Formularios y Tablas HTML

Introducción

En múltiples campos, el traspaso de información a través de formularios es una actividad cotidiana. El desarrollo web no está exento y más de una vez nos habremos topado con este tipo de recursos. A continuación, analizaremos qué sucede cuando enviamos un formulario, cómo confeccionarlos.

¿Qué son los formularios web?

Los formularios web son uno de los principales puntos de interacción entre un usuario y un sitio web o aplicación. Los formularios permiten a los usuarios la introducción de datos, que generalmente se envían a un servidor web para su procesamiento y almacenamiento. También es común su empleo del lado del cliente para provocar de alguna manera una actualización inmediata de la interfaz (por ejemplo, se añade otro elemento a una lista, o se muestra u oculta una función de interfaz de usuario).

El HTML de un formulario web está compuesto por uno o más controles de formulario (a veces llamados widgets (accesorios), además de algunos elementos adicionales que ayudan a estructurar el formulario. Los controles pueden ser campos de texto de una o varias líneas, cajas desplegables, botones, casillas de verificación o botones de opción, y se crean principalmente con el elemento <input> (entrada), aunque hay algunos otros elementos que también hay que conocer.

Los controles de formulario también se pueden programar para forzar la introducción de formatos o valores específicos (validación de formulario), y se combinan con etiquetas de texto que describen su propósito para los usuarios con y sin discapacidad visual.

Consejo previo

Antes de comenzar a escribir código, siempre es mejor dar un paso atrás y tomarte el tiempo necesario para pensar en tu formulario. Diseñar una maqueta rápida te ayudará a definir el conjunto de datos adecuado que deseas pedirle al usuario que introduzca. Desde el punto de vista de la experiencia del usuario (UX), es importante recordar que cuanto más grande es tu formulario, más te arriesgas a frustrar a las personas y perder usuarios. Tiene que ser simple y conciso: solicita solo los datos que necesitas.

Diseño del formulario

Supongamos un sistema breve de comentarios de un blog abierto para el diseño. Esto implica solicitar mail, nombre y el texto en cuestión. Por lo que emplearemos tres campos de texto y un botón. Al pulsar el botón sus datos se enviarán a un servidor web.

El paso siguiente es llevar esto que acabamos de mencionar a HTML. Para este caso seleccionamos los siguientes elementos HTML: <form>, <label>, <input>, <textarea> y <button>.

Ya en la introducción a html vimos que todos los formularios comienzan con un elemento <form> como este:

<form action="/my-handling-form-page" method="post">

</form>

Este elemento define formalmente un formulario. Es un elemento contenedor, como un elemento
<section> o <footer>, pero específico para contener formularios. También admite algunos atributos específicos para la configuración de la forma en que se comporta el formulario. Todos sus atributos son opcionales, pero es una práctica estándar establecer siempre al menos los atributos action y method. Action define la ubicación (URL) donde se envían los datos que el formulario ha recopilado cuando se validan. Mientras que method define con qué método HTTP se envían los datos (generalmente get o post).

El campo de entrada para el nombre es un campo de texto de una sola línea.
El campo de entrada para el correo electrónico es una entrada de datos de tipo correo electrónico: un campo de texto de una sola línea que acepta solo direcciones de correo electrónico.
El campo de entrada para el mensaje es <textarea>; un campo de texto multilínea.

En términos de código html, para implementar estos controles de formulario necesitamos algo como lo siguiente:

```
<form action="/my-handling-form-page" method="post">
 <ul>
  <li>
    <label for="name">Nombre:</label>
    <input type="text" id="name" name="user_name">
  </li>
  <li>
    <label for="mail">Correo electrónico:</label>
    <input type="email" id="mail" name="user_mail">
  </li>
  <li>
    <label for="msg">Mensaje:</label>
    <textarea id="msg" name="user_message"></textarea>
  </li>
 </ul>
</form>
```

Los elementos **<li>** se emplean para estructurar nuestro código convenientemente y facilitar la aplicación de estilo. Por motivos de usabilidad y accesibilidad incluimos una etiqueta explícita para cada control de formulario. Ten en cuenta el uso del atributo “for” en todos los elementos <label>, que toma como valor el id del control de formulario con el que está asociado; así es como asociar un formulario con su etiqueta.

Hacer esto presenta muchas ventajas porque la etiqueta está asociada al control del formulario y permite que los usuarios con mouse, touchpad y otros dispositivos táctiles hagan clic en la etiqueta para activar el control correspondiente, y también proporciona accesibilidad con un nombre que los lectores de pantalla leen a sus usuarios.

En el elemento **<input>**, el atributo más importante es **type**. Este atributo es muy importante porque define la forma en que el elemento **<input>** aparece y se comporta.
En nuestro ejemplo, usamos el valor **<input/text>** para la primera entrada, el valor predeterminado para este atributo. Representa un campo de texto básico de una sola línea que acepta cualquier tipo de entrada de texto.
Para la segunda entrada, usamos el valor <input/email>, que define un campo de texto de una sola línea que solo acepta una dirección de correo electrónico. Esto convierte un campo de texto básico en una especie de campo «inteligente» que efectúa algunas comprobaciones de validación de los datos que el usuario escribe.

También hace que aparezca un diseño de teclado más apropiado para introducir direcciones de correo electrónico (por ejemplo, con un símbolo @ por defecto) en dispositivos con teclados dinámicos, como teléfonos inteligentes. Encontrarás más información sobre la validación de formularios en el artículo de Validación de formularios por parte del cliente más adelante.

Por último, debemos observar la sintaxis de <input> en contraposición con la de <textarea></textarea>. Esta es una de las rarezas del html. La etiqueta <input> es un elemento vacío, lo que significa que no necesita una etiqueta de cierre. El elemento <textarea> no es un elemento vacío, lo que significa que debe cerrarse con la etiqueta de cierre adecuada. Esto tiene un impacto en una característica específica de los formularios: el modo en que defines el valor predeterminado. Para definir el valor predeterminado de un elemento <input>, debes usar el atributo value de esta manera:

<input type="text" value="por defecto este elemento se llena con este texto">

Por otro lado, si deseas definir un valor predeterminado para un elemento <textarea>, lo colocas entre las etiquetas de apertura y cierre del elemento <textarea>, así:

<textarea>
Por defecto, este elemento contiene este texto
</textarea>

Elemento <button>

El marcado de nuestro formulario está casi completo. Solo resta añadir un botón para permitir que el usuario envíe sus datos una vez que haya completado el formulario. Esto se hace con el elemento
<button>. Añadimos lo siguiente justo encima de la etiqueta de cierre </form>:

<li class="button">
<button type="submit">Envíe su mensaje</button>

El elemento <button> también acepta un atributo de type, que a su vez acepta uno de estos tres valores: submit, reset o button.

- Un click en un botón submit (el valor predeterminado) envía los datos del formulario a la página web definida por el atributo action del elemento <form>.

- Un click en un botón reset restablece de inmediato todos los controles de formulario a su valor predeterminado. Desde el punto de vista de UX, esto se considera una mala práctica, por lo que debes evitar usar este tipo de botones a menos que realmente tengas una buena razón para incluirlos.
- Un clic en un botón botón no produce efecto alguno a priori, pero es de suma utilidad para crear botones personalizados, cuya lógica está desarrollada con JavaScript.

Cabe mencionar que también se puede usar el elemento <input> con el atributo type correspondiente para generar un botón, por ejemplo <input type="submit">.

La ventaja principal del elemento <button> es que el elemento <input> solo permite texto sin formato en su etiqueta, mientras que el elemento <button> permite contenido html completo, lo que permite generar botones creativos más complejos.

Enviar los datos del formulario a un servidor web

La última parte, y quizás la más complicada, es manejar los datos del formulario en el lado del servidor. El elemento <form> define dónde y cómo enviar los datos gracias a los atributos action y method.

Proporcionamos un nombre (name) a cada control de formulario. Los nombres son importantes tanto en el lado del cliente como del servidor, le dicen al navegador qué nombre debe dar a cada dato mientras que en el lado del servidor, cada dato es manejado por su nombre. Los datos del formulario se envían al servidor como pares de nombre/valor.

Para poner nombre a los diversos datos que se introducen en un formulario, debes usar el atributo name en cada control de formulario que recopila un dato específico. Veamos de nuevo algunos de nuestros códigos de formulario:

```
<form action="/my-handling-form-page" method="post">
 <ul>
  <li>
   <label for="name">Nombre:</label>
   <input type="text" id="name" name="user_name" />
  </li>
  <li>
   <label for="mail">Correo electrónico:</label>
   <input type="email" id="mail" name="user_email" />
  </li>
  <li>
   <label for="msg">Mensaje:</label>
   <textarea id="msg" name="user_message"></textarea>
  </li>
```

En nuestro ejemplo, el formulario envía tres datos denominados «user_name», «user_email» y «user_message». Esos datos se envían a la URL «/my-handling-form-page» utilizando el método post de HTTP.

En el lado del servidor, la secuencia de comandos de la URL «/my-handling-form-page» recibe los datos como una lista de tres elementos clave/valor contenidos en la solicitud HTTP.

Tablas

Etiqueta <table> en HTML

La etiqueta <table> en HTML se utiliza para crear tablas y organizar datos en filas y columnas. Es una de las etiquetas fundamentales para estructurar contenido tabular en una página web.

Estructura básica de una tabla
Para crear una tabla, se siguen los siguientes pasos:

- Abre la etiqueta <table>, que indica el inicio de la tabla.
- Dentro de <table>, se agregan una o más filas utilizando la etiqueta <tr> (table row).
- Dentro de cada fila, se agregan las celdas utilizando la etiqueta <td> (table data).
- Es posible utilizar la etiqueta <th> (table header) dentro de la fila <tr> para definir encabezados de columna.

Table 1 : Ejemplo de una tabla simple con tres filas y tres columnas

```
<table>
    <tr>
      <th>Nombre</th>
      <th>Edad</th>
      <th>País</th>
    </tr>
    <tr>
      <td>Juan</td>
      <td>25</td>
      <td>Argentina</td>
    </tr>
    <tr>
      <td>María</td>
      <td>30</td>
      <td>España</td>
    </tr>
  </table>
```

En este ejemplo, las etiquetas <th> se usan para los encabezados de columna y las etiquetas <td> para los datos de la tabla.

Atributos de la etiqueta <table>

La etiqueta <table> también admite varios atributos para personalizar su apariencia y comportamiento. Algunos de los atributos más comunes son:

- border: especifica el ancho del borde de la tabla;
- width: especifica el ancho de la tabla;
- bgcolor: especifica el color de fondo de la tabla.

Otras etiquetas relacionadas

Además de <table>, existen etiquetas relacionadas que se pueden utilizar junto con las tablas para proporcionar estructura y contexto:

- <caption>: se coloca dentro de <table> y se utiliza para agregar un título o descripción a la tabla;
- <thead>, <tbody> y <tfoot>: se utilizan para agrupar las filas de la tabla en secciones como encabezado, cuerpo y pie de tabla, respectivamente.

Estas etiquetas adicionales pueden mejorar la accesibilidad y el diseño de la tabla, brindando más información y organización al contenido.