

## **FORMULARIOS**

Se podría diseñar una página HTML que permitiese a un usuario comprar un determinado producto.

Esta página contendría campos de datos que permitiesen seleccionar el tipo de producto, la forma de pago, cómo, dónde y cuando desea recibir el producto, etc.

Además, la página contendría algún elemento de control que permitiese al usuario dar la orden de compra.

Cuando un cliente rellena los campos de datos de la página de compra de un producto y da la orden de compra, estos datos se pasarían como entrada a un programa que se encargaría de realizar las acciones oportunas para que se llevase a cabo la compra del producto.

Cada campo de datos de un formulario tiene un nombre y un valor (el introducido por el usuario o el establecido por defecto). Los datos introducidos por un usuario se pasan al programa correspondiente como pares nombre-valor .

Las etiquetas que ofrece HTML para crear formularios son: `<BUTTON>`, `<FIELDSET>`, `<FORM>`, `<INPUT>`, `<ISINDEX>`, `<LABEL>`, `<LEGEND>`, `<OPTION>`, `<SELECT>` y `<TEXTAREA>`. La etiqueta `<ISINDEX>` genera un campo de texto para solicitar información al usuario. En HTML 4.0 esta etiqueta está obsoleta y ha sido sustituida por `<INPUT>`.

Los formularios utilizan etiquetas y elementos especiales para habilitar distintas opciones de introducción de información que el formulario mostrará y pondrá en funcionamiento.

Los elementos especiales se conocen como controles y vamos a ver como configurarlos y utilizarlos en los formularios.

Hay dos elementos claves, con sus atributos asociados, que debe conocer para poder crear formularios:

**<FORM>...</FORM>**: El elemento indispensable de todos los formularios, con etiquetas de apertura y cierre. Acepta varios atributos.

Los dos más importantes son **action**, que se combina con un URL al procesador del formulario y **method**, que puede tomar valores *get* o *post* dependiendo de cómo se configure la tecnología de formulario.

**<INPUT>**: Este elemento es responsable de la gestión de los controles de introducción de datos que se pondrán dentro de la etiqueta, que sólo tiene versión de apertura, no hay acompañante de cierre. Los atributos más utilizados con la etiqueta **<INPUT>** son:

- **type="x"**: Especifica el tipo de control al que se está llamando. Veremos detalles en breve.
- 
- **name="x"**: El atributo name designa el control.
- 
- **value="x"**: Describe el control de introducción de datos. Es optativo en todos los controles excepto en radio.
- **size="x"**: La anchura en píxeles del control de introducción de datos. A veces, el número de caracteres determina el tamaño del control, como en el caso de los controles text y password.
- **maxlength="x"**: El número máximo de caracteres a introducir permitido en un control de introducción de datos
- **checked="x"**: Esta opción preselecciona un botón de opción dado o un cuadro de comprobación dentro de un formulario.
- **<TEXTAREA>...</TEXTAREA>**: Este elemento crea un área de introducción de texto. Conceptualmente, es igual que un control, pero se gestiona utilizando un elemento en lugar de un atributo. Los atributos que acepta son name, **rows="x"**, donde x define el número de líneas de un cuadro y **cols="x"**, donde las columnas especifican la anchura del cuadro.

- **<SELECT>...</SELECT>**: El elemento SELECT crea un menú. Puede ser desplegable o de lista textual, dependiendo de cómo defina los atributos del elemento. Puede añadir el atributo múltiple si quiere que el visitante pueda elegir más de una opción en el menú.
- **<OPTION>...</OPTION>**: Define cada uno de los elementos individuales de una lista dentro de un menú. Se necesitan las dos etiquetas, de apertura y cierre. ¡Hay que señalar que la última etiqueta es optativa con el elemento OPTION!.
- Los **controles** definen la opción de introducción de datos que aparecerá en pantalla. Si los consideramos sintácticamente, son un valor HTML y se ponen en el lugar del valor del atributo type dentro de una declaración de la etiqueta <INPUT>.
- Los controles que se encuentran disponibles son los que aparecen indicados a continuación:
- **text**: Crea un cuadro de texto de introducción de datos que consiste en una sola línea. La anchura de la caja se controla con el atributo *size*.
- **password**: Es exactamente igual a text, con la única diferencia de que los caracteres introducidos por el visitante del sitio se reflejarán como asteriscos.
- **checkbox**: Crea un cuadro de comprobación que se puede marcar. Puede tener varios cuadros de comprobación en una selección y todos ellos pueden ser marcados cuando sea aplicable.
- **radio**: Con este control crea un botón de opción. Puede tener tantos como quiera en un área, pero sólo se puede seleccionar uno.
- **submit**: Este control crea el familiar botón para enviar la información, cuya apariencia 'elevada' le identifica como

botón a pulsar. Puede personalizarse la palabra en el control utilizando el atributo `value`.

- **reset**: Con las mismas características de aspecto y personalización que `submit`, este control borra el contenido del formulario para que el visitante pueda volver a introducir sus respuestas.

- **file**: Crea un control de selección de archivo. El visitante del sitio puede seleccionar, con él, un archivo a descargar.

- **hidden**: Los controles ocultos (esto es, `hidden`) son aquellos que no son presentados por el navegador. Se utilizan para que el receptor de los datos del formulario introduzca información.

- **image**: Permite la inserción de una imagen personalizada. Con él, el diseñador puede utilizar el gráfico en `submit` y `reset` en lugar de la opción predeterminada.

- **button**: Crea un botón que hay que pulsar. Debe estar asociado con un script para que funcione, ya que no hay una acción incorporada en ellos.

- Los atributos principales de un formulario son **action** y **method**. El primero permite determinar la dirección URL (Uniform Resource Locator) del programa o script que procesa los datos del formulario en el servidor. Es decir, asegura que los datos que se envían van al lugar correcto para su procesamiento.

- El segundo permite indicar el método que se va a utilizar para enviar los datos del formulario. Éste puede tomar dos valores: *GET* o *POST*. Normalmente se utiliza *POST*, que indica que los datos se envíen por la entrada estándar. Si se utiliza *GET*, los datos se envían unidos al URL.

Sólo con la etiqueta `<FORM>` no podemos obtener información de un usuario a través de una página HTML. Es necesario incluir dentro del formulario algún elemento que

sirva para este propósito. La etiqueta `<INPUT>` nos permite definir dentro de un formulario algunos elementos básicos para que el usuario introduzca información. Más adelante veremos otros elementos que también pueden incluirse dentro de un formulario.

- La etiqueta `<INPUT>` permite incluir los siguientes elementos en un formulario: botones (para enviar información, borrar u otras acciones), imágenes para enviar información, entrada de datos booleana (checkbox), campos excluyentes (radio), campos de texto, datos ocultos y campos de password.
- Conviene dar un nombre a los campos que introduzcamos en un formulario, ya que al enviar la información ésta va como pares nombre-valor. Para ello se dispone del atributo `name` en la etiqueta `<INPUT>`.
- Con `<INPUT>` pueden introducirse tres tipos de botones en un formulario:
  - `<INPUT type="submit" value="boton1">`
  - `<INPUT type="reset" value="boton2">`
  - `<INPUT type="button" value="boton3" onclick="procedimiento()">`
- El primero, al pulsarlo tiene como efecto enviar la información de pares nombre-valor de todos los elementos del formulario, como entrada al programa indicado en la URL de la etiqueta `<FORM>`. El segundo tipo de botón tiene como efecto borrar los datos que haya introducido el usuario en el formulario y dejar todos los campos con el valor por defecto
- El último tipo de botón tiene un efecto similar al primero, excepto que en lugar de invocar el programa indicado en la URL de la etiqueta `<FORM>`, se invoca al procedimiento indicado en la etiqueta.

- Existe la posibilidad de incluir una imagen que sustituya al botón submit. Para ello, basta con incluir:

- `<INPUT type="image" src="image.gif" width="50" height="25" alt="ve" value="submit">`

- En el formulario aparecerá la imagen "imagen.gif". Cuando el usuario pulse con el ratón encima de la imagen, el efecto obtenido será el mismo que al pulsar un botón de tipo submit. Esto dará lugar a que el botón se funda con el diseño de la página.

- También puede utilizar el elemento `BUTTON`, que pone la imagen dentro del botón correspondiente:

- `<BUTTON name="submit" value="submit" type="submit">`

- Enviar `<IMG src="imagen.gif" width="50" height="25" alt="ve">`

- `</BUTTON>`

Para preparar el botón repetir con este método sólo tiene que cambiar el atributo `value` a "reset". La siguiente figura muestra un ejemplo de cada estilo

Estos cuadros de comprobación se incluyen a través del campo booleano `checkbox`.

La apariencia que tiene es la de un pequeño cuadrado, que puede estar seleccionado (marca dentro del cuadro) o no (cuadro vacío).

Para activar o desactivar un campo de este estilo basta con pulsar con el ratón en el cuadrado. Si deseamos que una `checkbox` aparezca por defecto seleccionada, basta con incluir el atributo `checked` de la etiqueta `<INPUT>`

En el ejemplo siguiente pueden verse varios elementos del tipo `checkbox`.

Los cuatro primeros comparten el mismo nombre y el usuario puede seleccionarlos como desee (no son excluyentes). En este caso es importante no olvidar el

atributo `value` para saber cuáles están seleccionados y cuáles no, cuando lleguen como entrada al programa que se invoca pulsando el botón “submit”.

Un campo similar al checkbox es el radio, excepto que este último es de selección exclusiva. Es lo que se denomina botón de opción. Para que varios botones radio pertenezcan al mismo grupo y, por tanto, sólo uno de ellos pueda estar activo a la vez, basta con que todos ellos tengan el mismo nombre (atributo `name`). En el ejemplo siguiente hay dos botones de este estilo. Como puede verse, ambos tienen el mismo nombre. Para seleccionar uno de los dos por defecto, basta con indicarlo con el atributo `checked`. Al haber varios botones de radio con el mismo nombre, se debe utilizar el atributo `value` para saber cuál de ellos es el que está seleccionado cuando los pares nombre valor lleguen como entrada al programa que se invoca pulsando el botón “submit”

- Uno de los campos más utilizados en los formularios es el de **texto**. Permite agregar áreas en las que la gente pueda empezar a introducir datos en el formulario. Se utilizan para la información que se introduce en una sola línea, como su nombre, una dirección, un número de teléfono y una dirección de correo electrónico. Este tipo de campo se incluye en una página HTML de la siguiente manera:
  - `<INPUT type="text" name="nombre" size="numero" maxlength="numero">`
- El atributo `size` permite determinar el tamaño del campo en caracteres. Además, con el atributo `maxlength` se puede indicar un número máximo de caracteres que se permite que el usuario introduzca en un campo de texto. El nombre es necesario para que cuando el formulario sea devuelto, el receptor comprenda la información relacionada con él. Por ejemplo, si denomina a este campo “nombre”, el formulario procesado dará lugar a algo como nombre:Marta.
- Debe tener en cuenta que un campo texto puede recibir más caracteres de los que admite su tamaño visual. Esto

va a ocurrir siempre que el valor del campo size sea menor que el valor del campo maxlength.

- En el ejemplo anterior vemos varios campos de texto
  - [Ejemplo85.txt](#)
  - [ejemplo85.html](#)
- También podemos hacer que el texto introducido no sea reconocible, es decir, que cuando escribimos aparezca \*. Para ello basta con cambiar la etiqueta Input type="text" por Input type="password"
- Otro potente método para ofrecer elecciones a sus visitantes es un menú con opciones. Éstos son especialmente útiles cuando tiene muchos elementos en una lista, como una lista de estados o países.
- Las listas desplegables se crean utilizando el elemento SELECT y las etiquetas individuales OPTION. Hay varios atributos que pueden modificar ambos elementos.
- En el área que quiere colocar una etiqueta desplegable debe introducir, en primer lugar, el texto que quiere visualizar y las etiquetas SELECT de apertura y cierre. En el atributo size establece el tamaño por número de elementos que va a contener la lista. Si éste coincide con el número de elementos reales, entonces el menú se mostrará como un cuadro de lista, con todas las opciones visibles en pantalla. Por el contrario, si especifica un valor numérico menor que el de las opciones disponibles, aparecerá una barra de desplazamiento en el menú.
- El método más popular es dejar aparte este atributo para que el menú aparezca como una familiar lista desplegable.
- También puede permitir que se seleccionen varios elementos. Para ello debe agregar el atributo multiple a la etiqueta SELECT.
- Debe añadir ahora una etiqueta OPTION para cada una de las opciones que desea que aparezcan en el menú. Aquí se añade el atributo name con cada una de estas etiquetas para asegurar que los resultados del formulario muestren la opción seleccionada. Puede utilizar el atributo selected en la opción que quiera establecer como predeterminada en el menú de selección.
- El listado siguiente muestra un ejemplo:



- `<FORM` `method="post"`  
`action="mailto:virginia.carretero@educa.madrid.org"`  
`>`
- Autonomía:
- `<SELECT>`
  - `<OPTION name="madrid">Madrid`
  - `<OPTION name="andalucia">Andalucia`
  - `<OPTION name="castilla y león"`  
`selected>Castilla y León`
- `</SELECT>`
- `</FORM>`
- Los campos de texto resultan útiles para introducir datos en una línea. Sin embargo, si quiere ofrecer a sus visitantes más espacio con la intención de que añadan sus propias respuestas e ideas, puede proporcionarles un área de texto con `TEXTAREA`.
- Las áreas de texto se crean con la etiqueta `<TEXTAREA>` y su acompañante de cierre `</TEXTAREA>`. También, puede añadir atributos como `rows` y `cols` para determinar la cantidad de filas y columnas, respectivamente, de su área de texto.
- En el ejemplo anterior vemos un área de texto
  - [Ejemplo85.txt](#)
  - [Ejemplo85.html](#)
- El último tipo de campo que permite la etiqueta `<INPUT>` es el **oculto**. Cuando se incluye un campo oculto en un formulario, éste no es visible. Este tipo de campo puede ser útil, **por ejemplo**, el identificador del usuario que envía los datos.
- Para incluir un campo oculto se emplea la entrada `INPUT` con el atributo y valor `type = "hidden "` para ser invisibles dentro del formulario.
- Por ejemplo, si se está procesando un correo electrónico a través de un formulario y se desea ocultar el valor de la línea `from` del correo procesado se pondría:
  - `<INPUT name="from" type="hidden" value="site visitor">`
- Esta información insertará las palabras "site visitor" en la línea `from` de cualquier correo electrónico procesado por el formulario. El siguiente ejemplo es similar:

- `<INPUT name="subject" type="hidden" value="Feedback: Moly's Web Site">`
- En este caso, la línea subject del correo electrónico de este formulario aclarará que tiene que ver con respuestas al sitio Web de la autora.
- Cualquiera de los campos que se pueden introducir en un formulario con la etiqueta `<INPUT>` puede estar desactivado, usando el atributo `disabled`. Por ejemplo, puede ser interesante en alguna situación desactivar alguno de los botones de un formulario, no permitiendo así que el usuario lo pulse.
- En el siguiente ejemplo que ya hemos visto, contiene algunas de las etiquetas hasta aquí explicadas.
- Únicamente incluye un formulario. Si el usuario pulsa el botón etiquetado con Buscar, se invoca el correo electrónico cuya dirección es `virginia.carretero@educa.madrid.org`, pasándole como entrada los pares nombre-valor de todos los campos que se componen el formulario.
  - [Ejemplo84.html](#)
- Los tres tipos de botones descritos en la etiqueta `<INPUT>` pueden incluirse en un formulario con otra etiqueta: `<BUTTON>`. Además, se puede solicitar información a un usuario con campos de selección y áreas de texto, que la etiqueta `INPUT` no incluye.
- Con `<BUTTON>` se pueden definir los tres tipos de botones vistos con `<INPUT>`:
  - `<BUTTON type="submit">Enviar</BUTTON>`
  - `<BUTTON type="reset">Borrar </BUTTON>`
  - `<BUTTON type="button" >Función1</BUTTON>`
- La única diferencia viene marcada por la sintaxis. Si se utiliza `<BUTTON>` no debe olvidarse la etiqueta de cierre. Además, la etiqueta del botón viene determinada por el contenido que se introduzca entre las etiquetas `<BUTTON>` y `</BUTTON>`.
- Por último, los elementos de un formulario pueden ir etiquetados con `<LABEL>`. Esta etiqueta no sólo permite asignar texto a un elemento de un formulario, sino que comparte el control del mismo. Así, si se desea que la

selección de un campo tipo radio se pueda realizar también pulsando la etiqueta que lo acompaña, basta con incluir:

- `<INPUT type="radio" name="forma_pago" value="tarjeta" id="tarj">`
- `<LABEL for="tarj">Tarjeta</LABEL>`
- Este caso aparece en el ejemplo85.html, visto anteriormente. Los elementos descritos en este apartado también pueden estar desactivados si así se precisa, no pudiendo el usuario actuar sobre ellos. El atributo que permite este efecto es disabled. Los dos siguientes ejemplos incluyen las etiquetas aquí expuestas y algunas de las descritas en el apartado anterior.
  - [Ejemplo86.txt](#)
  - [Ejemplo86.html](#)

[Ejemplo85.html](#)

- Las últimas etiquetas relacionadas con los formularios son `<FIELDSET>` y `<LEGEND>`. No añaden nuevos elementos de entrada, pero permiten un diseño más aparente de las páginas. `<FIELDSET>` es una etiqueta que agrupa elementos de un formulario, creando una caja alrededor de los mismos. `<LEGEND>` permite dar un título a la caja que dibuja `<FIELDSET>`. El título puede ubicarse en la parte superior (top), inferior (bottom), a la izquierda (left) o a la derecha (right) de la caja usando el atributo align. Ambas etiquetas precisan de la utilización de la etiqueta de fin.
  - [Ejemplo87.txt](#)
  - [Ejemplo87.html](#)