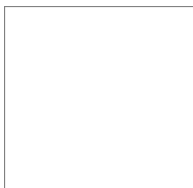


HTML 5 : Formularios



El formulario es una interfaz fundamental en las páginas web para interactuar con el usuario. Esta interacción se genera a través de variados elementos, como cajas para introducir texto, botones sencillos o del tipo *check* o *radio*, controles de rangos, cajas para selección de colores y otros.

Una de las grandes novedades de HTML5 es la creación de nuevas etiquetas que nos asisten en la creación de estos formularios.

HTML5 también incluye validaciones nativas para el navegador sin necesidad de usar javascript.

Ejemplo:

Formulario sencillo

Escribe tu nombre:

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Ejemplo Formulario 1</title>
  <meta charset="UTF-8">
</head>
<body>
  <h1>Formulario sencillo</h1>
  <form action="recibeDatos.php" method="post">
    Escribe tu nombre: <input type="text" id="campo_nombre" size="40" maxlength="40" /><br/>
    <input type="submit" value="Enviar" />
  </form>
</body>
</html>
```

Etiqueta <form>

- Encierra todos los contenidos del formulario (botones, cuadros de texto, listas desplegables, etc.)
- Los formularios no pueden anidarse unos a otros como sucede con las listas y las tablas.

Sintaxis y atributos de <form>

```
<form id="nombreFormulario" action="URL_recurso" method="post" enctype="tipo_MIME" >
  etiquetas-elementos-de-formularios
</form>
```

id. Representa el identificador o nombre del formulario para hacer referencia a éste desde CSS o scripts.

action. Indica el URL del recurso donde se procesarán los datos del formulario cuando el usuario lo envíe.

```
<form id="f1" action="recibeDatos.php" >
<form id="f2" action="recibeDatos.jsp" >
<form id="f3" action="recibeDatos.asp" >
```

Ejemplos:

method. Especifica el método HTTP que se empleará para enviar los datos del usuario que ha introducido en el formulario. Los valores posibles son: **get** (por defecto) y **post**.

- **method="get"** : Los datos del formulario se concatenan al URI o la URL

especificada por el atributo *action* con un signo de interrogación (?) como separador y esta nueva URL se envía al navegador. Los datos forman el *QueryString* (Cadena de consulta).

Limitaciones de "get":

- Como máximo admite 500 bytes de información
- No permite el envío de archivos adjuntos al formulario.
- Los datos al enviarse a través del URL se ven fácilmente.
- **method="post"** : Los datos del formulario se incluyen en el cuerpo del mensaje y se envían al navegador.

Como regla general, se dice que el método GET se debe utilizar en los formularios que no modifican la información (ejemplo en un formulario de búsqueda). Observa como en las consultas de los buscadores usan el método GET.

Mientras que cuando se modifica la información original (inserción, modificación o borrado) se debe usar el método POST.

Envío de datos

Cuando se envía un formulario (haciendo clic en el botón de envío), los datos del formulario se envían a un script CGI bajo la forma de *pares nombre/valor*, es decir conjuntos de datos representados por el nombre del elemento formulario, un signo igual ("=") y luego el valor asociado. Estos pares nombre/valor se separan unos de otros mediante el símbolo de unión("&"). Por lo tanto, los datos que se envían al script se verán así:

```
campo1=valor1&campo2;=valor2&campo3;=valor3
```

Con el método GET (enviar los datos mediante una dirección [URL](#)), la URL será una cadena como la siguiente:

```
http://es.kioskea.net/cgi-bin/script.cgi?campo1=valor1&campo2;=valor2
```

enctype. Indica el tipo [MIME](#) (Serie de convenciones o especificaciones dirigidas al intercambio a través de internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario) que se empleará para enviar los datos del formulario al servidor. Si se desea enviar los datos sin ninguna codificación debe emplearse el valor "text/plain".

Campos/controles de formulario que pueden incluir <form>

```
<form ....>
    <input>
    <textarea>
    <button>
    <select>
    <option>
    <optgroup>
    <fieldset>
    <label>
    <datalist> new HTML 5
    <keygen> new HTML 5
    <output> new HTML 5
</form>
```

La etiqueta <input ... >

La mayoría de los controles que presentan los formularios se crean con la etiqueta <input type=.....>, por lo su definición formal y su lista de atributos es muy extensa:














Atributos de la etiqueta input

El principal atributo de la etiqueta **input** es **type** ya que es el que le indica el tipo de datos que debe recibir.

- **type**: indica el tipo de control de datos que se ha elegido. Valores que puede incluir: text, password, button, checkbox, reset, radio, hidden, file, image, submit
nuevas en html5: color, date, datettime, datettime-local, email, month, number, range, search, tel, time, url, week.
- **src**: Para el control que permite crear botones con imagenes, indica el URL del archivo.
- **step**: (html5) permite establecer la cantidad de valores posibles dentro de un rango.
- **value**: indica el valor inicial del control.
- **autofocus**: (html5) fuerza el foco (la posición del cursor) dentro de un campo, una vez que se carga la página. en javascript conseguimos lo mismo con la función **focus()**.
- **required**: (html5) atributo que se aplica a un campo de entrada y obliga al usuario a escribir un valor en dicho campo. anteriormente se conseguía esto mediante javascript analizando la longitud del valor del campo en cuestión mediante *campo.value.length*.
- **placeholder**: (html5) muestra un texto por defecto en el input hasta que éste tiene el foco, en ese momento el texto desaparece para que el usuario introduzca el texto que desea. anteriormente este efecto se creaba con javascript.
- **autocomplete**: (html5) activa (**on**) o desactiva (**off**) el auto completado de la entrada de datos en base a los textos introducidos anteriormente por el usuario. puede usarse como atributo de un campo de entrada o del formulario completo.
- **alt**: se utiliza cuando el tipo es una imagen y no se puede cargar en la interfaz del usuario, se muestra este texto alternativo.
- **checked**: para indicar si el elemento está seleccionado por defecto en los controles checkbox y radiobutton.
- **disabled**: es un valor booleano que indica que el elemento está desactivado, con lo cual no admite entrada de datos. Y su valor no se envía al Servidor junto al resto de datos.
- **readonly**: valor booleano que indica que el valor del campo no se puede modificar, solo lectura.
- **formenctype**: (html5) indica el modo de envío de los datos.
application/x-www-form-urlencoded; multipart/form-data; text/plain
- **formaction**: (html5) es la dirección url que se utiliza al activar el elemento.
- **formnovalidate**: (html5) es un valor booleano que indica que el formulario no se valida antes de enviarse.
- **formtarget**: (html5) ventana del destino del formulario. *_blank, _self, _parent, _top, framename*
- **height**: (html5) altura del elemento tipo image.
- **width**: (html5) anchura del elemento tipo image.
- **size**: Tamaño inicial del control. En text/password indica la cantidad de caracteres que se pueden introducir en el campo, para el resto significa su tamaño en pixels.
- **max**: (html5) valor máximo que se puede entrar en el elemento de datos.
- **min**: (html5) valor mínimo que se puede entrar en el elemento de datos.
- **maxlength**: longitud máxima de caracteres para los campos text/password.
- **list**: (html5) es un identificador de una lista que se define con datalist

- **multiple:** (html5) valor booleano que indica que se permite la selección de valores múltiples.
- **name:** Nombre que identifica al campo de datos. Importante para su identificación por el Servidor.
- **pattern:** (html5) Expresión regular que se utiliza para validar entradas de datos. este atributo permite hacer validaciones muy complejas y utilizado correctamente puede ahorrarse muchas líneas de código.

Atributo `<input type="valor">` Podemos incluir los siguientes valores, veamos su significado:

Value	Descripción
button	Define un botón que se pulse (en su mayoría se utiliza con un JavaScript para activar una secuencia de comandos)
checkbox	Define una casilla de verificación
color 	Define un selector de color
date 	Define un control de fecha (año, mes y día (no hay tiempo))
datetime 	Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracción de segundo, basado en la zona horaria UTC)
datetime-local 	Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracciones de segundo (sin zona horaria))
email 	Define un campo para una dirección de e-mail
file	Define un campo de archivo de selección y un "Examinar ..." botón (para la carga de archivos)
hidden	Define un campo de entrada oculto
image	Define una imagen como botón submit
month 	Define un mes y control del año (sin zona horaria)
number 	Define un campo para introducir un número
password	Define un campo de contraseña (caracteres se enmascaran)
radio	Define un botón de opción
range 	Define un control para introducir un número cuyo valor exacto no es importante (como un control deslizante)
reset	Define un botón de reinicio (restablece todos los valores del formulario a valores por defecto)
search 	Define un campo de texto para introducir una cadena de búsqueda
submit	Define un botón de envío
tel 	Define un campo para introducir un número de teléfono
text	Default. Define un campo de texto de una sola línea (ancho predeterminado es de 20 caracteres)
time 	Define un control para entrar en un tiempo (sin zona horaria)
url 	Define un campo para introducir una URL
week 	Define a la semana y el control al año (no zona horaria)

Veamos ejemplos de uso de los tipos de la etiqueta input:

Cajas de texto: `<input type="text" ..>`

Visualiza una caja de texto del tamaño especificado por el atributo `size`. Se emplea para pedir datos de entrada relativamente cortos de una línea como por ejemplo; nombre, apellidos, dirección, etc.

Atributos

name: Asigna un identificador a la variable donde se guardara el valor del usuario también sirve como referencia en scripts aunque más recomendado usar `id`.

value: Visualiza un valor predeterminado en el campo de entrada.

size: Indica la longitud visible en número de caracteres que tendrá la caja de texto.

maxlength: Indica la longitud máxima de caracteres que puede introducir el usuario.

Ejemplo:

```
Nombre <INPUT name="e1" type="text" value="Introduce tu nombre" size="30" maxlength="40" />
```

Nombre

Cajas de texto tipo contraseña: `<input type="password" ..>`

Es muy similar al tipo `text`, con la diferencia de que el texto introducido en la caja aparece oculto, mostrando un símbolo * (asterisco) por cada carácter en el navegador. Como su nombre indica, sirve para introducir contraseñas y que nadie de nuestro alrededor pueda ver lo que se escribe.

Ejemplo:

```
Introduce Contraseña <INPUT name="pwd" type="password" />
```

Introduce Contraseña

Casillas de verificación: `<input type="checkbox" ..>`

Ejemplo:

Deportes practicados:

```
<input type="checkbox" value="futbol" id="futbol" name="deportes" checked>
<label for="futbol">Futbol</label>
<input type="checkbox" value="baloncesto" id="baloncesto" name="deportes">
<label for="baloncesto">Baloncesto</label>
<input type="checkbox" value="ciclismo" id="ciclismo" name="deportes">
<label for="ciclismo">Ciclismo</label>
```

Deportes practicados: Futbol Baloncesto Ciclismo

Botón de formulario: `<input type="button" ..>`

Es un botón pulsador de uso genérico al que se pueden asociar distintos eventos que son manejados desde los scripts. Un uso muy frecuente consiste en asociar a la pulsación del botón la llamada a una función, por ejemplo, en **Javascript**, que compruebe la coherencia y la validez de los datos introducidos en el formulario para versiones anteriores a HTML5.

Ejemplo:

```
<input type="button" name="botonAceptar" value="Aceptar" onclick="validaDatos()" />
```

Aceptar

Botón de reinicio de formulario: `<input type="reset" ..>`

Reinicia el formulario, es decir, elimina el contenido de todos los valores introducidos o seleccionados, volviendo a los valores por defecto que tenía el formulario al cargarse.

Ejemplo:

```
<input type="reset" name="botonReinicio" value="Reiniciar" />
```

Reiniciar

Botón de envío del formulario: `<input type="submit" ..>`

Provoca el envío del formulario para su procesamiento al recurso especificado en el atributo **action** de la etiqueta `<form>`.

Ejemplo:

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

Enviar

También puede invocarse su acción mediante la función de Javascript `submit()` a través del modelo DOM: Ejemplo: `document.forms[0].submit();`

Envío de ficheros: `<input type="file" ..>`

Permite el envío de ficheros adjuntos en los formularios, permitiendo al usuario seleccionar el archivo que será enviado junto al resto del formularios. Su apariencia consiste en una caja de texto tipo "text" junto a un botón examinar que al pulsarlo muestra la estructura de archivos locales para su selección. Para poder utilizar este tipo input, es preciso poner en la etiqueta `<form>` el atributo `enctype` con el valor "multipart/form-data"

Ejemplo:

```
<form name="f1" method="post" action="recibeDatos.php" enctype="multipart/form-data">
```

```
  Adjuntar archivo <input type="file" name="fichero" value="Examinar" />
```

Adjuntar archivo

Adjuntar archivo pruebas-foms.pdf

Boton de envío con imágenes: `<input type="image" ..>`

Crea un boton tipo submit, personalizandolo con una imagen.

Se usa el atributo src para especificar el URI de la imagen que decorará el botón.

También puede utilizarse el atributo alt, para especificar un texto alternativo a la imagen.

```
<input type="image" src="imagenes/boton2.gif" name="botonImagen" />
```

Envía tu formulario

Información no visible: `<input type="hidden" ..>`

En las aplicaciones, a veces es preciso enviar información en los formularios sin que el usuario tenga que introducir dicha información.

Por ejemplo, se puede desear incluir un campo con un número de sesión, pero que éste no sea visible por el usuario.

```
<input type="hidden" name="sesion" value="1" />
```

Control radiobutton: `<input type="radio" ..>`

Son similares a los controles de tipo checkbox, pero presentan una diferencia muy importante son pueden ser mutuamente excluyentes. Para ello las distintas opciones deben tener el mismo nombre del campo *name*.

De forma que se utilizan cuando el usuario solamente puede escoger una opción entre las distintas opciones relacionadas que se le presentan.

Ejemplo:

Sexo `
`

```
<input type="radio" name="sexo1" value="Hombre" checked /> Hombre<br/>
```

```
<input type="radio" name="sexo" value="Mujer" /> Mujer
```

Sexo

Hombre

Mujer


Nuevos atributos HTML-5

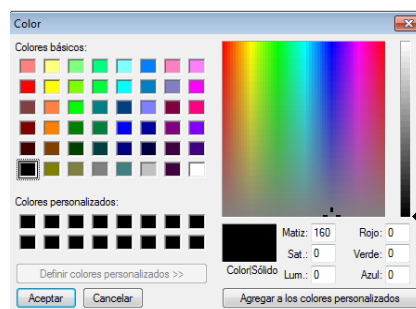
Color: `<input type="color" ..>`

Proporciona al usuario un selector de color, devuelve un valor RGB hexadecimal, como #FFF

Ejemplo:

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

Selecciona un color 



Selección de una Fecha con Date: `<input type="date" ..>`

Permite al usuario seleccionar una fecha (año, mes, día) aunque no la hora.
Ejemplo:

```
Fecha de Inicio<input type="date" name="fechaInc" />
```

Fecha de Inicio 16/11/2012

noviembre 2012						
lun	mar	mié	jue	vie	sáb	dom
29	30	31	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	1	2
3	4	5	6	7	8	9

Hoy Eliminar

Selección de una Fecha y hora con datetime: `<input type="datetime" ..>`

Permite al usuario seleccionar una fecha (año, mes, día) y la hora incluye zona horaria.
De momento solo disponible en Opera (versión 12.10)

Ejemplo:

```
Fecha y hora<input type="datetime" name="fechaHora" />
```

Fecha y hora 2012-11-18 23:57 UTC

Noviembre 2012						
Lun	Mar	Mie	Jue	Vie	Sab	Dom
29	30	31	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	1	2
3	4	5	6	7	8	9

Hoy

Selección de una Fecha y hora con datetime-local: `<input type="datetime-local" ..>`

Permite al usuario seleccionar una fecha (año, mes, día) y la hora NO incluye zona horaria.
De momento solo disponible en Opera (versión 12.10)

```
Fecha y hora local sin zona horaria<input type="datetime-local" name="fechaHora" />
```

Fecha y hora local sin zona horaria 2012-11-18 00:01

Noviembre 2012						
Lun	Mar	Mie	Jue	Vie	Sab	Dom
29	30	31	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	1	2
3	4	5	6	7	8	9

Hoy

Selección de un número: `<input type="number" ..>`

Proporciona una entrada de dato para introducir un número. Normalmente se trata de un cuadro de texto donde se puede escribir un número o hacer clic en las flechas arriba o abajo para seleccionarlo.

Tiene los atributos **max** y **min** para especificar los valores máximo y mínimo permitidos, también se puede utilizar el atributo **step**, este determina la cantidad a disminuir o aumentar cuando hace clic en las flechas.

Ejemplo:

```
Número<input type="number" name="num" /> Número 3
```

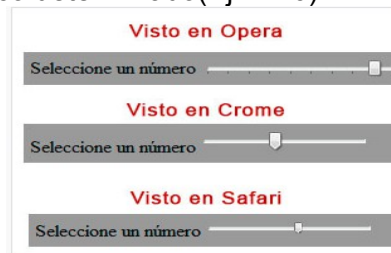
```
Número par del 0 al 10<input type="number" max="10" min="0" step="2" name="par" />
```

Número par del 0 al 10 2

Selección de un rango de números: `<input type="range" ..>`

Proporciona un deslizador en aquellos navegadores que lo soportan, al igual que el tipo number es soportado por Opera y los navegadores que trabajan con [Webkit](#) . Al igual que number permite los atributos max, min y step.

La diferencia entre number y range, es que con range no es relevante obtener el valor exacto de un número, sino un rango numérico determinado(Ej. 1-10).



Selección de una búsqueda: `<input type="search" ..>`

Proporciona un campo de búsqueda, es muy parecido a un campo de texto normal. Algunos navegadores agregan el icono X para borrar el contenido de la búsqueda.

```
Buscar<input type="search" name="busqueda" placeholder="buscar .." />  
<input type="submit" value="IR" />
```



Datos de identificación

Una interesante novedad en HTML 5 es la ampliación de la etiqueta input para simplificar la introducción y la validación de variados elementos de identificación y localización.

Los nuevos tipos relacionados con esta característica son:

Control para el email: `<input type="email" ..>`

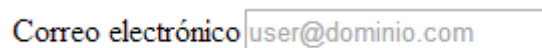
Se utiliza para especificar una o más direcciones de correo electrónico.

Soporta el **atributo booleano multiple**, para permitir varias direcciones separadas por coma.

Sigue pareciendo un campo de texto normal, sin embargo, si se intenta enviar el formulario y los datos introducidos no son formato de email, algunos navegadores ofrecen un mensaje de error.

Mensajes de error mostrados por algunos navegadores.

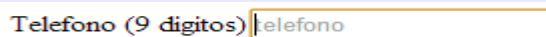
```
Correo electr&ocute;nico<input type="email" placeholder="user@dominio.com" name="email" />
```



Selección de un teléfono: `<input type="tel" ..>`

Para pedir un número de teléfono al usuario usaremos el tipo de entrada "tel". A diferencia de los tipos url y email, el tipo tel no impone su sintaxis o patrón determinado. Tanto las letras como los números son válidos. De todas formas, podemos crear nosotros mismos el formato que nos interese utilizando Javascript.

```
Telefono (9 digitos)<input type="tel" pattern="|^\\d{9}$|" placeholder="telefono" name="tlf" /></td>
```



Control para el url: `<input type="url" ..>`

Para pedir al usuario una entrada de una dirección web. Se muestra como un simple campo de texto normal. En muchas pantallas táctiles, el teclado se optimizará para la entrada de direcciones web, con una barra(/) y una tecla de método abreviado (.com).



HTML 5 no valida que realmente sea una url válida, si quieres proporcionar a tu formulario un patrón de url utiliza el atributo **pattern="regexp"** (expresión regular) <http://html5pattern.com/>

La etiqueta `<textarea > </textarea>`

Genera un campo de texto multilínea en el que introducir textos largos, como descripciones, etc.

Consiste en una caja de filas y columnas dentro de la cual se introduce el texto.

No dispone de límite en el número de caracteres, por lo que cuando el usuario introduce más texto del visualizado aparecen las barras de desplazamiento de la caja (scroll).

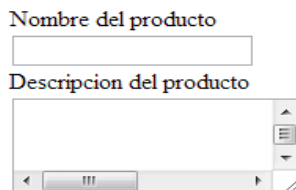
Se puede incluir un contenido inicial. Atributos:

cols: Especifica la anchura o columnas visibles en número de caracteres.

rows: Especifica la altura o filas de texto visibles

Ejemplo:

```
<form name="f1" action="insertar_producto.php" method="post">
  <label for="nombre">Nombre del producto</label> <br/><input type="text" name="nombre" /><br/>
  <label for="descripcion">Descripcion del producto</label><br/> <textarea name="descripcion" cols="20" rows="4"></textarea>
</form>
```



The screenshot shows a web form with two input fields. The first is a single-line text input labeled 'Nombre del producto'. The second is a multi-line text area labeled 'Descripcion del producto', which is currently empty and has visible scrollbars on the right and bottom.

La etiqueta `<select > </select>`

Genera un menú de opciones, con la posibilidad de visualizarse como una lista con deslizamiento o un menú desplegable.

Dispone de etiqueta de inicio y fin igual que *textarea*.

Atributos:

size: Si el elemento `<select>` se presenta como una lista con desplazamiento (“scrolled list box”), este atributo especifica el número de opciones/elementos de la lista que deberían ser visibles al mismo tiempo.

multiple: Atributo booleano que permite al usuario la posibilidad de seleccionar más de 1 opción de la lista, mediante la tecla control. Si no se especifica solo podrá seleccionar una.

La etiqueta `<select>` creó el menu junto a la combinación de la etiqueta `<option>` y, opcionalmente la etiqueta `<optgroup>`. Ambos con etiquetas también de cierre.

- Todo elemento `<select>` debe contener al menos un elemento `<option>`
- Con `<optgroup>` permitimos agrupar opciones.

Atributos de la etiqueta `<option>`

selected: Indica que ésta opción es la preseleccionada inicialmente.

value: Especifica el valor inicial de la opción a visualizarse. Si no se especifica el valor inicial es el texto incluido entre las etiquetas: `<option> texto </option>`. Obligatorio, ya que es el dato que se envía al Servidor cuando el usuario envía el formulario.

label: Permite especificar un rótulo más corto para la opción.

Ejemplo1:

```
<label for="so">Sistema Operativo </label> <br/>
<select id="so" name="so">
  <option value="" selected> -selecciona- </option>
  <option value="Linux">Linux Ubuntu</option>
  <option value="mac">OS MAC X</option>
  <option value="windows">Windows 7</option>
  <option value="solaris">Solaris</option>
  <option value="otro">Otro</option>
</select>
```

Sistema Operativo

Ejemplo2:

```
<label for="so">Sistema Operativo </label> <br/>
<select id="so" name="so" size="5">
  <option value="Linux">Linux Ubuntu</option>
  <option value="mac" selected >OS MAC X</option>
  <option value="windows">Windows 7</option>
  <option value="solaris">Solaris</option>
  <option value="otro">Otro</option>
</select>
```

Sistema Operativo

Linux Ubuntu	▲
OS MAC X	
Windows 7	
Solaris	
Otro	▼

La etiqueta <label> </label>

<label> se asocia a un único control del formulario y permite asociar una etiqueta a un control.

Mediante el atributo for se indica el nombre del control asociado a la etiqueta.

Para la asociación, el control debe incluir el atributo **id** (o bien el atributo **name**) con el valor al que se asocia la etiqueta.

La etiqueta <fieldset> </fieldset>

Mediante este elemento, se puede agrupar en una caja varios controles y etiquetas del formulario que estén relacionados. Además, añadiendo el elemento <legend> se asigna un título a esta caja.

Ejemplo de uso de <fieldset>, <legend> y formularios en tablas:

```
<!DOCTYPE HTML>
<html>
<head> <title>Ejemplo Formulario </title>
  <meta charset="UTF-8">
</head>
<body>
  <form id="f2" name="f2" action="fichero.php" method="post">
    <table>
      <tr><td>
        <fieldset>
          <legend> Mi equipo de Futbol</legend>
          <table>
            <tr><td> <input type="radio" name="equipo" />Real Madrid</td></tr>
            <tr><td> <input type="radio" name="equipo" />FC Barcelona</td></tr>
            <tr><td> <input type="radio" name="equipo" />Atlético de Madrid</td></tr>
          </table>
        </fieldset>
      </td>
      <td>
        <fieldset>
          <legend> Ganador de Formula 1</legend>
          <table>
            <tr><td> <input type="radio" name="gaf1" />Alonso</td></tr>
            <tr><td> <input type="radio" name="gaf1" />Vettel</td></tr>
            <tr><td> <input type="radio" name="gaf1" />Raikkonen</td></tr>
          </table>
        </fieldset>
      </td>
    </tr>
    <tr><td>
      <fieldset>
        <legend> Capital de Suecia</legend>
        <table>
          <tr><td> <input type="radio" name="capital" />Helsinki</td></tr>
          <tr><td> <input type="radio" name="capital" />Oslo</td></tr>
          <tr><td> <input type="radio" name="capital" />Estocolmo</td></tr>
        </table>
      </fieldset>
    </td>
    <td>
      <fieldset>
        <legend> Calificar</legend>
        <table>
          <tr><td> <input type="radio" name="calificar" />Bueno</td></tr>
          <tr><td> <input type="radio" name="calificar" />Regular</td></tr>
          <tr><td> <input type="radio" name="calificar" />Malo</td></tr>
        </table>
      </fieldset>
    </td>
  </tr>
</table>
</form>
</body>
</html>
```

<p>Mi equipo de Futbol</p> <ul style="list-style-type: none"><input type="radio"/> Real Madrid<input type="radio"/> FC Barcelona<input type="radio"/> Atlético de Madrid	<p>Ganador de Formula 1</p> <ul style="list-style-type: none"><input type="radio"/> Alonso<input type="radio"/> Vettel<input type="radio"/> Raikkonen
<p>Capital de Suecia</p> <ul style="list-style-type: none"><input type="radio"/> Helsinki<input type="radio"/> Oslo<input type="radio"/> Estocolmo	<p>Calificar</p> <ul style="list-style-type: none"><input type="radio"/> Bueno<input type="radio"/> Regular<input type="radio"/> Malo

La etiqueta <optgroup> </optgroup>

Las listas desplegables permiten agrupar sus opciones de forma que el usuario pueda encontrar fácilmente las opciones cuando la lista es muy larga, gracias a <optgroup>.

<optgroup> permite definir un nivel de agrupamiento de opciones <option> dentro de una lista desplegable <select>. El atributo <label> especifica el rótulo del grupo de opciones.

Ejemplo:

```
<form id="f2" name="f2" action="fichero.php" method="post">
  <label for="programa">Programa seleccionado</label> <br/>
  <select id="programa" name="programa">
    <optgroup label="Sistemas Operativos">
      <option value="Linux">Linux Ubuntu</option>
      <option value="mac" selected >OS MAC X</option>
      <option value="windows">Windows 7</option>
      <option value="solaris">Solaris</option>
      <option value="otro">Otro</option>
    </optgroup>
    <optgroup label="Navegadores">
      <option value="Firefox">Firefox</option>
      <option value="Chrome" selected >Chrome</option>
      <option value="Opera">Opera</option>
      <option value="Safari">Safari</option>
      <option value="Internet Explorer">Internet Explorer</option>
      <option value="Otro">Otro</option>
    </optgroup>
  </select>
</form>
```

Programa seleccionado

Chrome
Sistemas Operativos
Linux Ubuntu
OS MAC X
Windows 7
Solaris
Otro
Navegadores
Firefox
Chrome
Opera
Safari
Internet Explorer
Otro

La etiqueta <button> </button>

La etiqueta <button> inserta un control de botón en un formulario o documento. A diferencia del elemento <input type="button" ..>, este botón puede contener a otros elementos, es decir, código HTML. Veamos un ejemplo:

```
<button type="button" >
  <p><strong> El contenido HTML </strong> está permitido en este botón</p>
  <p>Para saber más sobre este tag revisa <a href="http://www.w3schools.com/tags/tag_button.asp">elemento <button></a></p>
</button>
```

El contenido HTML está permitido en este botón

Para saber más sobre este tag revisa [elemento <button>](http://www.w3schools.com/tags/tag_button.asp)

Nuevas etiquetas de formulario en HTML 5

La etiqueta `<datalist>` `</datalist>`

Permite automatizar la introducción de datos.

Genera una lista desplegable de opciones para que el usuario realice su selección.

La ventaja de este nuevo elemento es que reconoce los primeros caracteres introducidos por el usuario, para "autocompletar" la información.

Veamos un ejemplo:

```
<label for="pais"> Selecciona tu pais:</label>
<input type="text" name="pais" id="pais" list="países" />
<datalist id="países">
  <option value="Españtilde;a"> </option>
  <option value="Italia"></option>
  <option value="Portugal"></option>
  <option value="Grecia"></option>
</datalist>
```

Selecciona tu pais:

- España
- Italia
- Portugal
- Grecia

La etiqueta `<keygen>` `</keygen>`

Esta etiqueta genera 2 claves, una pública y una privada.

Cuando se envía el formulario, la clave privada se almacena localmente y sólo la clave pública es enviada al Servidor.

Es un elemento de seguridad útil para instituciones que deben validar el acceso de usuarios a información privada desde el exterior de la red.

Los atributos de esta etiqueta son:

challenge: genera una serie de caracteres de validación que se envían junto con la clave.

keytype: especifica el tipo de algoritmo de seguridad utilizado por la clave. Por defecto, usa RSA.

name: nombre asociado a keygen.

autofocus: asigna el focus al cargar la página.

form: nombre del formulario al cual pertenece esta etiqueta

disabled: atributo booleano, si su valor es true, indica que la etiqueta esta deshabilitada

Ejemplo:

```
<form action="demo_keygen.php" method="get">
  Nombre: <input type="text" name="nombre" /><br/>
  Encriptación <keygen name="security"></keygen><br/>
  <input type="submit" />
</form>
```

Nombre:

Encriptación

La etiqueta `<output>` `</output>`

Representa el resultado de un cálculo a través de un script.

Ejemplo:

```
<h4> Etiqueta <output> </h4>
<form oninput="resultado.value=parseInt(valor1.value) + parseInt(valor2.value)">
  Numero 1 <input type="number" name="valor1" /><br/>
  Numero 2<input type="number" name="valor2" /><br/>
  Resultado= <output name="resultado">0 </output>
</form>
```

Etiqueta `<output>`

Numero 1

Numero 2

Resultado= 3

Programación de documentos Web utilizando lenguajes scripts del lado Cliente: Javascript

Ejemplo de Validación de un formulario sencillo mediante Javascript:

[ejemplo-form-val.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Ejemplo Formulario 1</title>
  <meta charset="UTF-8">
  <script type="text/javascript" src="js/fichero.js"> </script>
</head>
<body>
  <h1>Formulario sencillo</h1>
  <form name="f1" method="get" action="recibeDatos.php">
    <table>
      <tr> <td>Usuario</td><td> <input name="login" type="text" maxlength="20" /> </td></tr>
      <tr><td>Introduce Contrase&ntilde;a;</td><td> <input name="pwd1" type="password" /> </td></tr>
      <tr><td>Repite Contrase&ntilde;a;</td><td> <input name="pwd2" type="password" /> </td></tr>
      <tr>
        <td>Deportes practicados:</td>
        <td>
          <input type="checkbox" value="futbol" id="futbol" name="deportes" checked>
          <label for="futbol">Futbol</label>
          <input type="checkbox" value="baloncesto" id="baloncesto" name="deportes">
          <label for="baloncesto">Baloncesto</label>
          <input type="checkbox" value="ciclismo" id="ciclismo" name="deportes">
          <label for="ciclismo">Ciclismo</label>
        </td>
      </tr>
      <tr><td><input type="button" name="botonAceptar" value="Aceptar" onclick="validaDatos()" /></td></tr>
    </table>
  </form>
</body>
</html>
```

[fichero.js](#)

```
function validaDatos() {
  //validando el campo login para que no sea vacio
  if(esVacio(document.f1.login.value)) {
    alert("Por favor, introduce login");
    document.f1.login.focus();
    return;
  }
  //Compruebo que el campo contraseña 1 no esté vacio
  if(esVacio(document.f1.pwd1.value)) {
    alert("Por favor, introduce tu password");
    document.f1.pwd1.focus();
    return;
  }
  //Compruebo que la longitud de la password 1 sea minimo 8 caracteres
  if(document.f1.pwd1.value.length<8) {
    alert("La password debe tener como minimo 8 caracteres");
    document.f1.pwd1.select();
    document.f1.pwd1.focus();
    return;
  }
  //Compruebo que el campo contraseña 2 no esté vacio
  if(esVacio(document.f1.pwd2.value)) {
    alert("Por favor, introduce tu password");
    document.f1.pwd2.focus();
    return;
  }
  //Compruebo que las claves coinciden
  if(!validaPwd(document.f1.pwd1.value,document.f1.pwd2.value)) {
    alert("Las claves no coinciden");
    document.f1.pwd2.select();
    document.f1.pwd2.focus();
    return;
  }
  alert("Datos introducidos OK");
  document.f1.submit();
}
function esVacio(campo) {
  if(campo.length==0)
    return true;
  else
    return false;
}
```

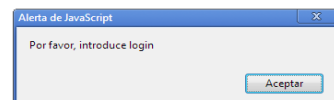
Formulario sencillo

Usuario

Introduce Contraseña

Repite Contraseña

Deportes practicados: Futbol Baloncesto Ciclismo



```
function validaPwd(pwd1,pwd2) {
  if(pwd1==pwd2)
    return true;
  else
    return false;
}
```