

Formularios <Form>

Los formularios fueron una de las primeras mejoras que se dieron a conocer de HTML5 y de las que más revuelo causaron, sobre todo por su gran capacidad de autovalidación de una manera extremadamente fácil y sin necesidad de utilizar JavaScript.

¿Qué es un formulario?

Se conoce como formulario a los mecanismos para enviar información por parte del usuario, a través de unos campos visuales de forma sencilla e intuitiva, que determinan la naturaleza del tipo de información que se va a enviar y como hacerla llegar a otro extremo donde se procesará esa información.

Nuestros objetivos al crear un formulario HTML deben ser los siguientes:

- Hacer lo más sencillo posible el proceso de inserción de datos por parte del usuario.
- Intentar que la experiencia de usuario sea lo más agradable y cómoda posible.
- Intentar que los datos introducidos estén en un formato predecible y esperado.
- Reducir los errores al introducir datos en la medida de lo posible.
- Comunicar clara y rápidamente si ocurren errores al introducir datos.

La etiqueta <form>

Para comenzar con la creación de un formulario, el primer paso es indicar una etiqueta contenedora <form>, que incluirá toda la información que se quiere recoger en ese formulario:

<form></form>

De esta forma, incluso se pueden crear varias etiquetas <form> por página, pudiendo así crear varios formularios con diferentes objetivos,

como, por ejemplo, uno para realizar una búsqueda en la página, otro para dejar un comentario y otro para enviar un email al autor, etc.

Es posible crear formularios sin utilizar una etiqueta `<form>`, sin embargo, tendrían que manejarse desde Javascript y probablemente, tengan peor rendimiento de cara a la accesibilidad.

La etiqueta `<form>` dispone de varios **atributos** que se pueden indicar:

Atributo	Descripción	Más información
<code>action</code>	Dirección URL del backend al cuál se enviará la información obtenida en el formulario.	
<code>method</code>	Método HTTP de envío. <code>GET</code> a través de URL, <code>POST</code> para envío extenso de información.	
<code>name</code>	Nombre del formulario. Útil para identificar el formulario y procesar posteriormente.	
<code>target</code>	Nombre del lugar donde se abrirá el formulario. <code>_blank</code> para nueva pestaña.	
<code>enctype</code>	Codificación para el envío del formulario. Importante para envío de archivos.	Envío de archivos
<code>accept-charset</code>	Fuerza a utilizar una codificación en los parámetros de texto del formulario.	
<code>autocomplete</code>	Activa <code>on</code> o desactiva <code>off</code> el autocompletado para todos los campos del formulario.	
<code>novalidate</code>	BOOLEAN Desactiva la validación HTML5 del formulario.	

Ejemplo básico 1

Así pues, vamos a ver un primer formulario base de ejemplo (aún vacío, sin campos de entrada de datos, sólo con la etiqueta contenedora) para ejemplificar su utilización:

```
<form name="formulario" method="post" action="/send/"></form>
```

- Nótese que en este primer ejemplo se definen 3 atributos base en `<form>`:

- El atributo name define el nombre del formulario para diferenciarlo de otros.
- El atributo method define el método HTTP a utilizar. Usaremos GET cuando nos interese enviar datos a través de la URL. En caso contrario, POST.
- El atributo action se usa para indicar la URL a donde se enviarán los datos cuando el usuario pulse el botón de enviar. En caso de omisión, se enviarán a la propia página actual, volviéndola a cargar.

Ejemplo básico 2

Veamos un nuevo ejemplo utilizando otros campos de formulario:

```
<form method="get" action="/search/" target="_blank" accept-charset="utf-8"></form>
```

En este caso enviamos a través de método GET a la URL <https://dominio.com/search/>:

Se ha indicado el atributo target con valor _blank, esto significa que se abrirá en una nueva pestaña.

Se ha indicado el atributo accept-charset. Esto forzará a que el texto enviado en el formulario sea interpretado como codificación UTF8. Útil si tienes problemas de caracteres extraños o similar.

Ejemplo básico 3

Un último ejemplo con los atributos restantes:

```
<form method="post" action="/saveform/" autocomplete="off" novalidate></form>
```

El atributo autocomplete indica el valor off. Por defecto, este atributo está a on. Desactivado, todos los campos del formulario no mostrarán sugerencias anteriores de autocompletado.

El atributo novalidate hará que el usuario pueda saltarse todas las validaciones HTML del mismo.

Información de un formulario

Para que un usuario pueda introducir información en un formulario, requiere que se le indiquen una serie de campos que le permitan introducir la información de la forma más cómoda y rápida posible. Para ello, debemos saber a priori que tipo de dato le vamos a pedir al usuario (texto, cantidades numéricas, fechas, etc...), y así saber qué tipo de campo de entrada de datos es más conveniente utilizar.

Esto es una lista de los tipos de datos que podemos utilizar:

Información a obtener	Ejemplos	Etiqueta y atributo a utilizar
Información de texto corto	Nombres, apellidos, textos cortos...	<code><input></code>
Información de texto largo	Comentarios, observaciones...	<code><textarea></code>
Números o cantidades	Edades, precios, cuantías...	<code><input type="number"></code> ó <code><input type="range"></code>
Fechas u horas	Fecha de nacimiento, inicio de evento...	<code><input type="date"></code> o derivados
Verdadero/falso	Si/No, Opción A/B, ON/OFF...	<code><input type="checkbox"></code>
Opción única	(Elegir una opción de 2 ó más posibles)	<code><input type="radio"></code> ó <code><select></code>
Varias opciones	(Elegir varias opciones de 2 ó más posibles)	<code><select multiple></code> ó <code><input type="checkbox"></code>
Opción única abierta	(Elegir una opción o indicar una propia)	<code><datalist></code>
Selección de color	(Escoger un color o tonalidad)	<code><input type="color"></code>
Selección de archivo	(Escoger un archivo para enviar)	<code><input type="file"></code>

Cada uno de estos campos tiene particularidades específicas, por lo que conviene aprenderlos por separado y entender cómo funciona cada uno de ellos. En cada uno de esos artículos explicaremos cómo funcionan.

Sintaxis básica de un formulario

Como hemos mencionado, la etiqueta `<form>` es sólo la etiqueta contenedora del formulario. Luego, hay que ir añadiendo campos para crear un formulario extenso y personalizado. Una de las etiquetas que más utilizaremos es la etiqueta `<input>`.

Por ejemplo, en el siguiente formulario, establecemos dos etiquetas `<input>`, una de ellas para obtener información de texto y la otra para incorporar un botón para enviar los datos del formulario:

```
<form name="userdata" method="post" action="/send/">  
  <input type="text" name="username">  
  <input type="submit">  
</form>
```

En este ejemplo tenemos un pequeño formulario en el cuál:

Se pide al usuario su nombre con la etiqueta `<input type="text">`

Botón para permitir enviar el formulario con la etiqueta `<input type="submit">`

Aunque vemos que se trata de la misma etiqueta `<input>`, dependiendo del valor indicado en el atributo `type` realizará una tarea u otra. En los siguientes artículos veremos cómo crear formularios mucho más personalizados, con más información, semánticos y que sean accesibles.