

Tema 9. Formularios.

Indice

1. Presentación.....	2
2. Declarar el formulario.....	3
3. Campo de texto.....	5
4. Área de texto.....	8
5. Lista desplegable.....	10
6. Botones de selección única (radio).....	13
7. Botones de selección múltiple (checkbox).....	14
8. Botón de envío.....	17
9. Botón de anulación.....	18
10. Botón de acción.....	19
11. Campos ocultos.....	22
12. Campos de transferencia de archivos.....	22
13. Campos de contraseña.....	24
14. Organización de los campos de formulario.....	25
15. Etiquetado de los campos de formulario.....	27
16. Campo de texto con lista de sugerencias.....	29
17. Campo de texto para la dirección de correo electrónico.....	30
18. Campo de texto url.....	32
19. Campo de texto con formato numérico.....	33
20. Campo de texto con formato de fecha.....	35
21. Campo de texto de búsqueda.....	38
22. Campo de texto de color.....	39
23. Cursores.....	40
24. Una aplicación completa.....	43

1. Presentación

Los formularios ocupan un lugar predominante en el diseño y explotación de una aplicación o un sitio Web. Es, en efecto, la única forma de recibir de vuelta la información que proviene directamente del usuario final, estructurada, además, según las necesidades del diseñador. Basta con pensar en todos los sitios de carácter comercial en los que los formularios resultan indispensables.

La primera preocupación de los desarrolladores de una aplicación Web es recuperar datos válidos. Tomemos, por ejemplo, el dato de un código postal español, de cinco caracteres numéricos, o una dirección de correo electrónico en la que la sintaxis sea válida. Una de las primeras aplicaciones de JavaScript ha sido (y sigue siendo) la validación directa (del lado cliente) de los datos introducidos por el usuario. De este modo, todos los formularios de cierto nivel incluyen, además del código Html, una buena dosis de JavaScript para realizar la validación de los datos. Una de las ideas principales de Html5 consiste en que el lenguaje Html pueda realizar (y, por tanto, que los navegadores puedan realizar) la validación de los campos de los formularios, liberando así al código fuente de scripts redundantes. Todo esto incluso en el caso en que se haya desactivado el código JavaScript en el navegador.

Los formularios forman parte del lenguaje Html desde hace ya unos quince años y no se han modificado desde entonces. Reconozcamos que gráficamente, incluso después de dotarlos con propiedades de estilo, se ha instalado cierto hastío tanto en el lado de los diseñadores como en el de los usuarios. Por otro lado, con la Web 2.0, el usuario se ha habituado a una presentación más cercana a las aplicaciones de escritorio que a las páginas Web clásicas. Citemos los calendarios, habituales en los sitios de reserva en línea, o los cursores, usados en multitud de programas.

Integrando en el lenguaje Html las especificaciones de Xforms, Html5 muestra un importante interés agregando a los formularios heredados de Html 4.0 una cantidad impresionante de funciones inéditas, aproximándose en gran medida a lo que serían las aplicaciones de escritorio. La integración de estos nuevos campos de formulario en los navegadores se está haciendo muy progresivamente y habrá que esperar todavía algunos años antes de que todos los navegadores los gestionen por completo. Pero es innegable que Html5 ha abierto, para todos los diseñadores y desarrolladores, considerables perspectivas al respecto.

El propósito último de los formularios es el procesamiento automático de los datos devueltos. Por ello es preciso mencionar, del lado del servidor, los lenguajes de

programación de gestión de bases de datos como, por ejemplo, PHP o MySQL. En este aspecto resulta imprescindible un aprendizaje que rebasa el marco de este módulo. Entramos aquí en un terreno reservado a los programadores amateurs más avanzados, o profesionales. Afortunadamente para los debutantes existe el protocolo mailto, que permite enviar datos de formularios a una dirección de correo electrónico. Si bien este sistema presenta la ventaja de que puede usarlo todo el mundo, sin tener que recurrir a recursos externos, conlleva no obstante ciertas limitaciones. Además, el procesamiento posterior de los datos tendrá que hacerse de forma manual.

2. Declarar el formulario.

La etiqueta `<form> ... </form>` tiene como única función la de declarar al navegador que debe incluir un formulario. Esta etiqueta englobará los elementos o campos de formulario como, por ejemplo, una línea de texto, casillas a marcar, listas desplegadas, etc.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Elementos del formulario
</form>
</body>
</html>
```

En sí, la etiqueta `<form>` no muestra nada en la ventana del navegador. No tiene sentido mostrar una captura de pantalla.

Los atributos usuales de la etiqueta `<form>` son:

name

Para asignar un nombre (name) al formulario.

action

Cuando demos la orden al navegador de enviar los datos del formulario, aquél precisa conocer la acción que debe realizar.

Esta acción será:

- Bien la llamada a un programa de procesamiento de datos, alojado en el servidor, en CGI, Perl, PHP, ASP...

Por ejemplo:

```
action="http://www.servidor/procesamiento.php"
```

- Bien una dirección de correo electrónico para recuperar simplemente los datos. Se utiliza el protocolo mailto.

Por ejemplo:

```
action="mailto:mi_correo@servidor"
```

- Bien el procesamiento de los datos del formulario de forma interna (de lado cliente) mediante JavaScript, en cuyo caso el atributo **action** está vacío.

Por ejemplo:

```
action=""
```

Si usas el validador de W3C (<http://validator.w3.org>), verás que la presencia del atributo action es obligatoria.

enctype

El atributo enctype especifica en qué formato informático (mime type) se transmitirán los datos del formulario. El valor por defecto es `application/x-www-form-urlencoded`. Para enviar archivos (véase la sección Campos de transferencia de archivos en este capítulo), este campo debe valer `multipart/form-data` y, por último, para el envío de datos a una dirección de correo electrónico mediante el protocolo `mailto`, el valor será `text/plain`.

method

La transmisión de datos de un formulario se realiza mediante el método GET o el método POST. El método GET realiza la transferencia de los datos en caracteres ASCII y los datos no pueden exceder los 100 caracteres. El método POST gestiona los caracteres no ASCII y tiene una capacidad de caracteres ilimitada. En la práctica se ha impuesto el método POST por motivos de facilidad y de eficacia.

Cuando los datos del formulario se procesan de forma interna (`action=""`), los atributos **method** y **enctype** son inútiles puesto que no se hace llamada al servidor.

Comentarios

- Esta etiqueta, básica en Html, es perfectamente compatible en los distintos navegadores.
- La etiqueta `<form>` debe, obligatoriamente, cerrarse. En caso de olvidar la etiqueta de cierre `</form>` el navegador no mostrará ningún campo del formulario.

3. Campo de texto

`<input type="text">`

Este campo de texto permite captar la introducción de datos tanto alfabéticos como numéricos.

Ejemplo: propongamos al usuario un campo de texto para introducir su nombre.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Nombre: <input type="text">
</form>
</body>
</html>
```

Comentarios

- Esta etiqueta, básica en Html, es perfectamente compatible en los distintos navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="text" />`, inspirada en Xhtml, está aceptada igualmente en Html5.

Los posibles atributos son:

name

Define un nombre (name) único para este elemento. Este atributo se utiliza para recoger el contenido del campo de texto cuando se envía el formulario.

size

Define el número de caracteres visibles del campo de texto y, por tanto, la longitud de la zona de texto. El usuario puede, no obstante, introducir tantos caracteres como quiera, incluso aunque sobrepasen la zona visible. El valor por defecto de este campo es 20.

maxlength

Determina el número máximo de caracteres que el usuario puede introducir en el campo de texto. Este atributo es particularmente útil para datos con un número concreto de

caracteres como, por ejemplo, cinco cifras para introducir un código postal español.

value

Define el valor por defecto del campo de texto. Aparece como valor del campo de texto cuando se carga la página.

Por ejemplo:

```
<input type="text" value=";Escriba su nombre aquí!">
```

readonly

Indica que el usuario no puede modificar el valor por defecto del campo de texto.

En Html5 han aparecido nuevos atributos. Actualmente no todos los navegadores los soportan. He aquí algunos ejemplos:

placeholder

Permite incluir una sugerencia acerca del valor que se debe introducir en el campo de texto. Esta sugerencia aparece de color gris en el campo de formulario tras cargar la página y desaparece cuando el usuario sitúa el foco en el elemento afectado.

Este atributo ya se encuentra implementado en Internet Explorer 10.0+, Firefox 4.0+, Chrome 4.0+, Safari 5.0+, Opera 11.6+, iOS Safari 3.2+ y Android Browser 2.1+. Este atributo se ignora en los navegadores que no lo implementan.

```
<input type="text" placeholder="Su apellido">
```

Hay quien podrá señalar que el atributo **value** realiza la misma función. La diferencia es sutil. Con **value**, el usuario tiene que borrar el contenido antes de introducir su propio texto. Con **placeholder**, el texto se borra automáticamente cuando se sitúa el foco sobre el elemento afectado. Además, con el atributo **value**, en caso de que se envíe el formulario sin haber modificado el valor de esta propiedad por defecto, será éste el valor que se envíe.

autofocus

Sitúa el foco sobre el elemento una vez cargada la página. Este atributo ahorra algunas líneas de JavaScript que realizan la misma función.

Este atributo está implementado actualmente en Internet Explorer 10+, Firefox 4.0+, Chrome 6.0+, Safari 5.0+ y Opera 11+.

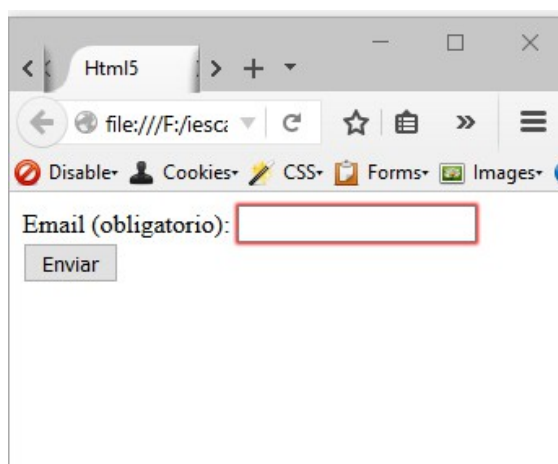
required

Convierte un elemento en obligatorio a la hora de enviar el formulario para su

procesamiento posterior. Es muy útil para definir aquellos elementos obligatorios como el nombre o la dirección de correo electrónico del usuario.

Esta etiqueta está implementada en Internet Explorer 10 y las versiones más recientes de Firefox, Google Chrome y Opera.

```
<form action="">  
Email (obligatorio):  
<input type="text" name="mail" required><br>  
<input type="submit" value="Enviar">  
</form>
```



Sería interesante tener, no obstante, una representación gráfica un poco más elaborada desde el punto de vista estético.

Pattern

Define una expresión regular JavaScript que se usará para realizar la validación del valor del campo. Por ejemplo `pattern="[0-9]"` indica que el valor del elemento debe ser un número entero comprendido entre 0 y 9.

Tomemos como ejemplo un campo de texto que va a contener un código postal (5 dígitos).

```
<form action="">  
Código postal : <input type="text" pattern="[0-9]{5}"><br>  
<input type="submit">  
</form>
```

Esta validación directa a través del atributo `pattern` sin tener que hacer una llamada a JavaScript, está implementada actualmente en Internet Explorer 10+, Firefox 4.0+, Chrome 10.0+, Opera 9.5+, pero no en Safari.

height

Determina, en píxeles o en porcentaje, la altura del campo de texto. Puede reemplazarse por la propiedad CSS `height`.

width

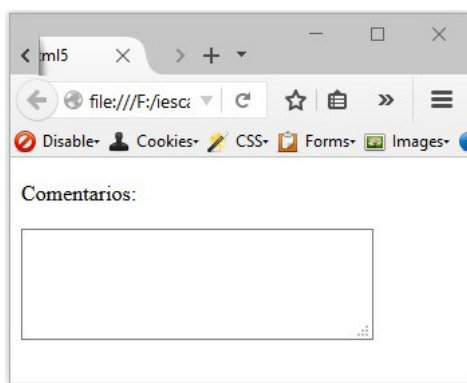
Determina, en píxeles o en porcentaje, la anchura del campo de texto. Puede reemplazarse por la propiedad CSS `width`.

4. Área de texto.

En ciertas situaciones, es preciso prever algo más de espacio para que el usuario pueda expresarse. Es el caso, por ejemplo, de los comentarios o sugerencias. Se utiliza en este caso la etiqueta `<textarea> ... </textarea>` que incluye un área de texto de varias líneas.

Ejemplo: un área de texto dedicada para los comentarios del usuario:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<p>Comentarios: </p>
<textarea rows="4" cols="30"></textarea>
</form>
</body>
</html>
```



Comentarios

- Esta etiqueta, básica en *Html*, es perfectamente compatible hacia atrás en los distintos navegadores.
- La etiqueta **<textarea>** precisa una etiqueta de cierre **</textarea>**.
- Es posible indicar un valor por defecto en el área de texto, no mediante el atributo *value* sino de la forma siguiente:

```
<textarea rows="4" cols="25">¡Escriba aquí sus sugerencias!  
</textarea>
```

Los posibles atributos son:

name

Define un nombre para el elemento.

cols

Define el número de caracteres visibles en anchura para el área de texto. Puede reemplazarse por la propiedad CSS *width*.

rows

Define el número de líneas visibles en altura para el área de texto. Puede reemplazarse por la propiedad CSS *height*.

readonly

Indica que el valor informado por defecto en el área de texto no puede modificarse.

En *Html5* han aparecido nuevos atributos. Actualmente no todos los navegadores los soportan. He aquí algunos ejemplos:

autofocus

Sitúa el foco sobre el elemento una vez cargada la página.

maxlength

Determina el número máximo de caracteres que el usuario puede introducir en el área de texto. Permite limitar a los usuarios prolijos.

required

Hace que sea obligatorio informar este elemento.

wrap

Especifica la forma en que se gestionan los saltos automáticos de línea en el texto a la hora de enviar el contenido. Con `wrap="hard"`, se inserta un carácter de salto de línea junto con el texto. Con `wrap="soft"` (por defecto), no se inserta ningún carácter de salto de línea a la hora de transmitir la información.

Para los puristas, este atributo ya no formaba parte de la especificación Html 4.0. Se ha vuelto a introducir en la especificación Html5, no obstante bajo un aspecto modificado.

placeholder

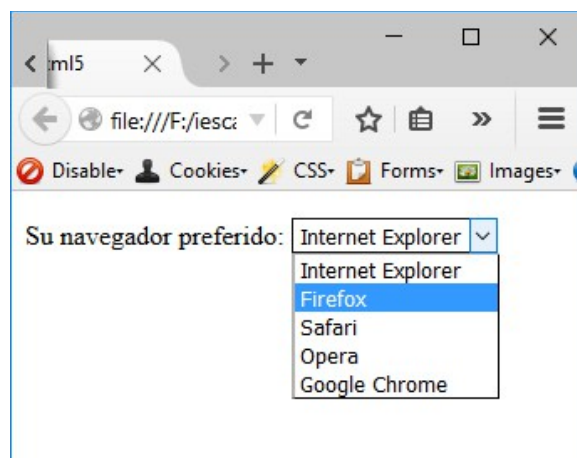
Inserta un texto en el campo del formulario tras la carga de la página. Este texto se borra automáticamente cuando el usuario sitúa el foco en el área de texto

5. Lista desplegable

La lista desplegable proporciona distintas opciones al usuario.

Ejemplo: preguntemos al usuario cuál es su navegador preferido:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<p>Su navegador preferido:
<select>
<option value="1">Internet Explorer</option>
<option value="2">Firefox</option>
<option value="3">Safari</option>
<option value="4">Opera</option>
<option value="5">Google Chrome</option>
</select>
</p>
</form>
</body>
</html>
```



Comentarios

- La etiqueta `<select> ... </select>` indica al navegador el uso de una lista desplegable. Los elementos de la lista se introducen mediante las etiquetas `<option> ... </option>`.
- Esta etiqueta es perfectamente compatible entre los distintos navegadores.

Los posibles atributos son:

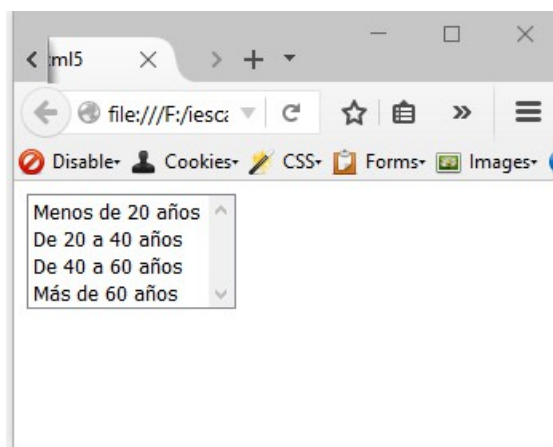
name

Define un nombre para la lista desplegable, para un posible procesamiento posterior.

size

Por defecto, el atributo `size` de la etiqueta **<select>** es igual a 1, lo que resulta muy práctico dado que permite ganar bastante espacio en la disposición de la página. No obstante, mediante el atributo `size="x"` puede definir el número de elementos del menú que será visible. ¡En este caso su menú no tendrá nada de desplegable!

```
<body>
<form action="">
<select size="4">
<option>Menos de 20 años</option>
<option>De 20 a 40 años</option>
<option>De 40 a 60 años</option>
<option>Más de 60 años</option>
</select>
</form>
</body>
```



multiple

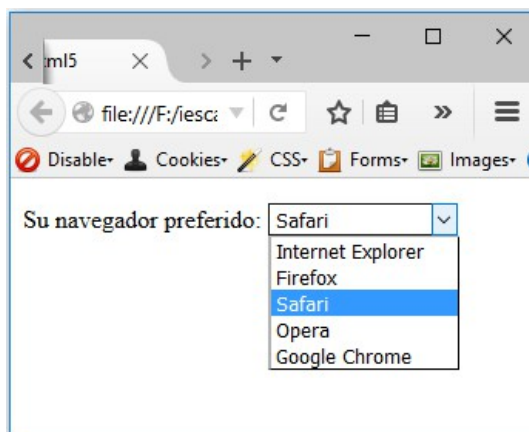
Por defecto, el usuario no puede seleccionar más de un elemento del menú desplegable. Con el atributo múltiple de la etiqueta `<select>`, es posible seleccionar varios elementos a la vez. Para hacer esto, el usuario debe mantener pulsada la tecla **[Ctrl]** en el teclado y hacer clic sobre los elementos con el ratón. En tal caso es aconsejable recordar al usuario en la página esta forma de proceder, poco común para un internauta intermedio.

El atributo `size` debe especificarse y ser igual al número de etiquetas `<option>`.

selected

Por defecto aparece seleccionado el primer elemento de la lista. No obstante es posible preseleccionar algún otro elemento mediante el atributo `selected` de la etiqueta `<option>`.

```
<body>
<form action="">
<p>Su navegador preferido:
<select>
<option value="IE">Internet Explorer</option>
<option value="FF">Firefox</option>
<option value="S" selected>Safari</option>
<option value="O">Opera</option>
<option value="GC">Google Chrome</option>
</select>
</p>
</form>
</body>
```



value

En principio es el texto del elemento seleccionado, situado tras la etiqueta `<option>`, el que se transmite a la hora de enviar el formulario. No obstante es posible especificar un valor diferente (generalmente un valor numérico) para transmitirlo, mediante el atributo `value="valor"`.

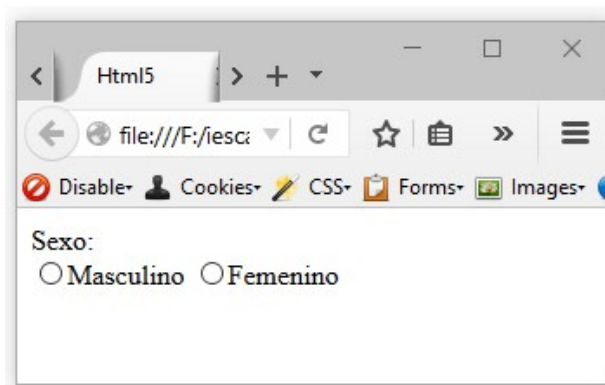
Dejando a parte `autofocus`, no hay nuevos atributos significativos en Html5.

6. Botones de selección única (radio)

Los botones de selección única, también llamados botones radio, tienen la particularidad de que sólo permiten seleccionar una opción a la vez (el "o" exclusivo).

Ejemplo: preguntemos el sexo del interlocutor:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Sexo:<br>
<input type="radio" name="sexo">Masculino
<input type="radio" name="sexo">Femenino
</form>
</body>
</html>
```



Comentarios

- *Esta etiqueta es perfectamente compatible entre los distintos navegadores.*
- *La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type = "radio"/>`, inspirada en Xhtml, está igualmente aceptada en Html5.*

Los posibles atributos son:

name

Aquí el atributo `name` es obligatorio. Además, en el caso de los botones radio, el nombre debe ser idéntico para todos los botones.

checked

Permite preseleccionar un botón radio.

value

En vista de un procesamiento posterior, es posible atribuir un valor a cada botón radio mediante el atributo `value="valor"`.

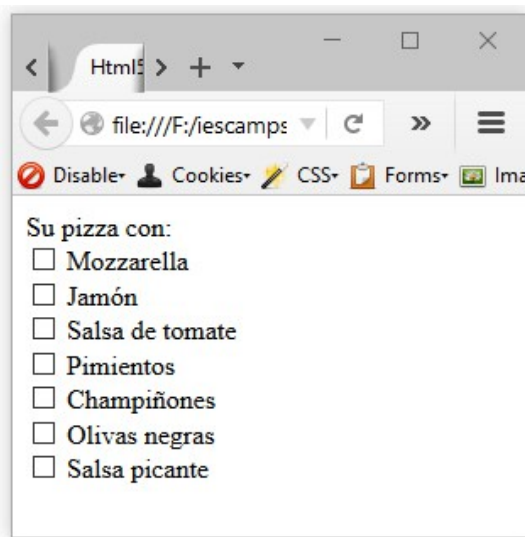
Dejando a parte autofocus, no hay nuevos atributos significativos en Html5.

7. Botones de selección múltiple (checkbox)

El uso de estos botones de selección múltiple, también llamados checkbox, es similar a los botones de selección única aunque, en este caso, es posible seleccionar varias opciones.

Ejemplo: pidamos al usuario que precise los ingredientes de una pizza.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Su pizza con:<br>
<input type="checkbox" name="n1"> Mozzarella<br>
<input type="checkbox" name="n2"> Jamón<br>
<input type="checkbox" name="n3"> Salsa de tomate<br>
<input type="checkbox" name="n4"> Pimientos<br>
<input type="checkbox" name="n5"> Champiñones<br>
<input type="checkbox" name="n6"> Olivas negras<br>
<input type="checkbox" name="n7"> Salsa picante
</form>
</body>
</html>
```



Comentarios

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="checkbox" />`, inspirada en Xhtml, está igualmente aceptada en Html5.

Los posibles atributos son:

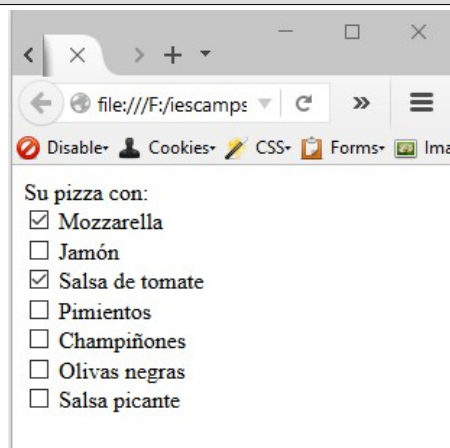
name

El atributo `name="nombre"` es obligatorio. Las reglas son menos precisas que para los botones de opción. Es posible usar nombres idénticos o nombres diferentes para cada casilla. No obstante, es preciso indicar nombres diferentes para tratar los elementos dentro de un script.

checked

El atributo `checked` permite preseleccionar una casilla.

```
<form action="">
Su pizza con:<br>
<input type="checkbox" name="n1" checked > Mozzarella<br>
<input type="checkbox" name="n2"> Jamón<br>
<input type="checkbox" name="n3" checked > Salsa de tomate<br>
<input type="checkbox" name="n4"> Pimientos<br>
<input type="checkbox" name="n5"> Champiñones<br>
<input type="checkbox" name="n6"> Olivas negras<br>
<input type="checkbox" name="n7"> Salsa picante
</form>
```



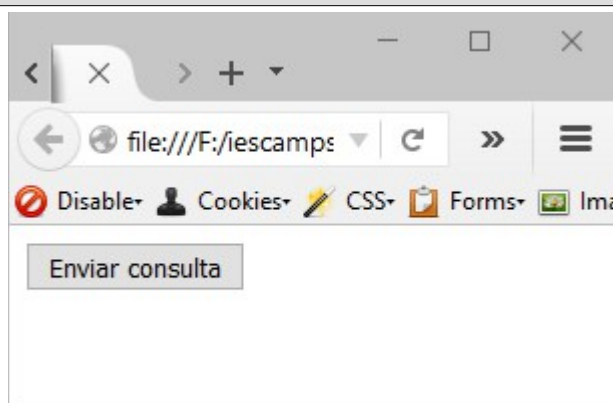
value

Para poder procesarlos posteriormente es preciso indicar un valor a cada botón checkbox mediante el atributo `value="valor"`.

Dejando a parte autofocus, no hay nuevos atributos significativos en Html5.

8. Botón de envío

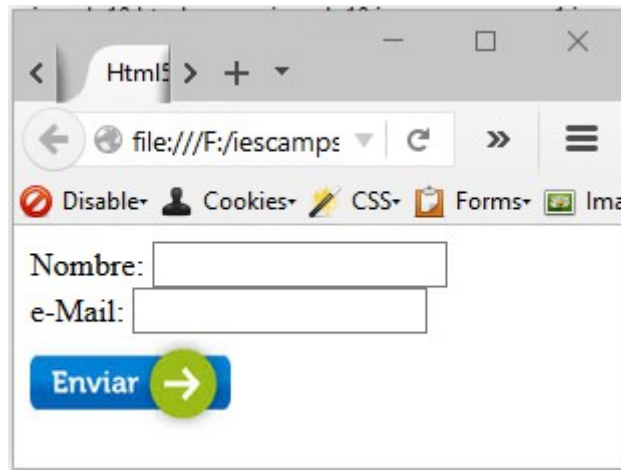
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<input type="submit">
</form>
</body>
</html>
```



Comentarios

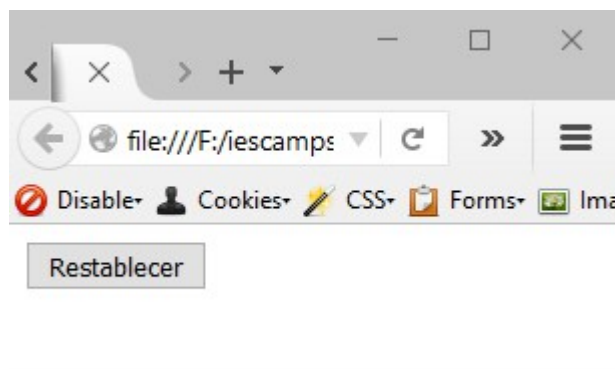
- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="submit" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- **El texto del botón lo selecciona el navegador. En efecto, varía de un navegador a otro: Enviar consulta en Internet Explorer y Firefox, Enviar en Opera, Safari y Google Chrome. Es posible modificar el texto por defecto del botón mediante el atributo value. La sintaxis es, por ejemplo, `<input type="submit" value="Enviar el formulario">`.**
- Es posible reemplazar el botón de envío por una imagen gráfica gracias a la etiqueta `<input type="image">`.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Nombre: <input type="text"><br>
e-Mail: <input type="text"><br>
<input type="image" src="submit.gif" alt="Enviar" width="50"
height="50">
</form>
</body>
</html>
```



9. Botón de anulación

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<input type="reset">
</form>
</body>
</html>
```



Comentarios

- *Esta etiqueta es perfectamente compatible entre los diferentes navegadores.*
- *La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="reset" />`, inspirada en Xhtml, está igualmente aceptada en Html5.*
- *El título del botón lo selecciona el navegador. La mayoría de ellos optan por Restablecer, salvo Safari que prefiere Restaurar. Es posible modificar el texto por defecto del botón mediante el atributo **value**. La sintaxis es, por ejemplo, `<input type="reset" value="Reiniciar">`.*

10. Botón de acción.

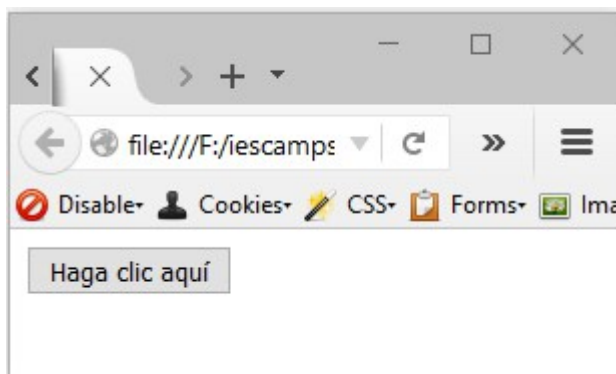
En materia de botones (button), existe también la etiqueta `<button> ... </button>`. Esta etiqueta permite ejecutar, al hacer clic sobre el botón, una acción específica definida por el diseñador del sitio Web, generalmente mediante algún código de JavaScript. Ofrece igualmente la posibilidad de realizar el envío y la anulación del formulario (submit y reset).

Por otro lado, esta etiqueta `<button>` tiene la ventaja de tener una etiqueta de apertura y otra de cierre. De este modo es posible incluir texto, imágenes o incluso contenido Html. Destaquemos también que esta etiqueta `<button>` no tiene por qué estar anidada obligatoriamente en un formulario `<form>` y puede usarse en múltiples contextos.

Todo esto la convierte en una etiqueta polivalente y justifica su éxito entre los desarrolladores.

Ejemplo: Al hacer clic en el botón, se abre una ventana de alerta:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<button onclick="alert('Acción JavaScript')">
Haga clic aquí</button>
</body>
</html>
```



Comentarios

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<button>` tiene una etiqueta de cierre: `</button>`.
- El texto del botón se declara entre ambas etiquetas, y no en el atributo `value`. Por ejemplo `<button>Haga clic aquí</button>`.

Los posibles atributos son:

name

Especifica un nombre para el botón.

type

El atributo `type` admite tres argumentos:

- **button**: se utiliza cuando el botón inicia un script (es el valor por defecto del atributo en la mayoría de navegadores).
- **submit**: para enviar el formulario.
- **reset**: para reinicializar el formulario.

Los sitios Web y los foros especializados recomiendan especificar siempre el valor del atributo type puesto que el argumento button no es el valor por defecto en todos los navegadores (no lo es, por ejemplo, en Internet Explorer 8 en modo Xhtml Strict).

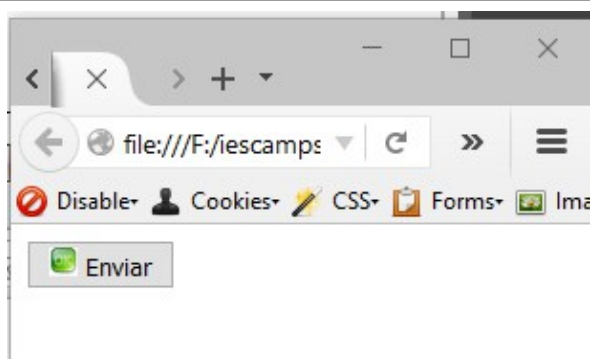
disabled

Permite desactivar el botón cuando se carga la página. En este caso el botón aparece sombreado en gris. Es posible activarlo (enabled) con un script mediante una acción del usuario, por ejemplo después de haber aceptado las condiciones de uso.

Dejando a parte autofocus, no hay nuevos atributos significativos en Html5.

Veamos la posibilidad de asignar una imagen a la etiqueta <button> agregando un pequeño icono (ok.png) disponible en la página Información.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<button>
Enviar
</button>
</body>
</html>
```



11. Campos ocultos.

Estos campos permiten almacenar datos que no estarán visibles al visitante de la página. Este tipo de elemento lo usan a menudo los programadores experimentados (especialmente en JavaScript) porque les permite almacenar datos de formularios anteriores, datos en bruto, o datos que provienen de algún otro script.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<input type="hidden" name="c" value="3.1416">
</body>
</html>
```

No tiene sentido realizar una captura de pantalla dado que el campo del formulario está oculto (hidden).

Comentarios

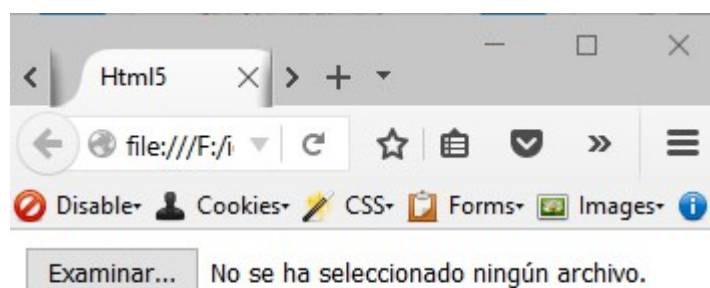
- *Esta etiqueta es perfectamente compatible entre los diferentes navegadores.*
- *La etiqueta <input> no tiene etiqueta de cierre. La sintaxis <input type="hidden" />, inspirada en Xhtml, está igualmente aceptada en Html5.*

12. Campos de transferencia de archivos.

La etiqueta <input type="file"> permite transferir un archivo (file) desde el puesto del usuario a otro ordenador de tipo servidor.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="" method="post" enctype="multipart/form-data">
<input type="file">
</form>
</body>
</html>
```



Un clic en el botón Examinar (de nombre variable según el navegador) abre el explorador de carpetas del ordenador del usuario para seleccionar el archivo que se quiere transferir.

Comentarios

- *Esta etiqueta es perfectamente compatible entre los diferentes navegadores.*
- *La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="file" />`, inspirada en Xhtml, está igualmente aceptada en Html5.*
- *Los atributos habituales de la etiqueta `<input>` pueden utilizarse igualmente. Destacamos, no obstante, el atributo `maxlength` que permite fijar el tamaño máximo del archivo a transferir.*
- *En la declaración de la etiqueta `<form>`, es preciso usar los atributos `method="post"` y `enctype="multipart/form-data"` para realizar la*

transferencia según el formato correcto del archivo.

- Existe también el atributo *accept* que permite limitar la transferencia a ciertos tipos de archivos concretos. Por ejemplo archivos de texto (txt), Word (doc), Excel (xls), pdf, archivos de imagen (jpeg, gif o png), etc. En la descripción de los archivos es posible usar el carácter * como comodín.
- Destacamos que esta etiqueta *Html* no sirve más que para seleccionar el archivo que se quiere transferir. La transferencia en sí debe tenerse en cuenta las aplicaciones del lado servidor, como por ejemplo PHP.

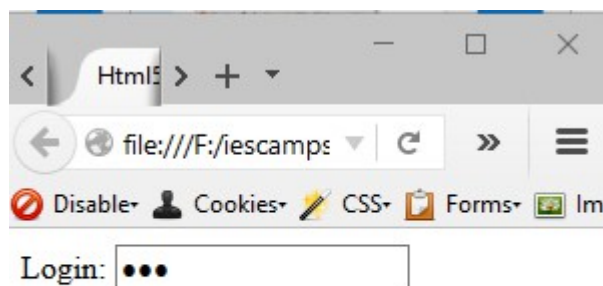
13. Campos de contraseña

Este tipo de campo es, de hecho, un simple campo de texto donde la codificación se sustituye, en la visualización, por puntos o asteriscos.

Este campo de contraseña no protege en ningún caso los datos, que se transmitirán sin cifrar. Únicamente nos protegen de las personas que pudieran estar mirando la pantalla en el momento de la introducción de la contraseña.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Login:
<input type="password">
</form>
</body>
</html>
```



Comentarios

- Esta etiqueta es perfectamente compatible entre los diferentes navegadores.
- La etiqueta `<input>` no tiene etiqueta de cierre. La sintaxis `<input type="password" />`, inspirada en Xhtml, está igualmente aceptada en Html5.
- Los atributos del campo de texto también pueden usarse aquí: `name`, `size`, `maxlength`, etc.

Dejando a parte `autofocus`, no hay nuevos atributos significativos en Html5.

14. Organización de los campos de formulario.

En el caso de formularios largos y complejos, en ocasiones resulta útil agrupar gráficamente ciertos elementos para organizar la página de forma lógica. Las etiquetas `<fieldset>` y `<legend>` permiten mejorar sensiblemente la ergonomía y la usabilidad de los formularios.

La etiqueta `<fieldset> ... </fieldset>` engloba los campos de formulario que determines. Estos campos se muestran en la pantalla rodeados por un borde.

La etiqueta `<legend> ... </legend>`, situada justo después de la etiqueta `<fieldset>`, agrega una leyenda relacionada con aquellos campos que hayamos agrupado mediante la etiqueta `<fieldset>` (véase la siguiente captura de pantalla).

Ejemplo: agrupemos los entrantes, las pizzas y los postres en una lista de selección múltiple.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<fieldset>
<legend>Nuestros entrantes</legend>
<input type="checkbox" name="n1"> Chikenito's<br>
<input type="checkbox" name="n2"> Ensalada de temporada<br>
```

```
<input type="checkbox" name="n3"> Buffalo wings<br>
</fieldset>
<fieldset>
<legend>Nuestras pizzas</legend>
<input type="checkbox" name="n4"> Clásica<br>
<input type="checkbox" name="n5"> Campesina<br>
<input type="checkbox" name="n6"> Diabólica<br>
</fieldset>
<fieldset>
<legend>Nuestros postres</legend>
<input type="checkbox" name="n7"> Helados variados<br>
<input type="checkbox" name="n8"> Cookies<br>
<input type="checkbox" name="n9"> Tarta de chocolate<br>
</fieldset>
<input type="submit"> <input type="reset">
</form>
</body>
</html>
```

Ejemplo: agrupemos los entrantes, las pizzas y los postres en una lista de selección múltiple.

Nuestros entrantes

- ☐ Chickenito's
- ☐ Ensalada de temporada
- ☐ Buffalo wings

Nuestras pizzas

- ☐ Clásica
- ☐ Campesina
- ☐ Diabólica

Nuestros postres

- ☐ Helados variados
- ☐ Cookies
- ☐ Tarta de chocolate

Comentarios

- *Esta etiqueta es perfectamente compatible entre los diferentes navegadores.*
- *No hay que olvidar incluir las etiquetas de cierre.*
- *Ya no existe una alineación de la leyenda a la derecha o a la izquierda en Html5. Esta operación debe realizarse mediante una hoja de estilos CSS.*

15. Etiquetado de los campos de formulario

La etiqueta `<label>` asocia explícitamente el título a un campo de formulario particular. Es como si pegáramos una etiqueta (label) delante de un elemento del formulario.

En un primer momento, la etiqueta `<label>` mejora en gran medida la ergonomía de los formularios permitiendo activar un elemento del grupo, por ejemplo un botón radio, haciendo clic sobre el botón radio o sobre el título, indiferentemente.

No obstante las etiquetas `<label>` son particularmente útiles en el dominio de la accesibilidad de los sitios Web por parte de personas invidentes. Estas "etiquetas" se tienen en cuenta en las ayudas técnicas (intérpretes de braille o programas de síntesis vocal) y facilitan en gran medida el uso de los formularios por parte de personas visualmente deficientes.

En primer lugar el texto asignado a un elemento de formulario debe situarse entre las etiquetas `<label> ... </label>`.

```
<label>Nombre</label>;  
<input type="text"><br>
```

A continuación, es preciso asociar esta etiqueta label al control del formulario correspondiente. Para ello, el elemento de formulario tiene que definirse mediante un identificador de tipo id.

```
<label>Nombre</label>;  
<input type="text" id="f1"><br>
```

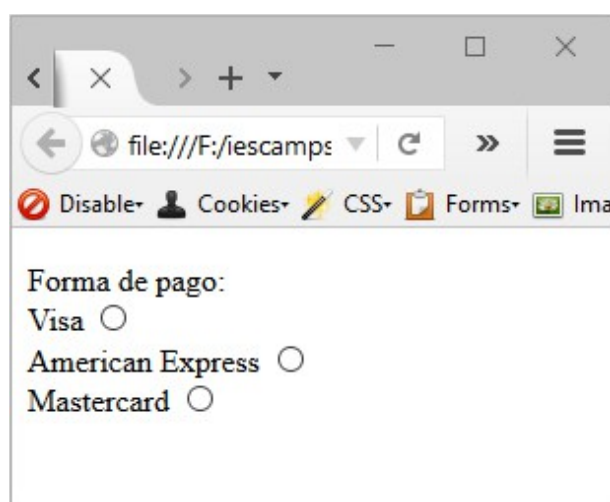
El atributo `for="..."` dentro de la etiqueta `<label>` permite asociar directamente la etiqueta al campo haciendo referencia a este identificador.

```
<label for="f1">Nombre</label>:
```

```
<input type="text" id="f1"><br>
```

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
<p>Forma de pago:<br>
<label for="cash1">Visa</label>
<input type="radio" name="cash" id="cash1"><br>
<label for="cash2">American Express</label>
<input type="radio" name="cash" id="cash2"><br>
<label for="cash3">Mastercard</label>
<input type="radio" name="cash" id="cash3">
</p>
</form>
</body>
</html>
```



No existen novedades significativas en Html5.

16. Campo de texto con lista de sugerencias

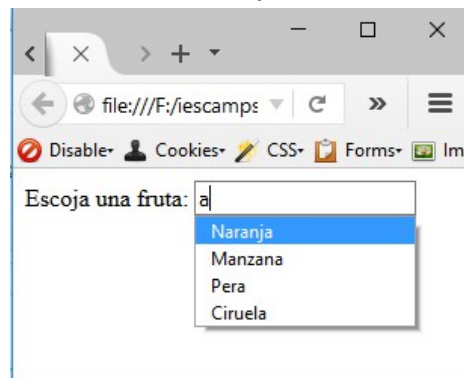
La etiqueta `<datalist>`, agregada a un campo de texto, abre una lista de sugerencias para introducir el dato sobre el que se ha situado el foco. Este comportamiento es similar al de **Google Suggest** a la hora de introducir alguna palabra clave en la barra de búsqueda.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Escoja una fruta:
<input type="text" list="frutas">
<datalist id="frutas">
<option value="Naranja"></option>
<option value="Manzana"></option>
<option value="Pera"></option>
<option value="Ciruela"></option>
</datalist>
</form>
</body>
</html>
```

La etiqueta `<datalist>` contiene una lista de etiquetas `<option>` con las sugerencias. Estas etiquetas `<option>` deben poseer un atributo `value`.

La etiqueta `<datalist>` está ligada a un campo de formulario mediante un identificador `id` que se asocia con el atributo `list` de aquél.



Esta etiqueta está implementada en Internet Explorer 10+, Firefox 4.0+, Chrome 20.0+ y Opera 10+.

17. Campo de texto para la dirección de correo electrónico.

¿Cuántas veces le han pedido su dirección de correo electrónico navegando por la red?

¡Y qué rompecabezas supone a los diseñadores de las aplicaciones Web validar una dirección de correo electrónico!

Html5 responde a esta necesidad con la etiqueta `<input type="email">`.

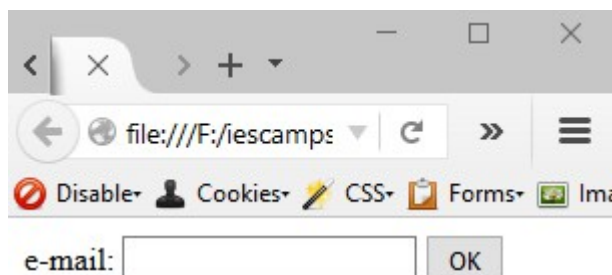
Esta etiqueta está implementada en Internet Explorer 10 y en las versiones actuales de Firefox, Google Chrome y Opera.

Se convierte en una etiqueta particularmente interesante a la hora de realizar la validación directa en el navegador, sin que el diseñador tenga que agregar JavaScript.

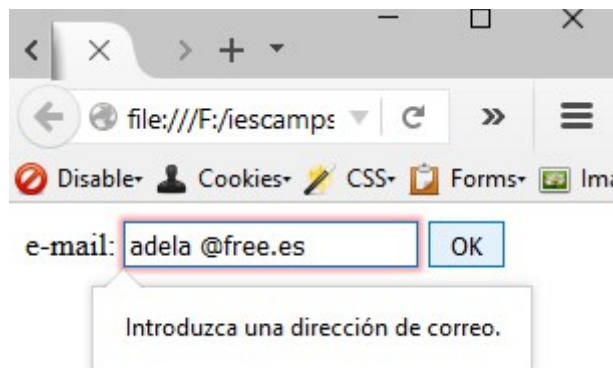
Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
e-mail: <input type="email" name="mail" required>
<input type="submit" value="OK">
</form>
</body>
</html>
```

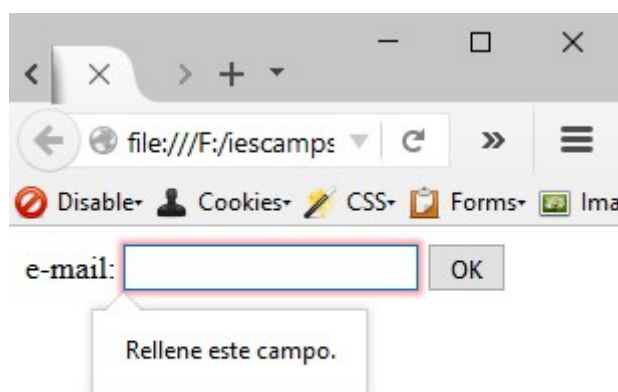
A priori, el campo de texto no se distingue de un campo de texto normal.



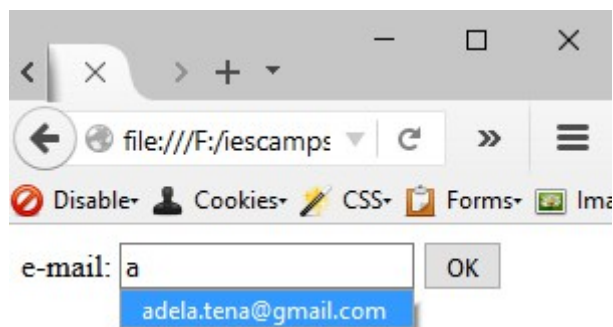
No obstante, cuando intentamos enviar una dirección de correo electrónico mal formada, el navegador nos informa de que la dirección es incorrecta.



Por otro lado, es posible agregar el atributo `required` que indica que se trata de un campo obligatorio.



Aprovechamos esta etiqueta `email` para hablar del atributo `autocomplete` de HTML5, disponible en varios elementos de los formularios. Cuando esta opción está activada en las preferencias del navegador, este atributo proporciona la opción de completar automáticamente el campo de formulario con su dirección de correo electrónico.



18.Campo de texto url.

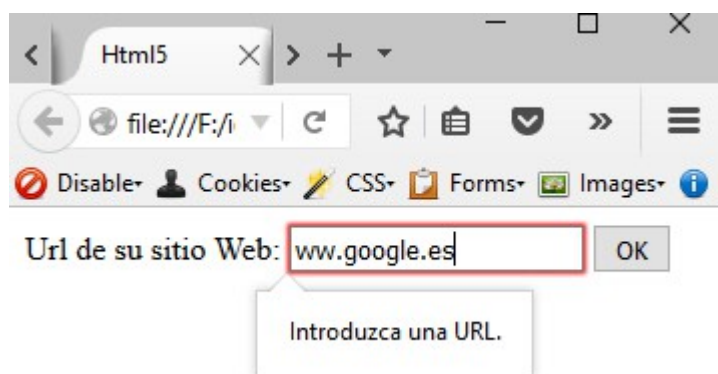
Con un comportamiento similar a la etiqueta anterior `<input type="email">`, la etiqueta `<input type="url">` se define para introducir direcciones Web (url).

Esta etiqueta se ha implementado en Internet Explorer 10 y en las versiones recientes de Firefox, Google Chrome y Opera.

En este caso también se realiza una validación de forma directa (sin JavaScript) en el propio navegador.

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Url de su sitio Web: <input type="url" name="web">
<input type="submit" value="OK">
</form>
</body>
</html>
```



19. Campo de texto con formato numérico.

Otro campo nuevo incluido en Html5 son los contadores numéricos. Estos contadores, que se encuentran frecuentemente en aplicaciones como los procesadores de texto, se usan también en aplicaciones Web.

Esta nueva etiqueta `<input type="number">` tiene atributos específicos:

min

Indica el valor mínimo del contador.

max

Indica el valor máximo del contador.

step

Determina el incremento del contador cada vez que se hace clic en las flechitas `images/plus.png` o `images/moins.png`.

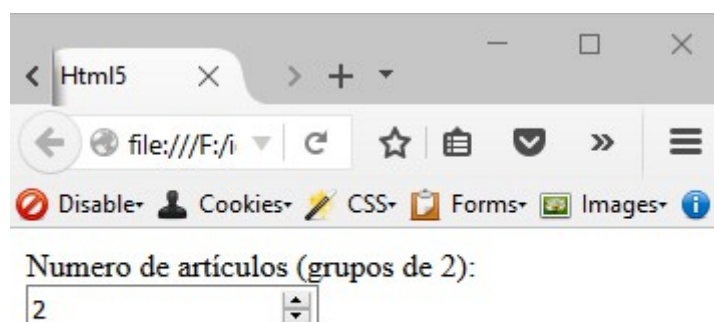
value

Indica el valor inicial del contador.

Ejemplo

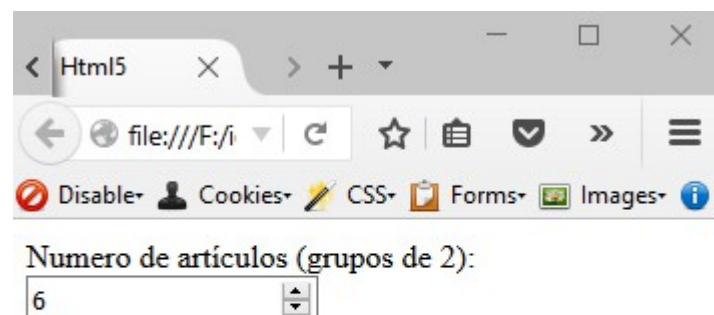
```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Numero de articulos (grupos de 2):<br>
<input type="number" name="q" min="2" max="10" step="2" value="2">
</form>
</body>
</html>
```

Al inicio:



A screenshot of a web browser window. The address bar shows a file path: file:///F:/i. Below the address bar, there are several icons for browser features: Disable, Cookies, CSS, Forms, Images, and an information icon. The main content area displays the text "Numero de artículos (grupos de 2):" followed by a spinner control (a text box with up and down arrows) containing the number 2.

Haciendo clic en las flechitas el usuario puede aumentar o disminuir el contador.



A screenshot of a web browser window, similar to the one above. The address bar shows the same file path. The main content area displays the text "Numero de artículos (grupos de 2):" followed by a spinner control containing the number 6.

Comentarios

- *A día de hoy, este contador se ha está implementado en Chrome 7.0+, Safari 5.0+ y Opera 9+.*
- *El código es, no obstante, compatible con el resto de navegadores que no tienen en cuenta esta nueva etiqueta y simplemente incluyen un campo de texto que debe completarse manualmente.*

20. Campo de texto con formato de fecha.

Los calendarios han cobrado especial importancia en las aplicaciones de la Web 2.0. Su uso es indiscutible en todos los sitios de reservas on-line, bien sea para reservar un billete de avión o una habitación de hotel.

Para los desarrolladores, los formatos de fecha suponen un verdadero problema, sobre todo en sitios Web de vocación internacional (dd/mm/aa o mm/dd/aa).

A día de hoy, estos calendarios se gestionaban mediante complejos scripts o incluso frameworks JavaScript (por ejemplo Dojo).

Html5 proporcionará (en un futuro) campos de formulario especialmente adaptados a la codificación de fechas que permitirán visualizar directamente en el navegador, sin el uso de plugins particulares, un calendario donde el usuario podrá escoger la fecha en el formato deseado por el diseñador.

Se trata de uno de los avances más importantes de Html5 en cuando al diseño de sitios Web.

A día de hoy, solamente Opera 10+ y Google Chrome 20.0+ implementan esta funcionalidad.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Fecha de llegada: <input type="date" name="in"><br>
Fecha de partida: <input type="date" name="out">
</form>
</body>
</html>
```

Esta etiqueta `<input>` con formato de fecha se declina en los siguientes formatos.

`<input type="date">`

La más general, ilustrada anteriormente. Permite seleccionar el año, el mes y el día.

`<input type="month">`

Para seleccionar el mes y el año. Es útil, por ejemplo, para introducir la fecha de expiración de una tarjeta de crédito.

A día de hoy sólo Opera 10+ y Chrome implementan esta funcionalidad.

Fecha de expiración: `<input type="month" name="in">`

`<input type="week">`

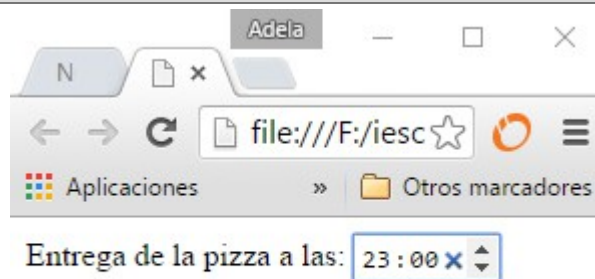
A día de hoy sólo Opera 10+ y Chrome implementan esta funcionalidad.

Para una semana determinada.

Semana reservada: `<input type="week" name="in">`

No obstante Html5 no se queda aquí. Proporciona también una herramienta para el formato horario:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Entrega de la pizza a las:
<input type="time" min="11:00" max="23:00" step="900"
value="11:00">
</form>
</body>
</html>
```



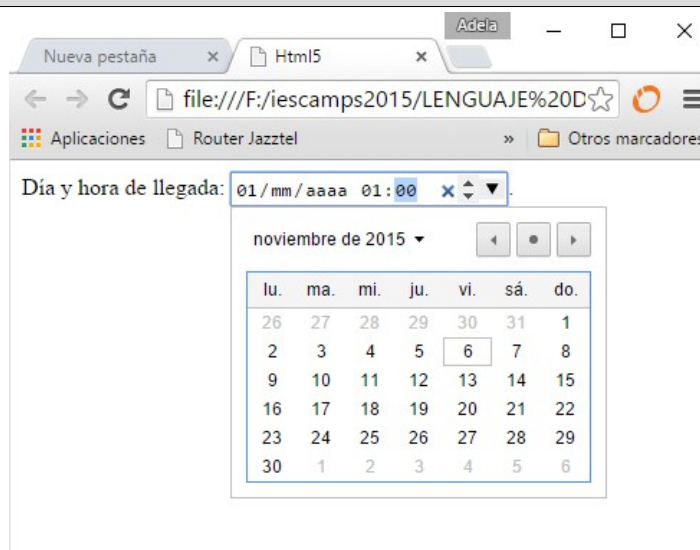
El atributo **min** permite fijar una hora de inicio del contador, **max** una hora de fin y **step** el paso de avance, en este ejemplo de 15 en 15 minutos (900 segundos) cada vez que se hace clic en una flechita.

A día de hoy, sólo Opera 10+ y Google Chrome 20.0+ implementan esta funcionalidad.

Por último, es posible combinar el calendario con el contador horario usando la etiqueta `<input type="datetime-local">`.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
```

```
<meta charset="UTF-8">
</head>
<body>
<form action="">
Día y hora de llegada:
<input type="datetime-local" name="in">
</form>
</body>
</html>
```



21. Campo de texto de búsqueda.

Son muy numerosos los sitios o aplicaciones Web que permiten realizar una búsqueda. Normalmente toman el aspecto de un campo de texto.

Html5 proporciona, con la etiqueta `<input type="search">`, un campo de texto dedicado especialmente a esta posibilidad de búsqueda.

Esta etiqueta todavía no ha sido implementada por los navegadores.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
```

```
<body>
<form action="">
Motor de búsqueda:
<input type="search">
<input type="submit" value "Go">
</form>
</body>
</html>
```

Este campo de texto dedicado a las herramientas de búsqueda debería tomar el aspecto de un campo de texto con bordes redondeados, como ofrece ya el framework jQuery Mobile.

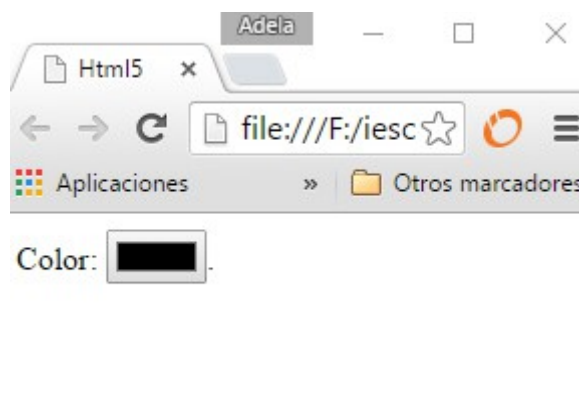


22. Campo de texto de color.

Html5 prevé igualmente un campo de texto destinado a recibir la codificación de un color (en formato hexadecimal) con la etiqueta `<input type="color">`.

Cuando se sitúa el foco en este campo de texto, se abre automáticamente una paleta de colores para realizar la selección del color, cuyo valor se reporta.

A día de hoy, sólo Chrome 20.0+ y Opera 10.0+ han implementado esta etiqueta.



23. Cursores.

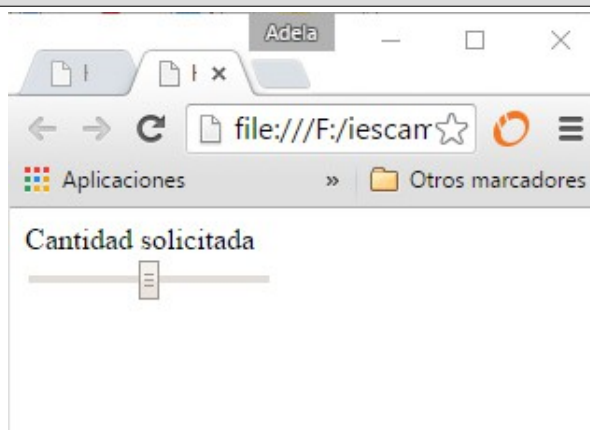
Los cursores son herramientas intuitivas de fácil manejo que permiten fijar un valor concreto en muchas aplicaciones de software, generalmente de procesamiento numérico. Están presentes, por ejemplo, en programas de procesamiento de texto u otros programas como Photoshop.

Los cursores se introducen en Html5 mediante la etiqueta `<input type="range">`. Apostamos a que este control tan original está llamado a tener éxito entre los diseñadores e invadirá las aplicaciones Web 2.0 en un futuro próximo.

Los cursores están operativos en numerosos navegadores. Como, por ejemplo, Internet Explorer 10+, Chrome 5.0+, Safari 3.1+, Opera 9.0+ e iOS Safari5.0+. Funcionan parcialmente en Android Browser 2.1+. Sólo Firefox no los implementa. Los demás navegadores los procesan como un campo de texto simple. Por tanto, no existe ningún motivo para no usarlos a día de hoy.

El código es:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Cantidad solicitada<br>
<input name="cursor" type="range">
</form>
</body>
</html>
```



Esta nueva etiqueta `<input type="range">` tiene atributos específicos, idénticos a los de `<input type="number">`.

min

Indica el valor mínimo del contador.

max

Indica el valor máximo del contador.

step

Determina el paso de avance del contador cada vez que se hace clic con el ratón en la flecha para aumentar o disminuir (véase la siguiente captura de pantalla).

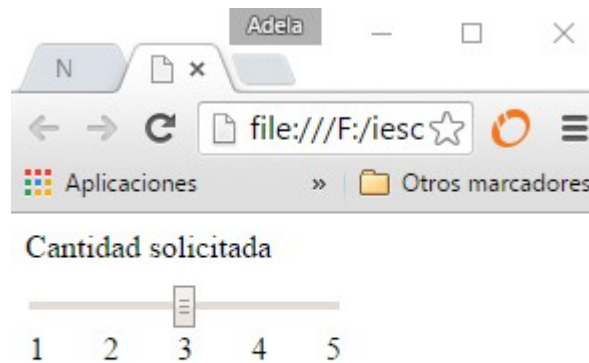
value

Determina el valor inicial del contador.

Usando algunas hojas de estilo, el siguiente ejemplo muestra un cursor que se posiciona en el valor inicial 3 (atributo `value`) donde los valores mínimo y máximo son 1 y 5 respectivamente.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
span {margin-right: 25px;}
</style>
</head>
<body>
<form action="">
Cantidad solicitada<br>
<input name="cursor" type="range" min="1" max="5" value="3"
step="1" style="margin-top: 10px; width: 155px;"><br>
<div style="margin-top: -2px">
<span style="margin-left: 2px;">1</span>
<span>2</span>
<span>3</span>
<span>4</span>
<span>5</span>
```

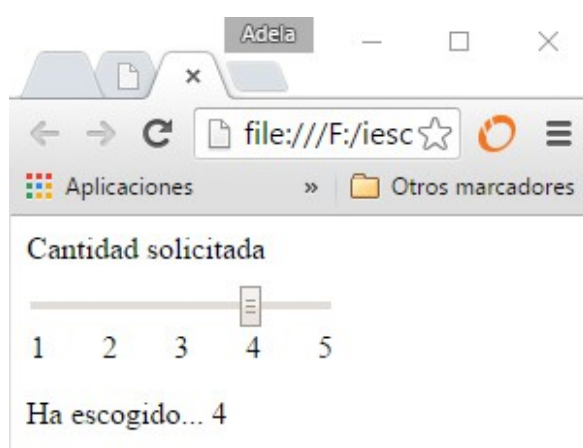
```
</div>
</form>
</body>
</html>
```



Para convencer al lector o confirmar al usuario la selección realizada, algunas líneas de JavaScript hacen aparecer el valor seleccionado.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
span {margin-right: 25px;}
</style>
<script type="text/javascript">
function mostrar(seleccion){
document.getElementById("valor").innerHTML = seleccion;
}
</script>
</head>
<body>
<form action="">
Cantidad solicitada<br>
<input name="cursor" type="range" min="1" max="5" value="3"
step="1" style="margin-top: 10px; width: 155px;"
onchange="mostrar(this.value)"><br>
<div style="margin-top: -2px">
<span style="margin-left: 2px;">1</span>
```

```
<span>2</span>
<span>3</span>
<span>4</span>
<span>5</span>
</div>
<p>Ha escogido... <span id="valor">3</span></p>
</form>
</body>
</html>
```



24. Una aplicación completa.

Elaboremos ahora un formulario completo para dar soporte a una actividad comercial. Los primeros campos del formulario que nos vienen a la mente procesarán los datos del cliente: nombre y apellidos, dirección, código postal y país (España exclusivamente). Los demás elementos servirán para procesar el pedido (modelo Road 66), la talla y la cantidad.

Los grupos Cliente y Pedido se pueden agrupar mediante la etiqueta `<fieldset>`.

El protocolo **mailto**, a pesar de sus límites, se utilizará para enviar un correo con los datos del formulario (por motivos prácticos de aprendizaje).

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
```

```
<meta charset="UTF-8">
<style>
fieldset { border : 1px solid black;}
</style>
</head>
<body>
<form action="mailto:su_direccion_email" method="post"
enctype="text/plain">
<fieldset>
<legend><i>Cliente</i></legend>
Nombre y apellidos: <input type="text" name="Nombre"><br>
Dirección: <input type="text" name="Dirección"><br>
Código Postal: <input type="text" name="CP" size="4"
maxlength="5"><br>
País: <input type="text" readonly name="País" value="España">
</fieldset>
<br>
<fieldset>
<legend><i>Pedido</i></legend>
Modelo: <input type="text" readonly name="Modelo" value="Road
66"><br>
Talla: <br>
<input type="radio" name="Talla" value="S">S<br>
<input type="radio" name="Talla" value="M">M<br>
<input type="radio" name="Talla" value="L">L<br>
<input type="radio" name="Talla" value="XL">XL<br>
Cantidad: <input type="text" name="Cantidad" value="1"><br>
</fieldset>
<br>
<input type="submit" value="Realizar pedido">
<input type="reset" value="Anular">
</form>
</body>
</html>
```

The screenshot shows a web browser window with the title 'Adela'. The address bar displays 'file:///F:/iescamp2'. The browser interface includes navigation buttons (back, forward, refresh) and a search bar. Below the browser window, there are two form sections:

Cliente

Nombre y apellidos:

Dirección:

Código Postal:

País:

Pedido

Modelo:

Talla:

- ☐ S
- ☐ M
- ☐ L
- ☐ XL

Cantidad:

At the bottom of the form, there are two buttons: 'Realizar pedido' and 'Anular'.

En el momento de la recepción, el correo electrónico puede tomar el siguiente aspecto:

Nombre=Andrés Espinar
Dirección=Calle de la Estación, 22 Madrid
CP=28000
País=España
Modelo=Road 66
Talla=L
Cantidad=2

Se trata de una lista de parejas con el nombre (name) que hayamos dado al campo de formulario seguido del signo igual = y los datos introducidos por el usuario.

Este formulario es muy similar a un formulario estándar de Html 4.0. Nos parece interesante agregar algunas etiquetas y atributos propios de Html5 para aprovechar algunas de las nuevas funcionalidades aportadas por Html5.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Html5</title>
<meta charset="UTF-8">
<style>
fieldset { border : 1px solid black;}
Span {margin-right: 24px;}
</style>
</head>
<body>
<form action="mailto:su_direccion_email" method="post"
enctype="text/plain">
<fieldset>
<legend><i>Cliente</i></legend>
Nombre y apellidos: <input type="text" name="Nombre" required
placeholder="Nombre y apellidos"><br>
Dirección: <input type="text" name="Dirección" required><br>
Código Postal: <input type="text" name="CP" size="4" maxlength="5"
required pattern="[0-9]"><br>
País: <input type="text" readonly name="País" value="España">
</fieldset>
<br>
<fieldset>
<legend><i>Pedido</i></legend>
Modelo: <input type="text" readonly name="Modelo" value="Road
66"><br>
Talla: <br>
<input type="range" name="Talla" min="1" max="4" step="1"
value="2">
<div>
<span>S</span>
<span>M</span>
<span>L</span>
<span>XL</span>
</div>
<br>
Cantidad: <input type="number" name="Cantidad" min="1" max="6"
step="1" value="1"><br>
</fieldset>
```

```
<br>
<input type="submit" value="Realizar pedido">
<input type="reset" value="Anular">
</form>
</body>
</html>
```

The screenshot shows a web browser window with the address bar displaying 'file:///F:/iescamp2'. The browser has tabs for 'Adela' and another unnamed tab. The page content is a form with two main sections: 'Cliente' and 'Pedido'.

Cliente Section:

- Nombre y apellidos:
- Dirección:
- Código Postal:
- País:

Pedido Section:

- Modelo:
- Talla: (with buttons S, M, L, XL below it)
- Cantidad:

At the bottom of the form are two buttons: 'Realizar pedido' and 'Anular'.

Este formulario, que incluye funcionalidades propias de Html5, es visualmente muy distinto al anterior, con la introducción del cursor para informar las tallas o del contador para la cantidad.

Los campos Nombre y apellidos y Dirección aparecen completados por una sugerencia sombreada (atributo `placeholder`).

Algunos campos se han complementado con el atributo `required`, que los vuelve obligatorios. En el momento de realizar el envío, aparece una etiqueta informativa indicando que es preciso completar este campo.

Hemos dotado al código postal de una expresión regular (atributo `pattern`) que exige una codificación exclusivamente numérica. Aparece una etiqueta informativa si el formato no es válido.

El contador, para introducir la cantidad, permite limitar una cantidad máxima, 6 unidades en este caso, mediante el atributo `max`.

He aquí las diferencias respecto a Html 4.0, que abren perspectivas prometedoras para las aplicaciones Html del futuro.