

MF0950_2-Contrucción de páginas web

UF1304: Elaboración de plantillas y formularios



Generalitat de Catalunya
Departament d'Empresa i Ocupació [1]



Unió Europea
Fons social europeu
L'FSE inverteix en el teu futur



MINISTERIO
DE TRABAJO
E INMIGRACIÓN

SERVICIO PÚBLICO
DE EMPLEO ESTATAL

Contenido

UF1304: Elaboración de plantillas y formularios.....	1
UF1304: Elaboración de plantillas y formularios.....	3
Descripción de los controles de los formularios.....	3
Atributos Globales para los elementos <input>	3
Name.....	3
Required.....	3
Placeholder.....	3
Min.....	4
Max.....	4
Maxlength.....	4
Pattern.....	4
ReadOnly.....	5
Controles de texto	5
<INPUT>.....	5
Radio Button (Botón de opción) type="radio"	7
El elemento <select>	7
El elemento <textarea>	7
Botones	8
Atributos etiqueta FORM	8
Atributo action.....	8
Atributo method.....	8

UF1304: Elaboración de plantillas y formularios

Descripción de los controles de los formularios

Un formulario web se compone de diferentes controles para la manipulación de datos de entrada. Estos controles se añaden en el HTML dentro de la etiqueta <FORM>, y se opta por uno u otro en función del tipo de dato que se maneje y la representación de este.

Dentro de los controles de un formulario se puede diferenciar:

- Controles de texto.
- Casillas de verificación.
- Radios o grupos de opciones.
- Etiquetas.
- Menús de selección.
- Campos ocultos.
- Botones.
- Subir archivos.

Atributos Globales para los elementos <input>

Todos los elementos de formulario disponen de los siguientes atributos globales.

Name

Cada campo de entrada debe tener un atributo de nombre “name” para ser enviado. Si se omite el atributo, los datos de ese campo de entrada no se enviarán en absoluto.

Required

El atributo required es un atributo booleano. Cuando está presente, especifica que se debe rellenar un campo de entrada antes de enviar el formulario. Nota: El atributo requerido funciona con los siguientes tipos de entrada: texto, búsqueda, url, tel, correo electrónico, contraseña, selectores de fecha, número, casilla de verificación, radio y archivo.

Placeholder

El atributo placeholder especifica una sugerencia que describe el valor esperado de un campo de entrada (un valor de muestra o una breve descripción del formato). La indicación se muestra en el campo

de entrada antes de que el usuario introduzca un valor. El atributo de marcador de posición funciona con los siguientes tipos de entrada: texto, búsqueda, url, tel, correo electrónico y contraseña.

Min

Define un valor mínimo numérico. También se puede definir una fecha para los campos de tipo fecha.

Max

Define un valor máximo numérico. También se puede definir una fecha para los campos de tipo fecha.

Maxlength

Limita el número máximo de caracteres.

Campo input con un máximo de 10 caracteres

```
<input type="text" name="username" maxlength="10">
```

Pattern

Muy útil para la validación de datos. A través de este atributo se puede configurar el tipo de dato que debe ser introducido, como solo letras, solo números, un dni, cuenta bancaria, número de teléfono, referencia de un artículo/pedido con determinados números y letras, etc.

Solo números

```
pattern="[0-9]"
```

Sólo letras mayúsculas o minúsculas

```
pattern="[A-Za-z]"
```

Sólo letras mayúsculas

```
pattern="[A-Z]"
```

3 letras mayúsculas o minúsculas

```
pattern="[A-Za-z]{3}"
```

Código postal

```
pattern="[0-9]{5}"
```

NIF con letra delante, 7 números y letra detrás o con 8 números y 1 letra. Todas las letras pueden ser mayúsculas o minúsculas

```
pattern="((([X-Za-z]{1})([0-9]{7})([A-Za-z]{1}))|((([0-9]{8})([A-Za-z]{1})))")
```

Que contenga 6 o más caracteres, útil para passwords

```
pattern=".{6,}"
```

ReadOnly

Añadiendo este atributo el usuario no podrá modificar el campo por sí ver si contenido, útil para cuando el dato se ha generado de forma dinámica y solo se muestra como información.

Controles de texto

Permiten introducir texto en el formulario. Es posible introducir un valor concreto o un texto simple. Los elementos HTML usados son los siguientes:

<INPUT>

Es el control más utilizado en los formularios. Se utiliza para introducir texto, introducir contraseñas, verificar datos, seleccionar entre varias opciones, seleccionar archivos para enviar e incluir botones para enviar el formulario, borrar los datos o ejecutar código JavaScript. Se define de la siguiente forma:

```
<form>
    Nombre:<br>
    <input type="text" name="nombre"><br>
    Apellido:<br>
    <input type="text" name="apellido">
</form>
```

Datalist

Facilitan el autocompletado de un campo tipo input. Los usuarios verán una lista desplegable de las opciones predefinidas a medida que introducen datos. El atributo de lista del elemento <input>, debe referirse al atributo id del elemento <datalist>.

```
<input list="browsers" name="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Variables del atributo "type" para campos de texto/números

type="text"

El más utilizado, permite introducir cualquier tipo de carácter.

type="number"

Define un campo de entrada numérico. También puede establecer restricciones sobre qué números se aceptan.

Ejemplo para números entre 1 y 5

```
<input type="number" name="cantidad" min="1" max="5">
```

type="password"

Oculta los caracteres introducidos.

type="submit"

Válido para un <input> como para un <button>: Realiza la acción de enviar la información del formulario.

type="reset"

Válido para un <input> como para un <button>: Realiza la acción de borrar la información del formulario.

type="hidden"

Campo de tipo oculto, muy útil cuando se ha de enviar información en el formulario que no depende del usuario, como la fecha, el resultado de una operación matemática previa, información del navegador, resolución de pantalla, ip pública, elemento de control para verificar el origen de los datos, etc.

type="color"

Muestra un selector de color.

type="date"

Muestra un selector de fechas. Se pueden usar los atributos min y max para establecer fechas anteriores u posteriores.

type="mail"

Se utiliza para los campos de entrada que deben contener una dirección de correo electrónico. Dependiendo del soporte del navegador, la dirección de correo electrónico se puede validar automáticamente cuando se envía. Algunos smartphones reconocen el tipo de correo electrónico y añaden ".com" al teclado para que coincida con la entrada de correo electrónico.

`type="month"`

Permite al usuario seleccionar un mes y un año.

Radio Button (Botón de opción) `type="radio"`

Es el elemento de formulario que da a escoger al usuario 1 de entre varias opciones. Las opciones están agrupadas a través del atributo `name` con el mismo valor, cambiando el atributo `value`

Ejemplo.

```
<form action="">
  <input type="radio" name="gender" value="masculino"> Masculino<br>
  <input type="radio" name="gender" value="femenino">Femenino<br>
  <input type="radio" name="gender" value="otros">Otros
</form>
```

El elemento `<select>`

Muestra una lista desplegable para escoger opción.

Los elementos `<option>` definen una opción que se puede seleccionar. De forma predeterminada, se selecciona el primer elemento de la lista desplegable. Para definir una opción preseleccionada, agregue el atributo `selected` a la opción. Para deshabilitar una opción agrega el atributo `disabled`.

```
<select name="coches">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

El elemento `<textarea>`

Permite la introducción de texto multilínea. Con los atributos `col` y `rows` se pueden especificar el número de columnas (ancho) y de filas (alto) como área de introducción de texto.

```
<textarea name="message" rows="10" cols="30">
</textarea>
```

Casilla de verificación (checkbox)

Permite al usuario activar o desactivar casilla en forma de cuadrado. Se le puede añadir el atributo “checked” para que por defecto esté activada.

```
<form>
  <input type="checkbox" name="vehicle1" value="bici">Tengo      bici      <br>
  <input type="checkbox" name="vehicle2" value="coche"> Tengo      coche
</form>
```

Botones

Existen principalmente 2 formas de introducir un elemento de formulario que permita al usuario enviar el formulario:

- Etiqueta input con type=”submit”
- Etiqueta button con type=”submit”

También existe la posibilidad de que un botón borre toda la información de un formulario añadiendo el valor “reset” al atributo “type”.

Atributos etiqueta FORM

Como ya se comenta anteriormente, todos los elementos de un formulario deben estar agrupados dentro de una etiqueta form la cual sirve para que el navegador obtenga los datos del formulario y se envíen a la página que tratará la información.

Atributo action

Se especifica a que Url debe enviar el navegador los datos del formulario para su tratamiento. Si no se especifica, por defecto, se envían a la misma página.

Atributo method

Se especifica cómo se envían los datos de formulario, GET o POST

Valor get (method=”GET”): Envía los datos del formulario a través de variables get añadidas a la ruta establecida en el action. Un ejemplo es el buscador Google, si creamos una búsqueda por la palabra jamón (el buscador es un campo de formulario), podemos observar en la url de la página se añade la variable “q” con el valor buscado: <https://www.google.es/search?q=jamon>

Valor post (method="POST"): Envía los datos del formulario a la url indicada en el action a través del método POST, un método interno donde las variables no se muestran fácilmente. Generalmente este es el método para cuando se quiere almacenar en base de datos la información introducida. Las ventajas respecto al GET son:

- Las solicitudes POST nunca se almacenan en caché.
- Las solicitudes POST no permanecen en el historial del navegador.
- Las solicitudes de POST no se pueden almacenar.
- Las peticiones POST no tienen restricciones en la longitud de los datos.

Todos estos parámetros forman parte del diseño de formularios, el tratamiento de los datos (guardar información en base de datos, ejecutar consultas, buscadores, etc.) se programan en lenguajes del lado del servidor como Php o Asp y/o desde lenguaje JavaScript o variantes como JQuery o TypeScript.