUT type="text" name="pdireccion"/>
    <BR /><BR />
    Tlf: <INPUT type="text" name="ptelefono"/><BR />
  </FIELDSET>
</FORM>
...
```

El resultado es:



The image shows a rendered HTML form. It is enclosed in a rectangular border. At the top left, the text "Datos personales" is displayed. Below this, there are three text input fields, each preceded by a label: "Nombre:", "Domicilio:", and "Tlf:". Each label and its corresponding input field are vertically aligned. The input fields are empty text boxes.

5. Formato

Cuando se dispone de un formulario en un sitio Web, es fundamental organizar y alinear el contenido para un mejor comprensión y visualización. El método más común es insertar cada elemento en las celdas de una o varias tablas. El número de tablas, celdas, columnas etc. dependerá de la estructura y la distribución que se desee conseguir. A continuación se muestra el ejemplo anterior tabulado en tablas (ver el Capítulo 5 de este módulo para recordar las directivas correspondientes a las tablas).

```

...
<FORM action="ejemplo.php" method="POST">
  <FIELDSET>
    <LEGEND>Datos personales</LEGEND>
    <TABLE>
      <TR>
        <TD>Nombre:&nbsp;</TD>
        <TD><INPUT type="text"
          name="pnombre"/></TD>
      </TR>
      <TR>
        <TD>Domicilio:&nbsp;</TD>
        <TD><INPUT type="text"
          name="pdireccion"/></TD>
      </TR>
      <TR>
        <TD>Tlf:&nbsp;</TD>
        <TD><INPUT type="text"
          name="ptelefono"/></TD>
      </TR>
    </TABLE>
  </FIELDSET>
</FORM>
...

```

Diseño y desarrollo web con HTML 5 y CSS

Datos personales

Nombre:	<input type="text"/>
Domicilio:	<input type="text"/>
Tlf:	<input type="text"/>

Gracias a las hojas de estilo CSS se podrían aplicar estilos a las celdas para ajustar espaciados, fuentes de texto, márgenes etc.