

Formularios en HTML

Los formularios son la forma en que podemos intercambiar información con nuestro visitante. Desde luego, este nuevo aspecto resulta primordial para gran cantidad de acciones que se pueden llevar a cabo mediante el Web: Comprar un artículo, rellenar una encuesta, enviar un comentario al autor...

Los formularios son esas famosas cajas de texto y botones que podemos encontrar en muchas páginas web. Son muy utilizados para realizar búsquedas o bien para introducir datos personales por ejemplo en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Usando HTML podemos únicamente enviar el formulario a un correo electrónico. Si queremos procesar el formulario mediante un programa tendremos que emplear otros lenguajes más sofisticados, como ASP o PHP, que nos permitirán, entre otras cosas, el tratamiento de formularios.

Los formularios son definidos por medio de las etiquetas `<FORM>` y `</FORM>`. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario. Dentro de esta etiqueta `<FORM>` debemos especificar algunos atributos:

action

Define el tipo de acción a llevar a cabo con el formulario. Como ya hemos dicho, existen dos posibilidades:

- El formulario es enviado a una dirección de correo electrónico
- El formulario es enviado a un programa o script que procesa su contenido

En el primer caso, el contenido del formulario es enviado a la dirección de correo electrónico especificada por medio de una sintaxis de este tipo:

```
<FORM action="mailto:direccion@correo.com" ...>
```

Si lo que queremos es que el formulario sea procesado por un programa, hemos de especificar la dirección del archivo que contiene dicho programa. La etiqueta quedaría en este caso de la siguiente forma:

```
<FORM action="dirección del archivo" ...>
```

method

Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar esta atributo son **post** y **get**. A efectos prácticos y, salvo que se os diga lo contrario, daremos siempre el valor post.

enctype

Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser "text/plain". Así conseguimos que se envíe el contenido del formulario como texto plano dentro del email.

name

Se utiliza para indicar el nombre del formulario.

Ejemplo de etiqueta <FORM> completa

Así, para el caso más habitual -el envío del formulario por correo- la etiqueta de creación del formulario tendrá el siguiente aspecto:

```
<FORM action="mailto:direccion@correo.com (o nombre del archivo de proceso)" method="post"
enctype="text/plain">
```

Entre esta etiqueta y su cierre colocaremos el resto de etiquetas que darán forma a nuestro formulario. .

Mandar formulario por correo electrónico

Los formularios se utilizan habitualmente para implementar un tipo de contacto con el navegante, que consiste en que éste pueda mandarnos sus comentarios por correo electrónico a nuestro buzón.

Elementos de Formularios. Campos de texto

El HTML nos propone una gran diversidad de alternativas a la hora de crear nuestros formularios. Estas van desde la clásica caja de texto hasta la lista de opciones pasando por las cajas de validación.

Veamos en qué consiste cada una de estas modalidades y como podemos implementarlas en nuestro formulario.

Texto corto

Las cajas de texto son colocadas por medio de la etiqueta <INPUT>. Dentro de esta etiqueta hemos de especificar el valor de dos atributos: **type** y **name**.

La etiqueta es de la siguiente forma:

```
<INPUT type="text" name="nombre">
```

De este modo expresamos nuestro deseo de crear una caja de texto cuyo contenido será llamado nombre (por ejemplo). El aspecto de este tipo de cajas es de sobra conocido, aquí lo podéis ver:

El nombre del elemento del formulario es de gran importancia para poder identificarlo en nuestro programa de procesamiento o en el mail recibido. Por otra parte, es importante indicar el atributo type, ya que, como veremos, existen otras modalidades de formulario que usan esta misma etiqueta.

El empleo de estas cajas esta fundamentalmente destinado a la toma de datos breves: palabras o conjuntos de palabras de longitud relativamente corta.

Además de estos dos atributos, esenciales para el correcto funcionamiento de nuestra etiqueta, existen otra serie de atributos que pueden resultarnos de utilidad pero que no son imprescindibles:

size

Define el tamaño de la caja en número de caracteres. Si al escribir el usuario llega al final de la caja, el texto ira desfilando a medida que se escribe haciendo desaparecer la parte de texto que queda a la izquierda.

maxlength

Indica el tamaño máximo del texto que puede ser tomado por el formulario. Es importante no confundirlo con el atributo size. Mientras el primero define el tamaño aparente de la caja de texto, maxlength indica el tamaño máximo real del texto que se puede escribir. Podemos tener una caja de texto con un tamaño aparente (size) que es menor que el tamaño máximo (maxlength). Lo que ocurrirá en este caso es que, al escribir, el texto ira desfilando dentro de la caja hasta que lleguemos a su tamaño máximo definido por maxlength, momento en el cual nos será imposible continuar escribiendo.

value

En algunos casos puede resultarnos interesante asignar un valor definido al campo en cuestión. Esto puede ayudar al usuario a rellenar más rápidamente el formulario o darle alguna idea sobre la naturaleza de datos que se requieren. Este valor inicial del campo puede ser expresado mediante el atributo value. Veamos su efecto con un ejemplo sencillo:

```
<INPUT type="text" name="nombre" value="Perico Palotes">
```

Genera un campo de este tipo:

Nota: estamos obligados a utilizar la etiqueta <FORM>

Aunque de lo que se lee en estos capítulos sobre formularios se puede entender bien esto, hemos querido remarcarlo para que quede muy claro: Cuando queremos utilizar en cualquier situación elementos de formulario debemos escribirlos siempre entre las etiquetas <FORM> y </FORM>. De lo contrario, los elementos se verán perfectamente en Explorer pero no en Netscape.

Dicho de otra forma, en Netscape no se visualizan los elementos de formulario a no ser que esten colocados entre las correspondientes etiquetas de inicio y fin de formulario.

Es por ello que para mostrar un campo de texto no vale con poner la etiqueta <INPUT>, sino que habrá que ponerla dentro de un formulario. Así:

```
<FORM>  
<INPUT type="text" name="nombre" value="Perico Palotes">  
</FORM>
```

Texto oculto

Podemos esconder el texto escrito por medio asteriscos de manera a aportar una cierta confidencialidad. Este tipo de campos son análogos a los de texto con una sola diferencia: reemplazamos el atributo type="text" por type="password":

```
<INPUT type="password" name="nombre">
```

En este caso, podéis comprobar que al escribir dentro del campo en lugar de texto veréis asteriscos.

Estos campos son ideales para la introducción de datos confidenciales, principalmente códigos de acceso. Se ve en funcionamiento a continuación.

Tipo File

```
<INPUT type="file" name="cualquierarchivo">
```

permite subir ficheros al servidor de la pagina web. En el botón aparece la palabra examinar, que ayuda a la selección del archivo a subir

Texto largo

Si deseamos poner a la disposición de usuario un campo de texto donde pueda escribir cómodamente sobre un espacio compuesto de varias líneas, hemos de invocar una nueva etiqueta: `<TEXTAREA>` y su cierre correspondiente.

Este tipo de campos son prácticos cuando el contenido a enviar no es un nombre teléfono o cualquier otro dato breve, sino más bien, un comentario, opinión, etc.

Dentro de la etiqueta `textarea` deberemos indicar, como para el caso visto anteriormente, el atributo `name` para asociar el contenido a un nombre que será asemejado a una variable en los programas de proceso. Además, podemos definir las dimensiones del campo a partir de los atributos siguientes:

rows

Define el número de líneas del campo de texto.

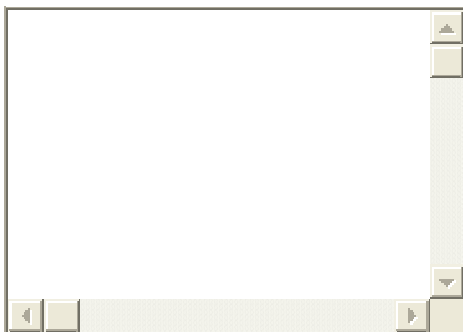
cols

Define el número de columnas del campo de texto.

La etiqueta queda por tanto de esta forma:

```
<TEXTAREA name="comentario" rows="10" cols="40"></TEXTAREA>
```

El resultado es el siguiente:



Asimismo, es posible predefinir el contenido del campo. Para ello, no usaremos el atributo `value` sino que escribiremos dentro de la etiqueta el contenido que deseamos atribuirle. Veámoslo:

```
<TEXTAREA name="comentario" rows="10" cols="40">Escribe tu comentario....</TEXTAREA>
```

Dará como resultado:

Otros elementos de formulario

Efectivamente, los textos son un manera muy práctica de hacernos llegar la información del navegante. No obstante, en muchos casos, los textos son difícilmente adaptables a programas que puedan procesarlos debidamente o bien, puede que su contenido no se ajuste al tipo de información que requerimos. Es por ello que, en determinados casos, puede resultar más efectivo proponer una elección al navegante a partir del planteamiento de una serie de opciones.

Este es el caso de, por ejemplo, ofrecer una lista de países, el tipo de tarjeta de crédito para un pago,...

Este tipo de opciones pueden ser expresadas de diferentes formas. Veamos a continuación cuales son:

Listas de opciones

Las listas de opciones son ese tipo de menús desplegables que nos permiten elegir una (o varias) de las múltiples opciones que nos proponen. Para construirlas emplearemos una etiqueta con su respectivo cierre: <SELECT>

Como para los casos ya vistos, dentro de esta etiqueta definiremos su nombre por medio del atributo name. Cada opción será incluida en una línea precedida de la etiqueta <OPTION>.

Podemos ver, a partir de estas directivas, la forma más típica y sencilla de esta etiqueta:

```
<SELECT name="estacion">
  <OPTION>Primavera</OPTION>
  <OPTION>Verano</OPTION>
  <OPTION>Otoño</OPTION>
  <OPTION>Invierno</OPTION>
</SELECT>
```

El resultado es:

Esta estructura puede verse modificada principalmente a partir de otros dos atributos:

size

Indica el número de valores mostrados de la lista. El resto pueden ser vistos por medio de la barra lateral de desplazamiento.

multiple

Permite la selección de varios elementos de la lista. La elección de más de un elemento se hace como con el explorador de Windows, a partir de las teclas ctrl o shift. Este atributo se expresa sin valor

alguno, es decir, no se utiliza con el igual: simplemente se pone para conseguir el efecto, o no se pone si queremos una lista desplegable común.

Consejo: Si es posible, no uses multiple

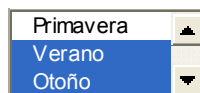
No recomendamos especialmente la puesta en práctica de esta opción ya que el manejo de las teclas ctrl o shift para elegir varias opciones puede ser desconocido para el navegante. Evidentemente, siempre cabe la posibilidad de explicarle como funciona aunque no dejara de ser una complicación para más para el visitante.

Veamos cual es el efecto producido por estos dos atributos cambiando la línea:

```
<SELECT name="estacion">
```

por:

```
<SELECT name="estacion" size="3" multiple>
```



La lista quedara de esta forma:

La etiqueta **<OPTION>** puede asimismo ser matizada por medio de **otros atributos**

selected

Del mismo modo que multiple, este atributo no toma ningún valor sino que simplemente indica que la opción que lo presenta esta elegida por defecto.

Así, si cambiamos la línea del código anterior:

```
<OPTION>Otoño</OPTION>
```

por:

```
<OPTION selected>Otoño</OPTION>
```

El resultado será:  **value**

Define el valor de la opción que será enviado al programa o correo electrónico si el usuario elige esa opción. Este atributo puede resultar muy útil si el formulario es enviado a un programa puesto que a cada opción se le puede asociar un número o letra, lo cual es más fácilmente manipulable que una palabra o texto, podríamos así escribir líneas del tipo:

```
<OPTION value="1">Primavera</OPTION>
```

De este modo, si el usuario elige primavera, lo que le llegara al programa (o correo) es una variable llamada estacion que tendrá como valor 1. En el correo electrónico recibiríamos: estacion=1

Botones de radio

Existe otra alternativa para plantear una elección, en este caso, obligamos al internauta a elegir únicamente una de las opciones que se le proponen.

La etiqueta empleada en este caso es `<INPUT>` en la cual tendremos el atributo `type` ha de tomar el valor `radio`. Veamos un ejemplo:

```
<INPUT type="radio" name="estacion" value="1">Primavera <br>
<INPUT type="radio" name="estacion" value="2">Verano <br>
<INPUT type="radio" name="estacion" value="3">Otoño <br>
<INPUT type="radio" name="estacion" value="4">Invierno
```

Nota: Hay que fijarse que la etiqueta `<INPUT type="radio">` sólo coloca la casilla pinchable en la página. Los textos que aparecen al lado, así como los saltos de línea los colocamos con el correspondiente texto en el código de la página y las etiquetas HTML que necesitamos.

El resultado es el siguiente:

- Primavera
- Verano
- Otoño
- Invierno

Como puede verse, a cada una de las opciones se le atribuye una etiqueta `input` dentro de la cual asignamos el mismo nombre (`name`) para todas las opciones y un valor (`value`) distinto. Si el usuario elige supuestamente Otoño, recibiremos en nuestro correo una línea tal que esta: `estacion=3`

Cabe señalar que es posible preseleccionar por defecto una de las opciones. Esto puede ser conseguido por medio del atributo **checked**:

```
<INPUT type="radio" name="estacion" value="2" checked>Verano
```

Veamos el efecto:

- Primavera
- Verano
- Otoño
- Invierno

Cajas de validación

Este tipo de elementos pueden ser activados o desactivados por el visitante por un simple clic sobre la caja en cuestión. La sintaxis utilizada es muy similar a las vistas anteriormente:

```
<INPUT type="checkbox" name="paella">Me gusta la paella
```

El efecto: Me gusta la paella La única diferencia fundamental es el valor adoptado por el atributo type.

Del mismo modo que para los botones de radio, podemos activar la caja por medio del atributo **checked**.

El tipo de información que llegara a nuestro correo (o al programa) será del tipo: paella=on (u off dependiendo si ha sido activada o no)

Envío, borrado y demás en formularios HTML

Los formularios han de permitirnos su envío mediante un botón. También puede resultar práctico poder proponer un botón de borrado o bien acompañarlo de datos ocultos que puedan ayudarnos en su procesamiento.

botón de envío

Para dar por finalizado el proceso de relleno del formulario y hacerlo llegar a su gestor, el navegante ha de validarlo por medio de un botón previsto a tal efecto. La construcción de dicho botón no reviste ninguna dificultad una vez familiarizados con las etiquetas input ya vistas:

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

Con este código generamos un botón como este:



Como puede verse, tan solo hemos de especificar que se trata de un botón de envío (type="submit") y hemos de definir el mensaje del botón por medio del atributo value.

botón de borrado

Este botón nos permitirá borrar el formulario por completo en el caso de que el usuario desee rehacerlo desde el principio. Su estructura sintáctica es análoga a la anterior:

```
<INPUT type="reset" value="Borrar">
```

A diferencia del botón de envío, indispensable en cualquier formulario, el botón de borrado resulta meramente optativo y no es utilizado frecuentemente. Hay que tener cuidado de no ponerlo muy cerca del botón de envío y de distinguir claramente el uno del otro.

Datos ocultos

En algunos casos, aparte de los propios datos enviados por el usuario, puede resultar práctico enviar datos definidos por nosotros mismos que ayuden al programa en su procesamiento del formulario. Este tipo de datos, que no se muestran en la página pero si pueden ser detectados solicitando el código fuente, no son frecuentemente utilizados por páginas construidas en HTML, son más bien usados por páginas que emplean tecnologías de servidor. No os asustéis, veremos más adelante qué quiere decir esto. Tan solo queremos dar constancia de su existencia y de su modo creación. He aquí un ejemplo:

```
<INPUT type=hidden name="sitio" value="www.thico.net">
```

Esta etiqueta, incluida dentro de nuestro formulario, enviara un dato adicional al correo o programa encargado de la gestión del formulario. podríamos, a partir de este dato, dar a conocer al programa el origen del formulario o algún tipo de acción a llevar a cabo.

Botones normales

Dentro de los formularios también podemos colocar botones normales, pulsables como cualquier otro botón. Igual que ocurre con los campos hidden, estos botones por si solos no tienen mucha utilidad pero podremos necesitarlos para realizar acciones en el futuro. Su sintaxis es la siguiente.

```
<INPUT type=button value="Texto escrito en el botón">
```

Quedaría de esta manera:

El uso más frecuente de un botón es en la programación en el cliente. Utilizando lenguajes como JavaScript, que veremos en la segunda parte del curso, podemos definir acciones a tomar cuando un visitante pulse el botón de una página web.

Crear un enlace o link en un botón

Este ejemplo muestra como se puede poner un link en un botón:

```
<FORM action="http://www.thico.net/" method="post">
  <INPUT type="submit" value="Web del Thico">
</FORM>
```

Que se puede [ver también en una página](#).

Ejemplo de formulario

Pasemos ahora a ejemplificar todo lo aprendido a partir de la creación de un formulario que consulta el grado de satisfacción de los usuarios de una línea de autobuses ficticia. El formulario está construido para que envíe los datos por correo electrónico a un buzón determinado.

Vemos el formulario en esta página. Vosotros tratar de construirlo para ver si habéis entendido bien los temas sobre formularios.

Nombre

Email @

Población

Sexo

Hombre Mujer

Frecuencia de los viajes

Comentarios sobre su satisfacción personal

- Deseo recibir notificación de las novedades en las líneas de autobuses.

El formulario se puede [ver también en una página a parte](#). Recordad que podéis ver el código fuente de cualquier página web utilizando los menús de vuestro navegador, así podréis revisar el código que hemos utilizado para construir el formulario.

Para acabar, vamos a ver lo que recibirían por correo electrónico en la empresa de autobuses cuando un usuario cualquiera rellenase este formulario y pulsase sobre el botón de envío.

```

nombre=Federico Mijo Silvestre
email=fede@terramix.com
poblacion=Astorga, León
sexo=Varon
utilizacion=2
comentarios=No creo que sea una buena línea. Poner más autobuses.
recibir_info=on

```

Otras etiquetas de las cuales daremos una breve explicación.

<FIELDSET>... </FIELDSET>

Hasta ahora, no disponíamos de ninguna manera de agrupar visualmente varios controles, si no echábamos mano de elementos que no son del formulario, como tablas o imágenes. Ahora, si encerramos una parte de un formulario dentro de la etiqueta FIELDSET se mostrara un rectángulo alrededor de los mismos. Además, podemos indicar un título por medio de la etiqueta LEGEND, que admite el parámetro align="left / center / right / top / bottom", lo que nos permite alinear el título horizontal y verticalmente. La única pega es que deberemos introducir el conjunto en una celda de tabla con un ancho determinado, ya que si no lo hacemos así el recuadro abarcará todo el ancho de pantalla disponible.

Ejemplo.

```

<FORM action="cgi-bin/control.exe" method="post" enctype="text/plain" name="miform">
<TABLE width="200">
<TR>
<TD>
<FIELDSET>
<LEGEND align="left"><font color="red">Caja de texto</font></LEGEND>
    pon tu nombre: <INPUT type="text" size="15">
</FIELDSET>
</TD>
</TR>
</TABLE>
</FORM>

```

<BUTTON>... </BUTTON>

A partir de la implementación de los estándares HTML 4.0 contamos con varias etiquetas nuevas para construir formularios, siendo BUTTON una de ellas, bastante útil por cierto.

Esta etiqueta proporciona un método único para la implementación de cualquier tipo de botón de formulario. Sus principales atributos son:

- type= " tipo ", que puede tomar los ya conocidos valores submit (por defecto), reset y button.
- name= " nombre ", que asigna un nombre identificador único al botón.
- value= " texto ", que define el texto que va a aparecer en el botón.

La principal ventaja que aporta estas etiquetas es que ahora vamos a poder introducir dentro de ellas cualquier elemento de HTML, como imágenes y tablas.

Ejemplos.

```

<FORM action="cgi-bin/control.exe" method="post" enctype="text/plain" name="miform">
<BUTTON name="boton_1" type="button">
<TABLE width="10" cellspacing="0" cellpadding="2" border="1">
<TR>
<TD>uno</TD>
<TD>dos</TD>
</TR>
<TR>
<TD>tres</TD>
<TD>cuatro</TD>
</TR>
</TABLE>
</BUTTON>
</FORM>

```

Se puede [ver una posible solución del ejercicio en este enlace](#).