

Unidad Formativa UF3: Elaboración de plantillas y formularios

Acción Formativa: Web Development Bootcamp

Autor: Javier Gil Motos - EOI

Índice

Contenido

Índice.....	2
Qué es un formulario	4
Controles	5
Atributos comunes de los controles.....	6
Tipos de controles	8
Botones: <code><input /></code> y <code><button></code>	8
Botones Submit y Reset mediante <code><input /></code>	8
Botones Submit y Reset mediante <code><button></code>	9
Caja de texto: <code><input type="text" /></code> , <code><input type="password" /></code> y <code><textarea></code> 10	
Caja de texto de una sola línea: <code><input type="text" /></code> y <code><input type="password" /></code>	10
Caja de texto de varias líneas: <code><textarea></code>	11
Casilla de verificación: <code><input type="checkbox" /></code>	12
Botón radio: <code><input type="radio" /></code>	12
Menú: <code><select></code>	13
Selector de archivo: <code><input type="file" /></code>	14
Control oculto: <code><input type="hidden" /></code>	15
Grupos de controles: <code><fieldset></code>	15
Accesibilidad: <code><label></code>	15
Otros controles.....	16
Selector de color: <code><input type="color" /></code>	16
Número: <code><input type="number" /></code>	16
Rango: <code><input type="range" /></code>	17
Fecha: <code><input type="date" /></code>	17

Correo electrónico: <code><input type="email" /></code>	18
Nuevos atributos de controles en HTML5	18
Bibliografía	20

Qué es un formulario

Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc.) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.

La etiqueta que delimita un formulario es la etiqueta `<form> ... </form>`. Los atributos más importantes de la etiqueta `<form>` son:

- **action:** contiene el nombre del agente que procesará los datos remitidos al servidor (por ejemplo, un script de PHP)
- **method:** define la manera de enviar los datos al servidor. Los valores posibles son:
 - get: los valores enviados se añaden al final de la dirección indicada en el atributo `action`.
 - post: los valores se envían de forma separada.

Si el atributo `method` no está establecido, Internet Explorer y Firefox se comportan como si el valor fuera `get`.

La etiqueta `<form>` es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.

Los etiquetas más típicas que crean los controles en los formularios son `<input />`, `<button>`, `<select>`, `<optgroup>`, `<option>` y `<textarea>`. Además, se pueden estructurar los controles con las etiquetas `<fieldset>` y `<legend>`. Por último, la etiqueta `<label>` permite mejorar la accesibilidad de los controles.

El navegador envía únicamente los datos de los controles contenidos en el formulario (siempre que tengan un `name` establecido). En una misma página puede haber varios formularios que envíen datos al mismo o a diferentes agentes.

Aquí se muestra un ejemplo de un formulario simple que incluye rótulos, radiobuttons y botones para reinicializar el formulario o para enviarlo:

```
<form action="http://equipo03/formularios/login" method="post">
  <div>
    <label for="nombre">Nombre: </label>
    <input type="text" name="nombre" id="nombre"><br>

    <label for="apellidos">Apellidos: </label>
    <input type="text" name="apellidos" id="apellidos"><br>

    <input type="radio" name="sexo" value="V"> Varón<br>
    <input type="radio" name="sexo" value="M"> Mujer<br>

    <input type="submit" value="Enviar"> <input type="reset">

  </div>
</form>
```

Controles

Los usuarios interaccionan con los formularios a través de los llamados *controles*. El "*nombre de control*" de un control viene dado por su atributo `name`. Cada control tiene tanto un valor inicial como un valor actual, que son ambas cadenas de caracteres (es recomendable consultar la definición de cada control para obtener información sobre los valores iniciales y las posibles restricciones que puede imponer cada control sobre sus valores). En general, el "*valor inicial*" de un control puede especificarse con el atributo `value` del elemento de control. Sin embargo, el valor inicial de un elemento `textarea` viene dado por sus contenidos, y el valor inicial de un elemento `object` de un formulario está determinado por la implementación del objeto (es decir, se sale fuera del alcance de estos apuntes).

El "*valor actual*" del control se rellena en primer lugar con el valor inicial y, a partir de ese momento, el valor actual del control puede ser modificado a través de la interacción con el usuario (introduciendo un texto, por ejemplo) y mediante scripts. El valor inicial de un control no cambia; así, cuando se reinicializa el formulario (por ejemplo al recargar la página), el valor actual de cada control se

reinicializa a su valor inicial. Si el control no tiene un valor inicial, el efecto de una reinicialización del formulario sobre ese control es indefinido. Cuando se envía un formulario para su procesamiento, para algunos controles se empareja su nombre con su valor actual, y estas parejas se envían con el formulario.

A modo de resumen, podríamos decir que en un formulario habrá una serie de controles (cajas de texto, checkboxes, radiobuttons, etc.) y cuando enviemos el formulario (por ejemplo, haciendo click en un botón submit) se enviarán al destino del formulario los nombres y los valores de los controles en forma de parejas.

Ejemplo:

```
<form action="http://equipo03/formularios/login" method="post">
  <div>
    <label for="nombre">Nombre: </label>
    <input type="text" name="nombre" id="nombre"><br>

    <label for="apellidos">Apellidos: </label>
    <input type="text" name="apellidos" id="apellidos"><br>

    <input type="radio" name="sexo" value="V"> Varón<br>
    <input type="radio" name="sexo" value="M"> Mujer<br>

    <input type="submit" value="Enviar"> <input type="reset">
  </div>
</form>
```

Si hacemos click en el botón “Enviar” del formulario, se enviarán los siguientes datos:

Destino	<code>http://equipo03/formularios/login</code>
nombre	El valor introducido por el usuario en el campo “nombre”.
apellidos	El valor introducido por el usuario en el campo “apellidos”
sexo	“V” o “M”, dependiendo de lo seleccionado por el usuario.

Atributos comunes de los controles

name, value, disabled, readonly, tabindex y accesskey

El atributo `name` identifica al control. El formulario únicamente envía al servidor los datos de los controles que tienen establecido el atributo `name`, por tanto si un

control no tiene establecido su atributo `name`, el control simplemente se ignora. En principio, y salvo en el caso de los radiobuttons, no deberían existir dos controles en la misma página que tengan un atributo `name` idéntico.

Nota: dado que la función del atributo `name` es prácticamente la misma que la del atributo `id`, es muy habitual dar el mismo valor a ambos atributos (salvo en el caso de los radiobuttons).

El atributo `value` permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta. El único control sin atributo `value` es el área de texto (<textarea>).

En algunos casos (botones) el valor se muestra al usuario y el usuario no puede modificarlo. En otros casos (cajas de texto, contraseña) el valor se muestra en la página y el usuario puede modificarlo. En otros casos (casillas de verificación, botones radio, opciones de menú, oculto, imagen) el valor no se muestra en la página y el usuario no puede modificarlo.

En algunos casos (botones, oculto, botones radio) el atributo `value` es necesario. En otros casos (casilla de verificación, opciones de menú) no es obligatorio establecerlo, si bien es conveniente hacerlo. En otros casos (cajas de texto, contraseña, imagen) puede omitirse si no deseamos que el control tenga un valor establecido por defecto.

El atributo `disabled` permite deshabilitar el control. Una vez deshabilitado, el control ni siquiera puede coger el foco. Además, cuando se envíe el formulario, el valor de los elementos deshabilitados no se enviará.

El atributo `readonly` permite que el control no sea modificable, aunque el control puede coger el foco. El usuario puede, incluso, copiar el contenido del control (en el caso de las cajas de texto), aunque no puede cambiarlo.

El atributo `tabindex` permite controlar el orden en que el foco pasa de un elemento a otro mediante el tabulador (`Tab` para avanzar y `Shift+Tab` para retroceder). Los valores de `tabindex` pueden ser números naturales (incluido el cero), no necesariamente consecutivos. Si no está presente este atributo, los controles se visitan en el orden en que aparecen en el texto. Si está presente, los controles se visitan de menor a mayor.

El atributo `accesskey` permite definir teclas de acceso (atajos de teclado). El problema con las teclas de acceso es que a veces entran en conflicto con combinaciones de teclas ya definidas por el navegador o el sistema operativo. Para acceder a las teclas de acceso, cada navegador utiliza una combinación de teclas distinta:

- Firefox utiliza la combinación `Alt+Shift+tecla`
- Internet Explorer utiliza la combinación `Alt+tecla` o `Alt+Shift+tecla`
- Google Chrome utiliza normalmente la combinación `Alt+tecla`, pero cuando esa combinación ya está asociada a alguna acción usa la combinación `Alt+Shift+tecla`

Tipos de controles

Botones: `<input />` y `<button>`

Los botones se crean mediante la etiqueta `<button>`, aunque por compatibilidad con versiones anteriores de HTML, se pueden crear también con la etiqueta `<input />`.

Normalmente los botones no suelen llevar atributo `name`, pero se puede poner si se quiere (tendría sentido si un formulario contuviera dos botones de envío y quisiéramos saber en cuál de ellos ha hecho click el usuario, pero esto no es muy habitual).

Botones Submit y Reset mediante `<input />`

El botón Submit es el que permite al usuario enviar los datos del formulario al servidor. Se crea mediante una etiqueta `<input />` cuyo atributo `type` tiene el valor `submit`. El texto que se muestra en el botón se define mediante el atributo `value`.


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

Submit

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

Enviar

El botón Reset restablece los valores iniciales del formulario. Se crea mediante una etiqueta `<input />` cuyo atributo `type` tiene el valor `reset`. El texto que se muestra en el botón se define mediante el atributo `value`.

```
<input type="reset" value="Reset" />
```

Reset

```
<input type="reset" value="Borrar todo" />
```

Borrar todo

El atributo `value` sólo puede contener texto, no imágenes.

Botones Submit y Reset mediante `<button>`

La etiqueta `<button>` permite crear botones de tipo `submit` o `reset` o botones de tipo general que deben asociarse a scripts para hacer algo. Los botones `submit` o `reset` se crean mediante el atributo `type` con el valor `submit` o `reset`. El botón de uso general se crea mediante el atributo `type` con el valor `button`.

<code><button type="submit">Enviar</button></code>	<input type="submit" value="Enviar"/>
<code><button type="reset">Borrar todo</button></code>	<input type="reset" value="Borrar todo"/>
<code><button type="button">Botón</button></code>	<input type="button" value="Botón"/>

Los botones `<button>` pueden contener texto e imágenes (o estructuras más complejas, pero no mapas de imágenes).

<pre> <button> Botón </button> </pre>	<input type="button" value="Botón"/>
---	--------------------------------------

Si un botón no lleva el atributo `type`, se comporta como un botón de tipo submit.

<code><button>Botón</button></code>	<input type="submit" value="Botón"/>
---	--------------------------------------

Caja de texto: `<input type="text" />`, `<input type="password" />` y `<textarea>`
Existen dos tipos de cajas de texto: de una sola línea y de varias líneas

Caja de texto de una sola línea: `<input type="text" />` y `<input type="password" />`

Las cajas de texto de una sola línea se crean mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `text`.

```
<input type="text" name="texto" />
```

El atributo `value` (optativo) contiene el valor inicial de la caja de texto. El atributo `size` indica el tamaño en caracteres de la caja en la pantalla (por omisión, las cajas suelen tener 20 caracteres de tamaño) si bien es recomendable no usar este atributo y establecer el tamaño de las cajas de texto con las propiedades CSS `width` y `height`. El atributo `maxlength` indica el número máximo de caracteres que puede escribir el usuario en la caja de texto.

```
<input type="text" name="texto" maxlength="5" />
```

Existe una caja de texto de una sola línea especial para escribir contraseñas que se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `password`.

```
<input type="password" name="contrasena" />
```

Al escribir en una caja de contraseña, en vez de letras aparecen puntos gruesos (o asteriscos). Es importante señalar que estas cajas no proporcionan ninguna seguridad en la transmisión, simplemente ocultan al usuario lo que éste escribe.

Caja de texto de varias líneas: `<textarea>`

Las cajas de texto de varias líneas se crean mediante la etiqueta `<textarea>`. Si deseamos que la caja de texto contenga algún valor por defecto, deberemos colocarlo entre las etiquetas `<textarea>` y `</textarea>`.

```
<textarea name="texto" >Escribe algo</textarea>
```

Escribe algo

Casilla de verificación: `<input type="checkbox" />`

Las casillas de verificación se crean mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `checkbox`.

```
<input type="checkbox" name="casilla" />Casilla 1
```

☐ Casilla 1

Si el atributo `checked` está establecido, la casilla aparece marcada.

```
<input type="checkbox" name="casilla" checked />Casilla 1
```

☒ Casilla 1

Las casillas de verificación sólo se envían si se han marcado. El atributo `value` contiene el valor que envía el formulario si la casilla de verificación está marcada. Si el atributo `value` no está establecido, el formulario envía el valor "on".

Botón radio: `<input type="radio" />`

Los botones radio se crean mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `radio`.

```
<input type="radio" name="radio" value="1" />Opción 1
```

☐ Opción 1

Los botones radio que tienen el mismo atributo `name` forman un grupo, es decir, que si se marca uno de ellos se desmarca automáticamente el resto.

```
<input type="radio" name="radio" value="1" />Opción 1  
<input type="radio" name="radio" value="2" />Opción 2
```

☐ Opción 1
☐ Opción 2

Los dos ejemplos anteriores, aunque estén separados, forman el mismo botón radio ya que su atributo `name` tiene el mismo valor (en este caso "radio"). Se puede comprobar en una página cualquiera que haga uso de estos botones de radio, pulsando en uno y observando cómo se desmarca el otro ejemplo. Para que fueran independientes, bastaría con que sus atributos `name` fueran distintos.

Si uno de los botones tiene el atributo `checked`, el botón aparece marcado.

```
<input type="radio" name="radio3" value="1" />Opción 1<br />  
<input type="radio" name="radio3" value="2" checked />Opción 2
```

☐ Opción 1
☒ Opción 2

Los botones radio sólo se envían si se han marcado. El atributo `value` contiene el valor que envía el formulario si el botón radio está marcado. Si el atributo `value` no está establecido, el formulario envía el valor "on", así que para poder saber cuál ha sido la opción elegida por el usuario es necesario establecer con valores distintos los atributos `value` de todos los elementos de un botón radio.

Menú: <select>

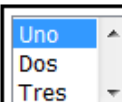
Los menús se crean mediante la etiqueta `<select>`. Cada opción del menú se define mediante la etiqueta `<option>`. El atributo `selected` indica la opción por omisión.

```
<select name="menu">  
  <option selected value="1">Uno</option>  
  <option value="2">Dos</option>  
  <option value="3">Tres</option>  
</select>
```

Uno
Uno
Dos
Tres

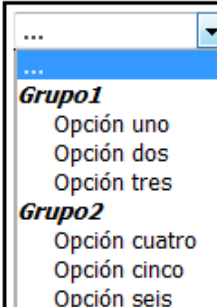
El atributo `size` permite definir la altura del control. Si usamos este atributo, el menú ya no será un “combo desplegable”, sino una lista con tantos elementos visibles como indique el atributo `size`, mientras que el resto de elementos serán accesibles mediante la barra de scroll del elemento `<select>`.

```
<select name="menu" size="3">
  <option value="1" selected>Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
</select>
```



Se pueden agrupar opciones utilizando la etiqueta `<optgroup>`.

```
<select name="menu">
  <optgroup label="Grupo1">
    <option value="1">Opción uno</option>
    <option value="2">Opción dos</option>
    <option value="3">Opción tres</option>
  </optgroup>
  <optgroup label="Grupo2">
    <option value="4">Opción cuatro</option>
    <option value="5">Opción cinco</option>
    <option value="6">Opción seis</option>
  </optgroup>
</select>
```

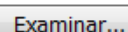


El atributo `value` de cada opción contiene el valor que envía el formulario si la opción está elegida. Si no se define el atributo `value`, el formulario envía como valor el texto que aparece en el menú.

Selector de archivo: `<input type="file" />`

El selector de archivo se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `file`. En esta etiqueta, el atributo `value` se ignora si tiene algún valor predefinido y no surte ningún efecto.

```
<input type="file" name="archivo" />
```



No se ha seleccionado ningún archivo.

Control oculto: `<input type="hidden" />`

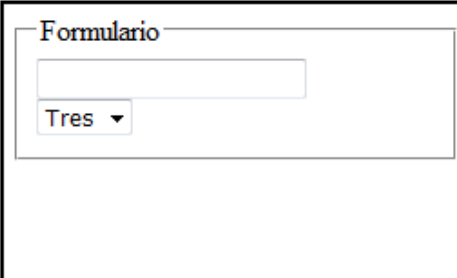
El control oculto se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor "hidden". Lógicamente, los navegadores no muestran estos controles en la pantalla (aunque pueden verse en el código fuente). Se utilizan para almacenar información que de otro modo se perdería (por ejemplo, cuando hay varios formularios encadenados) o cuando tenemos que enviar algún dato a otra página web que no controlamos nosotros.

<pre><input type="hidden" name="nombre" value="pepito" /></pre>	
---	--

Grupos de controles: `<fieldset>`

La etiqueta `<fieldset>` permite agrupar un conjunto de controles. Los navegadores muestran una caja alrededor de cada grupo de controles.

La etiqueta `<legend>` permite añadir una leyenda al `<fieldset>`. Los navegadores muestran la leyenda sobre el borde que rodea el grupo de controles.

<pre><fieldset><legend>Formulario</legend> <input type="text" name="texto" />
 <select name="menu"> <option>Uno</option> <option>Dos</option> <option selected>Tres</option> </select> </fieldset></pre>	
--	--

Accesibilidad: `<label>`

La etiqueta `<label>` permite asociar un control con un texto, de manera que mejore la accesibilidad de los formularios. La etiqueta `<label>` usa el atributo `for` para ser enlazada al atributo `id` de un control de formulario.

Por ejemplo, en el caso de una casilla de verificación, la etiqueta `<label>` permite que la casilla se marque o desmarque haciendo clic en el texto, como se muestra en el ejemplo siguiente:


```
<input type="checkbox" name="casilla" id="casilla1" />
<label for="casilla1">Casilla 1</label>
```

☒ Casilla 1

En el caso de las cajas de texto `<input type="text" />`, la etiqueta `<label>` permite que el cursor se sitúe en la caja de texto haciendo clic en el texto.

Otros controles

Existen otros tipos de controles que han sido incorporados recientemente al estándar HTML5 y que, por lo tanto, aún no están reconocidos por todos los navegadores más famosos. Algunos de ellos son los siguientes:

Selector de color: `<input type="color" />`

El control de selector de color se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `color`. Permite elegir un color mediante el selector de color proporcionado por el sistema operativo.

```
<input type="color" name="color" />
```



Número: `<input type="number" />`

El control de número se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor `number`.

En principio, los valores admitidos por el control son únicamente números enteros. Los navegadores alertan al usuario cuando no ha escrito un valor correcto, como mínimo al enviar el formulario.


```
Escriba un número:
<input type="number" name="numero" />
```

Escriba un número: 16

- El atributo `min` establece un valor mínimo en la respuesta.
- El atributo `max` establece un valor máximo en la respuesta.
- El atributo `step` establece la cantidad, entera o decimal, en la que aumenta o disminuye el valor cuando hacemos click en las flechas de subir y bajar.
- Para poder escribir cualquier número, el atributo `step` debe tener el valor "any".

Rango: `<input type="range" />`

El control de rango se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor "range". El rango de valores predeterminado es de 0 a 100 (valores enteros). Los atributos `min`, `max` y `step` permiten elegir respectivamente el valor mínimo, el máximo y el incremento.

<p>Indique su nivel (de 0 a 1, de 0,1 en 0,1):</p> <pre><input type="range" name="rango" min="0" max="1" step="0.1" /></pre>	<p>Indique su nivel:</p> 
--	--

Fecha: `<input type="date" />`

El control de fecha se crea mediante la etiqueta `<input />` cuyo atributo `type` tiene el valor "date". Los atributos `min` y `max` permiten elegir respectivamente el valor mínimo y máximo de la fecha que puede ser seleccionada:

<p>Selecciona una fecha anterior a 1980-01-01:</p> <pre><input type="date" name="bday" max="1979-12-31" /></pre>	<p>Selecciona una fecha anterior a 1980-01-01:</p> <p>14/12/1979 X ▾ ▴ ▾</p> <p>diciembre de 1979 ▾ ◀ ● ▶</p> <table border="1"> <thead> <tr> <th>lun</th> <th>mar</th> <th>mié</th> <th>jue</th> <th>vie</th> <th>sáb</th> <th>dom</th> </tr> </thead> <tbody> <tr> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> </tr> <tr> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> </tr> <tr> <td>17</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> <td>22</td> <td>23</td> </tr> <tr> <td>24</td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> </tr> <tr> <td>31</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> </tbody> </table>	lun	mar	mié	jue	vie	sáb	dom	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6
lun	mar	mié	jue	vie	sáb	dom																																												
26	27	28	29	30	1	2																																												
3	4	5	6	7	8	9																																												
10	11	12	13	14	15	16																																												
17	18	19	20	21	22	23																																												
24	25	26	27	28	29	30																																												
31	1	2	3	4	5	6																																												

Correo electrónico: `<input type="email" />`

El control de caja de texto para introducir una dirección de correo electrónico se construye con la etiqueta `<input />` cuyo atributo `type` tiene el valor "email". Si el texto introducido no es una dirección de email válida, el navegador no permitirá enviar el formulario:

<pre><form action="/enviar_email"> E-mail: <input type="email" name="email"> <input type="submit"> </form></pre>	<p>E-mail: <input type="text"/></p> <p>Enviar consulta</p> <p>Introduzca una dirección de correo.</p>
--	---

Nuevos atributos de controles en HTML5

El lenguaje HTML5 ha introducido nuevos atributos comunes para los controles de formulario, los cuales mejoran sustancialmente la usabilidad. Algunos de estos nuevos atributos son los siguientes:

autocomplete: indica si se permite al navegador almacenar el dato introducido por el usuario, de forma que pueda ofrecer este valor como sugerencia, en futuras visitas al formulario donde se encuentra ese control. Admite los valores "on" y "off". Este atributo se suele colocar en cajas de texto donde se recoge información sensible, como por ejemplo números de tarjetas de crédito.

autofocus: es un atributo booleano (no tiene valor, sólo nombre) que especifica a qué control se le debe dar el foco una vez que termine de cargarse la página web donde se encuentra. Sólo se debe colocar este atributo en un control de entre todos los presentes en todos los formularios que compongan una página.

placeholder: muestra un texto de ejemplo dentro de un control, como por ejemplo en un input de tipo text o en un textarea. Este texto se muestra sombreado y sirve sólo para dar una pista al usuario acerca de lo que debe introducir. Una vez que el control coja el foco, el placeholder desaparecerá. Es importante indicar que si el control tiene un `value` establecido, el placeholder no se mostrará, ya que en su lugar irá el contenido del atributo `value`.

E-mail:

required: indica si el control actual debe rellenarse o marcarse para que se permita enviar el formulario. En las cajas de texto, obliga a introducir un valor, mientras que en el resto de controles obliga a que el usuario seleccione una opción o un valor. En el caso de los grupos de radiobuttons, sólo es necesario colocar este atributo a uno de los radiobuttons que componen el grupo.

list: asocia una lista de elementos al control actual. Esta funcionalidad es muy útil, por ejemplo, en las cajas de texto, donde podemos usar esta funcionalidad para sugerir al usuario una serie de valores posibles que podría seleccionar. Estos valores no son más que una sugerencia y es el usuario, en última instancia, quien decidirá si selecciona alguno de ellos o si escribe aquello que él desee.

Las listas de valores se construyen con la etiqueta `<datalist>`:

<pre><form action="encuesta.php" method="post"> Navegador: <input list="listaNavegadores" name="navegador"> <datalist id="listaNavegadores"> <option>Internet Explorer</option> <option>Firefox</option> <option>Chrome</option> <option>Opera</option> <option>Safari</option> </datalist> <button type="submit">Enviar</button> </form></pre>	<p>Navegador:</p> <div style="display: flex; align-items: center;"> <input type="button" value="Enviar"/> <div style="border: 1px solid #ccc; padding: 5px; margin-left: 10px;"> Internet Explorer Firefox Chrome Opera Safari </div> </div>
--	--

Nota: los atributos comunes a todas las etiquetas HTML vistos en la Unidad Formativa 1, como por ejemplo `title`, son también de aplicación en los controles de formulario, ya que también son etiquetas HTML.

Bibliografía

Este documento es obra de Javier Gil Motos y ha sido construido, en parte, a partir de las siguientes fuentes de documentación:

- Referencia oficial del W3C - <http://w3.org/TR/html401/interact/forms.html>
- Portal MCLIBRE – Material Curricular Libre - <http://www.mclibre.org/>

El contenido de esta guía se encuentra cubierto por la licencia Creative Commons *Attribution-ShareAlike 4.0 International* (CC BY-SA 4.0):

https://creativecommons.org/licenses/by-sa/4.0/deed.es_ES