

¿Qué es un formulario?

Un formulario es la única solución que brinda el lenguaje HTML a la necesidad de los dueños de sitios web de solicitarle información al usuario de forma efectiva.

Donde necesitamos que el usuario nos proporcione sus datos de contacto, que envíe curriculums, que opine sobre encuestas y votaciones, que publique un mensaje en un foro, que busque en un buscador, que se registre como usuario, etc., en todos estos casos, se usan los formularios HTML.

Los objetivos de los formularios son dos:

Presentar "controles" en la pantalla (zonas de escritura, listas desplegadas y botones para elegir cosas) para que el usuario escriba o elija algo;

Hacer que esos datos escritos o elegidos por el usuario en su pantalla (en su casa, en su monitor), lleguen hasta el servidor web (hosting).

Los datos escritos o seleccionados por el usuario dentro de una zona de formulario, siguen el sentido contrario al de una página web normal: son datos que van desde "la casa" del usuario, desde su "pantalla", su navegador, su teclado y su mouse, hacia el hosting.

El lenguaje HTML, combinado con el protocolo HTTP que comunica a un navegador con un servidor web, solo es capaz de "transportar" los datos hasta el servidor. Pero es imprescindible que, una vez recibidos en el servidor esos datos que el usuario envía, exista algún programa EN EL SERVIDOR que "haga algo" con esos datos (que los envíe por email al dueño del sitio, que los almacene en una base de datos, etc.). Esos programas que deben estar instalados en el hosting para que los formularios funcionen son los intérpretes de lenguajes de servidor, como PHP, ASP, etc.

La etiqueta form y sus atributos

El primer elemento que indica que estamos en una "zona de formulario" es la etiqueta "form":

```
<form></form>
```

Todo lo que el usuario escriba o elija en aquellos controles o "elementos" de formulario

que coloquemos en el código fuente encerrados entre esas dos etiquetas de apertura y cierre del formulario, será enviado hacia el servidor cuando el usuario despache el formulario. Esta etiqueta no produce ningún efecto visual, es invisible, solo delimita el inicio y fin de una zona de recolección de datos. Acotemos que puede haber más de un formulario por página, cada uno delimitado por su respectiva etiqueta "form".

Esta etiqueta "form" tiene dos atributos obligatorios, que son fundamentales para que el formulario realmente funcione.

El primero de esos atributos es "action", y su valor debe ser la URL de la página programada en lenguaje de servidor (como PHP) que "procesará" los datos provenientes del formulario.

```
<form action="resultados.php">
```

Pensémoslo como un enlace hacia la siguiente página a la que nos llevará. No sirve que esa URL apunte hacia una página HTML común, ya que se perderían los datos. DEBE apuntar hacia una página programada en lenguajes de servidor como PHP (nos damos cuenta por la extensión de la página, que no es ".html" sino que es ".php").

El otro atributo fundamental es "method" y no vamos a entrar en muchos detalles por ser importante solo para programación, no para diseño, pero digamos que debe tomar uno de dos valores posibles: "get" o "post", y usualmente se utiliza "post" en un 99% de los casos.

Entonces, va quedando:

```
<form action="resultados.php" method="post">
```

Pero si ejecutamos este código en un navegador, todavía no veremos nada en la pantalla. Para poder permitir que el usuario ingrese textos o elija opciones, debemos comenzar a agregar dentro de esta etiqueta "form", otras etiquetas que

mostrarán elementos de formulario (también llamados "campos" o "controles").

Pero ante todo, como todos los campos de formulario son elementos "en línea", siempre utilizaremos algún elemento para envolver todos los campos

Campos de texto

Los campos de texto se utilizan para que el usuario escriba. Son uno de los más comunes tipos de campos, como los utilizados en buscadores (Google), y se emplean siempre que necesitemos que el usuario escriba datos que sería imposible conocer de antemano (como su nombre y apellido, su email, comentarios, etc.).

Su apariencia es ésta:

Esto es un campo de texto:

La etiqueta HTML que se usa para crear ese campo es "input". Y el principal atributo que nunca debe faltar en un campo de texto es el atributo "type", con el valor "text", que indica que ese campo es justamente un campo de texto, y no otro de los posibles campos que también se crean con la misma etiqueta "input", que es común a varios otros elementos de formulario.

Otro atributo imprescindible es "name", ya que este atributo es el que servirá para identificar lo escrito por el usuario dentro del campo, una vez que los datos lleguen al servidor. Es fundamental para la programación.

Se suele repetir el mismo valor dado a "name" (salvo excepciones que ya veremos) en otro atributo, el "id" (identificador), necesario para asociar un texto descriptivo al campo.

La sintaxis mínima de un campo de texto, entonces sería:

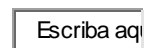
```
<input type="text" name="apellido" id="apellido" />
```

Donde lo único que podemos cambiar a gusto son los valores de los atributos "name" e "id", poniendo siempre algo que simbolice lo mejor posible los datos que se espera contenga ese campo. Por motivos de programación, los valores de las etiquetas "name" o "id" no pueden contener espacio ni acentos ni ñe, ni caracteres "raros", solo letras, números y guiones.

Otro atributo, ya no obligatorios, solo opcional, es "value", que se utiliza cuando deseamos colocar un texto que se vea escrito dentro del propio campo de texto:

```
<input type="text" name="apellido" id="apellido" value="Escriba aquí su apellido" />
```

Lo cual se verá así:



Notemos que el texto queda "cortado" en la mayoría de las resoluciones de pantalla y navegadores. Para remediarlo, se utiliza el atributo "size", que indica la cantidad de caracteres de ancho que tendrá el campo:

```
<input type="text" name="apellido" id="apellido" value="Escriba aquí su apellido" size="40" />
```

Código que generaría el siguiente resultado, bastante más legible:



Además del ancho de la caja, también podemos impedir el ingreso de una cantidad mayor al máximo que definamos con el atributo "maxlength" (que también especifica una cantidad de caracteres, pero esta vez, la cantidad máxima de caracteres que se permitirá escribir dentro del campo):

```
<input type="text" name="apellido" id="apellido" value="Escriba aquí su apellido" size="40" maxlength="10" />
```

Código que generaría el siguiente resultado (notemos que no corta el texto del "value", pero en cuanto lo borramos e intentamos escribir, nos limita a la cantidad de caracteres especificada, pruébenlo!):

Escriba aquí su apellido

Etiqueta "label"

Para mostrar un texto descriptivo al costado de cualquier campo (sea campo de texto o cualquier otro control de formulario), se debe usar la etiqueta "label". Esta etiqueta debe envolver al texto descriptivo, pero tiene dos maneras de asociar su texto con el control respectivo:

1) Asociación explícita:

Se utiliza cuando la etiqueta "label" está separada del control de formulario, usando el atributo "for" para indicar a cuál control de formulario está asociada.

Por ejemplo:

```
<label for="apellido">Escriba su apellido: </label>
<input type="text" name="apellido" id="apellido" />
```

Esto produciría el siguiente resultado:

Escriba su apellido:

Notemos que la asociación entre una etiqueta "label" y su correspondiente control se realiza haciendo coincidir el valor del atributo "for" de la etiqueta label con el valor del atributo "id" del campo, en este caso, "apellido".

De esa manera, se sabe que el texto mostrado en esa etiqueta, aunque visualmente con CSS pudiera moverse, ocultarse o alejarse, pertenece a ese campo, y será leído

justo antes de hacer foco en el campo en un lector de pantalla.

2) Asociación implícita:

Se realiza asociando una etiqueta "label" con su campo simplemente envolviendo al control con la etiqueta, cerrando el label luego del final del control:

```
<label>Escriba su apellido:  
<input type="text" name="apellido" id="apellido" />  
</label>
```

Notemos que el cierre de la etiqueta "label" está ubicado luego de finalizar el control.

3) Ambas combinadas:

Puede hacerse una combinación de ambas técnicas de asociación, explícita e implícita, si además de envolver al campo, usamos también el atributo "for" para explicitar la relación (es lo mejor, realizar AMBAS asociaciones, ya que los navegadores antiguos no interpretan la asociación explícita usando for, pero sí la implícita envolviendo el control):

```
<label for="apellido">Escriba su apellido:  
<input type="text" name="apellido" id="apellido" />  
</label>
```

Una ventaja de usar esta etiqueta es que al pulsar sobre el texto del label, el foco se centra en el campo asociado, algo muy útil en el caso de casillas y botones de selección (pueden probarlo con los botones tipo "radio" de Google, esos que dicen: Buscar en "la web", "páginas en español", etc.).

El texto del label puede colocarse antes o después del control de formulario al que va asociado.

Un ejemplo de colocar el texto del label después del campo:

```
<label for="apellido">  
<input type="text" name="apellido" id="apellido" />  
Escriba su apellido</label>
```

Lo cual se vería así:

Escriba su apellido

También puede usarse más de una etiqueta "label" por cada control de formulario, de ser necesario, siempre que la asociación sea explícita (no puede "envolverse" con la etiqueta label más de un input de manera implícita).

Áreas de texto

A diferencia de los campos de texto, en las áreas de texto puede escribirse más de un renglón, por lo cual son ideales para solicitarle al usuario datos como "comentarios", "consultas", "opiniones", mensajes de un foro, etc. donde debe expresarse a gusto.

No se utiliza la etiqueta input, sino la etiqueta "textarea", que a diferencia de input, no lleva cierre abreviado:

```
<label for="comentarios">Escriba sus comentarios:  
<textarea name="comentarios" id="comentarios"></textarea>  
</label>
```

Lo cual produciría este resultado:



Escriba sus comentarios:

Esta etiqueta no puede utilizar el atributo "value"; si queremos mostrar un texto escrito dentro del área, debemos envolver ese texto entre la apertura y el cierre de la etiqueta:

```
<label for="comentarios">Escriba sus comentarios:  
<textarea name="comentarios" id="comentarios">Díganos su opinión escribiendo  
aquí!</textarea></label>
```

Tamaño del cuadro:

Los atributos rows (filas) y cols (columnas) se utilizan para especificar la cantidad de renglones (rows) de alto, y cantidad de caracteres de ancho (cols) del área de texto; pero no nos confundamos, no pueden impedir escribir cualquier cantidad de texto dentro del textarea (no existe para los textarea un atributo similar al "maxlength" de los campos de texto):

```
<textarea name="comentarios" id="comentarios" rows="8" cols="25"></textarea>
```

Cuando uno ingresa texto en mayor cantidad de lo que puede ser visualizado, automáticamente aparecen barras de desplazamiento (scroll).

Botones para enviar

Todo lo que hemos visto hasta ahora son campos de texto de una línea o de varias (áreas de texto), la etiqueta descriptiva "label", y la etiqueta "form", delimitadora de una "zona" que recopilará información allí ingresada.

Pero nada de esto tendría sentido sin la posibilidad de "enviar" realmente los datos, dando por finalizada la carga o elección de datos, para proceder a enviarlos hacia el servidor.

Para disparar ese envío, se utiliza de nuevo la etiqueta input, pero esta vez, con otro valor en el atributo "type". El tipo de input que genera un botón de enviar es el input de tipo "submit", y esta es su sintaxis:

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

Muy simple, especificamos el "tipo" y eso ya lo vuelve uno de esos típico botones grises pulsables (por supuesto que con CSS puede cambiarse su apariencia por completo). Al pulsarlo, el navegador envía al servidor todos los datos escritos o elegidos por el usuario entre la apertura y el cierre de la etiqueta "form" dentro de la cual esté el botón submit que fue pulsado. A la vez, realiza una petición hacia la página indicada en el atributo action de la etiqueta form, como si fuera un enlace hacia esa página. Nos lleva a ella.

Si agregamos el atributo value, lo que escribamos como valor de ese atributo será el texto visible del botón.

Por ejemplo:

