

Tema 4. Formularios

Tema 4. Formularios	1
1) Campos de formulario anteriores a html5 y sus novedades	3
<form>.....	4
<textarea>	4
<button>.....	5
<select>.....	6
<option>	7
<optgroup>	7
<fieldset>	7
<label>	8
2) Etiquetas nuevas de formulario en html5.....	8
<datalist>.....	8
Ejemplo	9
<keygen>	10
<output>	12
Ejemplo	12
<input>	13
<input type="text">y <input type="password">	14
<input type="checkbox"> y <input type="radio">	15
<input type="file">	15
<input type="submit">	16
<input type=image>	16
Nuevos tipos de controles.....	17
<date>	18
Ejemplo	18
< datetime >	20
Elemento de entrada datetime-local.....	20
Ejemplo	21
Elemento de entrada time	22
Ejemplo	22
Elemento de entrada week	23

Elemento de entrada month	23
Ejemplo	24
Elemento de entrada number	25
Elemento de entrada range	26
Ejemplo	27
Elemento de entrada tel.....	27
Elemento de entrada email	28
Ejemplo	29
Elemento de entrada url	29
Ejemplo	30
Elemento de entrada color	31
Ejemplo	31
Elemento de entrada search.....	33
Ejemplo	33
Ejercicios.....	35
Ejercicio 1: Crear un formulario completo con los nuevos controles de HTML5.....	35
Pasos a seguir	36

1) Campos de formulario anteriores a html5 y sus novedades

Las etiquetas que hemos venido utilizando hasta ahora relacionadas con los formularios y que todos conocemos son las siguientes:

<i>Etiqueta HTML</i>	<i>Definición</i>
<form>	Etiqueta que define un formulario de entrada para el usuario; en él pueden aparecer todos los elementos que tenemos a continuación.
<input>	Etiqueta que define un campo de entrada para el usuario (presenta muchas novedades en html5).
<textarea>	Etiqueta que define un control de texto de varias líneas.
<button>	Etiqueta que define un botón pulsable.
<select>	Etiqueta que define una lista desplegable.
<option>	Etiqueta que define las diferentes opciones dentro del elemento <i><select></i> .
<optgroup>	Etiqueta que define un grupo de elementos <i><option></i> para que la lista desplegable aparezca agrupada.
<fieldset>	Etiqueta que define un grupo de elementos relacionados dentro de un formulario.
<label>	Etiqueta que define una etiqueta para un elemento <i><input></i> .

Estas etiquetas se van a seguir utilizando porque son básicas. Además están disponibles en todos los navegadores más importantes.

Las novedades que podemos encontrar en HTML5, se encuentran en los **atributos** que podemos utilizar con estas etiquetas.

<form>

Es la etiqueta que define el formulario. Dentro de esta etiqueta aparecen todos los demás controles de formulario. Por tanto, los atributos que se pongan en el <form> afectan a todo su contenido.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

accept-charset: Atributo que especifica una lista de codificaciones de caracteres que el servidor acepta. Ej.: utf-8

action: Atributo que especifica la dirección del elemento al cual se envía el formulario, normalmente, un archivo conteniendo un script hecho especialmente para procesar esta información. Ej.: agente.php (un script en el lado del servidor).

- **Otra posibilidad sería:** action="mailto:midireccion@gmail.com"

enctype: Atributo que especifica de qué forma se deben codificar los datos cuando se presenten en servidor. Solo cuando el método sea *post*. Valores posibles:

- [application/x-www-form-urlencoded](#)
- [multipart/form-data](#)
- [text/plain](#)

method: Atributo que especifica el método HTTP mediante el cual se enviarán los datos del formulario.

Normalmente GET o POST.

"POST " (los datos del formulario son adjuntados al cuerpo del mismo) ó " GET " (los datos del formulario son adjuntados a la URL).

name: Atributo que especifica el nombre del elemento.

target: Destino en el que se recibe la respuesta del formulario.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario. *(Nuevo en HTML5)*

novalidate: Atributo que indica si se debe o no validar el formulario antes de enviarlo. *(Nuevo en HTML5)*

<textarea>

Esta etiqueta, define un entrada de texto de varias líneas

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

autofocus: Atributo que focaliza el elemento en el momento de cargar la página. *(Nuevo en HTML5)* . Es decir, recibe el foco (el cursor).

cols: Atributo que especifica el ancho del elemento.

disabled: Atributo que desactiva dicho elemento.

form: Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. *(Nuevo en HTML5)*

maxlength: Número máximo de caracteres que permitimos introducir en el elemento. *(Nuevo en HTML5)*

name: Atributo que especifica el nombre del elemento.

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado. *(Nuevo en HTML5)*

readonly: Atributo que convierte el elemento en un elemento de solo lectura.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario. *(Nuevo en HTML5)*

rows: Atributo que especifica el alto del elemento.

wrap: Atributo que define la forma en la que se envuelve el contenido al ser enviado, por ejemplo, si se respetan los saltos de línea o no. *(Nuevo en HTML5)*

<button>

Esta etiqueta, crea un botón que al ser pulsado por el usuario, en la mayoría de los casos invoca a una función. A partir de ahora, aparecen nuevos atributos que pueden sustituir a los que únicamente se colocaban dentro de la etiqueta `<form>`, y que definen por ejemplo el destino del formulario o la forma de envío de los datos.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

autofocus: Atributo que focaliza el elemento en el momento de cargar la página. *(Nuevo en HTML5)*

disabled: Atributo que desactiva dicho elemento.

form: Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. *(Nuevo en HTML5)*

formaction: Atributo que especifica la dirección del elemento al cual se envía el formulario. *(Nuevo en HTML5)*

formenctype: Atributo que especifica de qué forma se deben codificar los datos cuando se presenten en servidor. Solo cuando el método sea *post*. *(Nuevo en HTML5)*

formmethod: Atributo que especifica el método HTTP mediante el cual se enviarán los datos del formulario. *(Nuevo en HTML5)*

formnovalidate: Atributo que indica si se debe o no validar el formulario antes de enviarlo. *(Nuevo en HTML5)*

formtarget: Atributo que especifica un nombre o una palabra clave que indica el lugar para mostrar la respuesta que se recibe después de enviar el formulario. *(Nuevo en HTML5)*

name: Atributo que especifica el nombre del elemento.

type: Atributo que especifica el tipo de botón que vamos a crear.

value: Atributo que define el valor o contenido del elemento.

Ejemplo

Si utilizamos el siguiente código:

```
button.html x
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Nuevas opciones del elemento BUTTON</title>
5  </head>
6  <body>
7    <form action="comprueba.php" method="GET">
8      <label for="nom">Nombre:</label>
9      <input type="text" id="nom" placeholder="Introduce nombre">
10     <br><label for="ema">Email:</label>
11     <input type="text" id="ema" placeholder="Introduce email">
12     <br><input type="submit" value="Enviar">
13     <button formaction="comprueba.php" formmethod="POST">Enviar por POST</button>
14   </form>
15 </body>
16 </html>
17
```

El resultado en el navegador sería el siguiente:

Nombre:

Email:

Vemos cómo hemos creado un formulario de lo más normal, con dos campos de texto y un elemento `<INPUT>` de tipo *submit* para enviar el formulario al destino indicado. Pero aparte de estos elementos, hemos colocado también un botón dentro del formulario. Si nos fijamos, al enviar el formulario pulsando sobre el elemento `<INPUT>`, el destino será el fichero *comprueba.php* y el método de envío será GET, sin embargo, si pulsáramos sobre el botón, el destino sería el mismo pero el método en este caso POST. Por lo tanto, vemos como ahora, podemos definirle a un botón, atributos que antes solo estaban disponibles para la etiqueta `<FORM>`.

<select>

Esta etiqueta, define una lista desplegable.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

autofocus: Atributo que focaliza el elemento en el momento de cargar la página. (*Nuevo en HTML5*)

disabled: Atributo que desactiva dicho elemento.

form: Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. *(Nuevo en HTML5)*

multiple: Atributo que especifica que se pueda introducir más de un valor en el elemento.

name: Atributo que especifica el nombre del elemento.

size: Atributo que define el número de elementos visibles en la lista desplegable.

<option>

Esta etiqueta, define los elementos que serán parte de una lista desplegable.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

disabled: Atributo que desactiva dicho elemento.

label: Atributo que especifica una descripción corta del elemento.

selected: Atributo que especifica si el elemento está seleccionado por defecto o no.

value: Atributo que define el valor o contenido del elemento.

<optgroup>

Esta etiqueta, agrupa los elementos que serán parte de una lista desplegable en función de diferentes criterios.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

disabled: Atributo que desactiva dicho elemento.

label: Atributo que especifica una descripción corta del elemento.

<fieldset>

Esta etiqueta, agrupa los elementos de un formulario que estén relacionados entre sí. Parece un marco que rodea una serie de elementos relacionados.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

disabled: Atributo que desactiva dicho elemento. *(Nuevo en HTML5)*

form: Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. *(Nuevo en HTML5)*

name: Atributo que especifica el nombre del elemento. *(Nuevo en HTML5)*

<label>

Esta etiqueta, crea una etiqueta para uno o varios elementos de uno o varios formularios. Sirve para dar nombre.

Los atributos que podemos utilizar con esta etiqueta en HTML5 son los siguientes:

Atributos globales.

for: Atributo que contiene el identificador del elemento al que pertenece.

form: Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. *(Nuevo en HTML5)*

2) Etiquetas nuevas de formulario en html5

Etiqueta HTML	Definición
<datalist>	Etiqueta que especifica una lista de opciones predefinidas para los controles de entrada.
<keygen>	Etiqueta que define un campo generador de par de claves.
<output>	Etiqueta que define el resultado de un cálculo.

Vamos a ver ahora, más en profundidad, el funcionamiento de estas tres nuevas etiquetas.

<datalist>

Es un elemento que especifica una lista de opciones predefinidas para un elemento *<input>*. Cuando hacemos clic sobre el elemento de entrada, se despliega esta lista de opciones que hemos predefinido mediante esta etiqueta, y podemos seleccionar una de ellas. De la misma manera, cuando comenzamos a escribir en el elemento, solo aparecerán las opciones de dicha lista que coincidan con lo ya escrito. No tiene la flechita del desplegable de la select, sino un caja de texto con todas las opciones.

El elemento *<input>* y este elemento *<datalist>*, quedarán unidos mediante el atributo *list* disponible para el primero de ellos y que tendrá como valor el identificador del segundo.

Los atributos que podemos utilizar para esta etiqueta son:

Atributos globales.

NAVEGADOR	FUNCIONA
Chrome	

Safari	✗
Opera	✓
Firefox	✓
Internet Explorer	✗

Ejemplo

Si utilizamos el siguiente código:

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Etiqueta DATALIST</title>
5    </head>
6
7    <body>
8      <form>
9        <input type="text" list="listado">
10       <datalist id="listado">
11         <option value="Antena3"></option>
12         <option value="Tele5"></option>
13         <option value="LaSexta"></option>
14         <option value="Cuatro"></option>
15         <option value="TVE"></option>
16       </datalist>
17     </form>
18   </body>
19 </html>

```

El resultado en el navegador sería el siguiente:

	T
Antena3	Tele5
Tele5	TVE
LaSexta	
Cuatro	
TVE	

Vemos cómo en el momento que hacemos clic o nos colocamos sobre el elemento de entrada, se despliega la lista predefinida que hemos creado utilizando la etiqueta `<datalist>`. En el momento que empezamos a escribir, solo se mantienen en la lista los resultados que coinciden con lo que hemos escrito.

Si miramos el código, vemos como el atributo `list` de la etiqueta `<input>` hace referencia al identificador de la etiqueta `<datalist>`, de esta manera quedan unidos.

<keygen>

Es un elemento que especifica un campo generador de par de claves, privada y pública, dentro de un formulario. Cuando se envía el formulario, la clave privada se almacena localmente, y la clave pública se envía al servidor. Posteriormente, el servidor debería de firmar esta clave con su propio certificado y enviarla de vuelta al navegador. De esta manera, estaríamos estableciendo una seguridad para el cliente, y solo podríamos acceder al sitio teniendo dicho certificado. Es un proceso bastante complejo, que no vamos a tratar en este curso.

Cuando generamos un elemento de este tipo en un formulario, nos da opción de generar una clave de grado elevado de seguridad, o de grado medio.

Los atributos que podemos utilizar para esta etiqueta son:

Atributos globales.

autofocus: Especifica si este campo de formulario será focalizado automáticamente cuando cargue la página.

challenge: Especifica una cadena de desafío que se presenta junto con la clave pública. El valor predeterminado es una cadena vacía si no se especifica.

disabled: Especifica si el elemento está desactivado o no.

form: Especifica el identificador de uno o varios formularios a los que este elemento pertenece.

keytype: Especifica el algoritmo de seguridad de la clave. Puede ser RSA , DSA o EC . El valor por defecto es RSA.

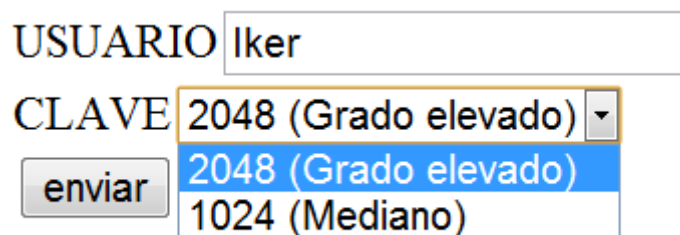
name: Especifica un nombre para este elemento.

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✓
Internet Explorer	✗

Si utilizamos el siguiente código:

```
keygen.html X
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Etiqueta KEYGEN</title>
5   </head>
6
7   <body>
8     <form action="verclave.php" method="post">
9
10      <label for="usuario">USUARIO</label>
11      <input type="text" id="usuario" name="usuario"><br>
12
13      <label for="clave">CLAVE</label>
14      <keygen id="clave" name="clave"></keygen><br>
15
16      <input type="submit" value="enviar">
17    </form>
18  </body>
19 </html>
```

El resultado en el navegador sería el siguiente:



USUARIO Iker

CLAVE 2048 (Grado elevado) ▼

enviar 2048 (Grado elevado)
1024 (Mediano)

Si enviamos el formulario, y un fichero PHP muestra los valores, veríamos esto:

USUARIO: Iker
CLAVE:
MIICQDCCASgwggEiMA0GCSqGSIb3DQEBAQ

Vemos como hemos incluido un elemento `<input>` de texto, un elemento de clave y un botón de envío dentro del formulario. Antes de ejecutar el envío, vemos como el elemento de clave nos da opción a elegir entre un grado elevado o un grado medio de seguridad para la clave que va a generar.

Cuando lo enviamos, el fichero `verclaclave.php` en servidor, recoge el contenido de los dos elementos y lo muestra en pantalla. La clave generada, es muchísimo más larga que lo que aparece en la imagen.

Como hemos dicho, la clave privada se queda almacenada en local, y esa clave pública que se envía al servidor deberá ser firmada y devuelta al navegador. Pero como hemos dicho, es un tema complejo que no vamos a tratar en este curso.

<output>

Es un elemento que representa el resultado de un cálculo, por ejemplo, uno creado mediante un script. A la vista, no tiene ninguna diferencia con una etiqueta *<label>* por ejemplo, pero es un elemento mucho más significativo para los motores de búsqueda o cualquier otro tipo de interfaz capaz de interpretar código HTML. Probablemente, sea el elemento más importante dentro del formulario, y aunque su resultado podamos presentarlo en cualquier elemento *<input>* no tiene la misma repercusión.

Los atributos que podemos utilizar para esta etiqueta son:

Atributos globales.

for: Especifica los identificadores de los elementos que tienen relación con el resultado de salida.

form: Especifica el identificador de uno o varios formularios a los que este elemento pertenece.

name: Especifica un nombre para este elemento.

Está disponible en los siguientes navegadores:

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✓
Internet Explorer	✗

Ejemplo

Si utilizamos el siguiente código:

```
output.html X
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Etiqueta OUTPUT</title>
5    </head>
6    <script>
7      function calcula(){
8        var precio = document.getElementById("precioproducto").value;
9        var cantidad = document.getElementById("cantidadproductos").value;
10       var salida = document.getElementById("resultado");
11       salida.value = precio * cantidad;
12     }
13   </script>
14
15   <body>
16     <form>
17       <label for="precioproducto">Precio del producto: </label>
18       <input type="text" id="precioproducto">
19
20       <br><label for="cantidadproductos">Nº de productos: </label>
21       <select id="cantidadproductos" onchange="calcula()">
22         <option>1</option>
23         <option>2</option>
24         <option>3</option>
25         <option>4</option>
26         <option>5</option>
27       </select>
28
29       <br><label for="resultado">Total: </label>
30       <output id="resultado">
31     </form>
32   </body>
33 </html>
```

El resultado en el navegador sería el siguiente:

Precio del producto:

Nº de productos: ▼

Total: 108

Vemos como hemos creado un formulario en el que tenemos un elemento `<input>`, que será donde introduzcamos el precio del producto, un elemento `<select>`, para seleccionar la cantidad de productos que queremos, y un último elemento `<output>`, que mostrara el precio total, es decir, el precio multiplicado por la cantidad.

Para realizar la operación, hemos creado una función de JavaScript que se ejecuta en el momento en el que cambia el elemento `<select>`. La función lo que hace es coger el valor del elemento `<input>`, es decir el precio, coger el valor del elemento `<select>`, es decir la cantidad, y multiplicarlos colocando ese valor en el elemento `<output>`. De esta manera, cada vez que cambiemos la cantidad de productos actualizara el precio final.

<input>

Uno de los elementos más utilizado y que más opciones presenta a la hora de trabajar con los formularios, es el elemento `<input>`. A través de su atributo *type*, ofrece un montón de controles diferentes. Los controles de los que disponíamos hasta ahora y que todos conocemos son los siguientes:

<i>Control de formulario</i>	<i>Definición</i>
<code><input type="text"></code>	Control que define un campo de entrada de texto.
<code><input type="password"></code>	Control que define un campo de entrada de contraseña, la diferencia con el control de texto, es que al escribir el contenido aparece oculto.
<code><input type="checkbox"></code>	Control que define un elemento de entrada de selección.
<code><input type="radio"></code>	Control que define un elemento de entrada de elección.
<code><input type="button"></code>	Control que define un botón que el usuario puede pulsar.
<code><input type="hidden"></code>	Control que define un campo de entrada oculto.
<code><input type="file"></code>	Control que define un campo de entrada para seleccionar un fichero del sistema de ficheros local.
<code><input type="submit"></code>	Control que define un botón de envío de formulario.
<code><input type="image"></code>	Control que define una imagen que será el botón de envío de formulario.
<code><input type="reset"></code>	Control que define un botón que resetea el formulario, y lo pone en las condiciones iniciales.

Todos estos controles ya los venimos utilizando desde hace tiempo, pero HTML5 presenta alguna novedad en alguno de ellos como veremos a continuación.

`<input type="text">` y `<input type="password">`

El primero de ellos es una caja de entrada texto y el segundo también lo es pero los caracteres que se escriben aparecen representados por puntos negros, para que no se vean al escribir.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

maxlength: Número máximo de caracteres que permitimos introducir en el elemento.

size: Atributo que define el tamaño que ocupará el elemento.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento. Solo en el tipo *text*. (Nuevo en HTML5)

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario. (Nuevo en HTML5)

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario. (Nuevo en HTML5)

pattern: Atributo que especifica una expresión regular que debe cumplir el elemento, se compara con el contenido antes de ser enviado el formulario. (Nuevo en HTML5)

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado. (Nuevo en HTML5)

`<input type="checkbox">` y `<input type="radio">`

Como ya sabemos, estos dos controles definen campos de selección y elección respectivamente. A partir de HTML5, el funcionamiento no varía mucho, pero aparecen nuevos atributos que podemos utilizar con ellos.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

checked: Atributo que especifica si dicho elemento estará seleccionado por defecto.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario. (Nuevo en HTML5)

`<input type="file">`

Este tipo de control, sirve para crear un campo que nos permita seleccionar un fichero desde nuestro sistema de ficheros local, para después subirlo al servidor o hacer con él lo que queramos. Una de las principales novedades en HTML5 para este control, es la posibilidad de seleccionar en un mismo elemento varios ficheros como veremos a continuación.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

accept: Atributo que especifica qué tipo de archivo es aceptado para este elemento.

multiple: Atributo que especifica que se pueda introducir más de un valor en el elemento. *(Nuevo en HTML5)*

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario. *(Nuevo en HTML5)*

<input type="submit">

Mediante este control, creamos un botón de envío del formulario. Al igual que hemos dicho respecto a la etiqueta <button>, en este caso también aparecen atributos que pueden sustituir a los que anteriormente solo se colocaban dentro del elemento <form> y que definían por ejemplo el destino del formulario o la forma de enviar los datos.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos <input>.

formaction: Atributo que especifica la dirección del elemento al cual se envía el formulario. *(Nuevo en HTML5)*

formenctype: Atributo que especifica de qué forma se deben codificar los datos cuando se presenten en servidor. Solo cuando el método sea *post*. *(Nuevo en HTML5)*

formmethod: Atributo que especifica el método HTTP mediante el cual se enviarán los datos del formulario. *(Nuevo en HTML5)*

formtarget: Atributo que especifica un nombre o una palabra clave que indica el lugar para mostrar la respuesta que se recibe después de enviar el formulario. *(Nuevo en HTML5)*

formnovalidate: Atributo que indica si se debe o no validar el formulario antes de enviarlo. *(Nuevo en HTML5)*

<input type="image">

Mediante este control, creamos una imagen que es un botón de envío del formulario. Al igual que hemos dicho respecto a la etiqueta <button> y al control *submit*, en este caso también aparecen atributos que pueden sustituir a los que anteriormente solo se colocaban dentro del elemento <form> y que definían por ejemplo el destino del formulario o la forma de enviar los datos.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos <input>.

alt: Atributo que define un texto alternativo a la imagen en caso de que no puede mostrarse.

src: Atributo que especifica la dirección de la imagen a mostrar.

formaction: Atributo que especifica la dirección del elemento al cual se envía el formulario. *(Nuevo en HTML5)*

formenctype: Atributo que especifica de qué forma se deben codificar los datos cuando se presenten en servidor. Solo cuando el método sea *post*. *(Nuevo en HTML5)*

formmethod: Atributo que especifica el método HTTP mediante el cual se enviarán los datos del formulario. *(Nuevo en HTML5)*

formtarget: Atributo que especifica un nombre o una palabra clave que indica el lugar para mostrar la respuesta que se recibe después de enviar el formulario. *(Nuevo en HTML5)*

height: Atributo que especifica la altura de la imagen. *(Nuevo en HTML5)*

width: Atributo que especifica la anchura de la imagen. *(Nuevo en HTML5)*

Nuevos tipos de controles

En HTML5, dada la importancia que tienen los formularios dentro de los espacios Web, aparecen nuevos controles mediante el uso de la etiqueta `<input>`, que vamos a ir viendo con detalle a continuación.

Control de formulario	Definición
<code><input type="date"></code>	Nuevo control que permite al usuario introducir una fecha mediante la selección de la misma en un calendario.
<code><input type="datetime"></code>	Nuevo control que permite al usuario introducir una fecha y una hora.
<code><input type="datetime-local"></code>	Nuevo control que permite al usuario introducir una fecha y una hora con la zona horaria local.
<code><input type="time"></code>	Nuevo control que permite al usuario introducir una hora.
<code><input type="week"></code>	Nuevo control que permite al usuario introducir una semana mediante la selección de la misma en un calendario.
<code><input type="month"></code>	Nuevo control que permite al usuario introducir un mes mediante la selección del mismo en un calendario.
<code><input type="number"></code>	Nuevo control que permite al usuario introducir un número desde una lista numérica.
<code><input type="range"></code>	Nuevo control que permite al usuario introducir un número desde una barra dentro de un rango.

<code><input type="tel"></code>	Nuevo control que permite al usuario introducir un número de teléfono.
<code><input type="email"></code>	Nuevo control que permite al usuario introducir una dirección de correo electrónico.
<code><input type="url"></code>	Nuevo control que permite al usuario introducir una dirección Web.
<code><input type="color"></code>	Nuevo control que permite al usuario seleccionar un color desde una paleta de colores.
<code><input type="search"></code>	Nuevo control que permite al usuario introducir un contenido a buscar, su comportamiento es similar al de un campo de texto.

<date>

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir una fecha mediante la selección de la misma en un calendario.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

Ejemplo

Si utilizamos el siguiente código:

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control DATE</title>
5 </head>
6 <body>
7   <form>
8     <label for="elem">Selecciona fecha:</label>
9     <input type="date" id="elem" max="2016-04-14" min="2002-08-19">
10  </form>
11 </body>
12 </html>
```

El resultado en el navegador sería el siguiente:

Selecciona fecha: 19/08/2002 ▼

<< < agosto 2002 ▼ > >>

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	31	1
2	3	4	5	6	7	8

Hoy Eliminar

Selecciona fecha: 14/04/2016 ▼

<< < abril 2016 ▼ > >>

lun	mar	mié	jue	vie	sáb	dom
28	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

Hoy Eliminar

Vemos como hemos creado un campo de entrada de tipo *date*, y una etiqueta para dicho control. Al hacer clic sobre el elemento, automáticamente se despliega un calendario para seleccionar la fecha elegida. Como se ve en las imágenes, hemos definido una fecha máxima para el 14/04/2016 y una fecha mínima para el 19/08/2002, y por tanto si nos fijamos, esos son los límites de fechas que nos permite seleccionar el calendario.

< datetime >

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir una fecha y una hora (UTC) mediante la selección de la misma en un calendario y en un campo de selección de hora.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos <input>.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento <datalist> que predefine una lista para dicho elemento.

NAVEGADOR	FUNCIONA
Chrome	✗
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

Elemento de entrada datetime-local

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir una fecha y una hora local mediante la selección de la misma en un calendario y en un campo de selección de hora.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos <input>.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

Ejemplo

Si utilizamos el siguiente código:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control DATETIME</title>
5 </head>
6 <body>
7   <form>
8     <label for="elem">Selecciona fecha y hora UTC:</label>
9     <input type="datetime" id="elem" />
10    <br><br><label for="elem2">Selecciona fecha y hora local:</label>
11    <input type="datetime-local" id="elem2" />
12  </form>
13 </body>
14 </html>
```

NAVEGADOR	FUNCIONA
Chrome	✗
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

El resultado en el navegador sería el siguiente:

Selecciona fecha y hora: 2012-07-06 12:00 UTC

Selecciona fecha y hora local: 2012-07-14 12:00

Julio

2012

Lun	Mar	Mie	Jue	Vie	Sab	Dom
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Hoy

Vemos como hemos creado en primer lugar, un campo de entrada de tipo *datetime*, y una etiqueta para dicho control, y en segundo lugar, un campo de entrada de tipo *datetime-local*, y otra etiqueta para dicho control. Vemos que son similares, en ambos aparece un campo para la fecha que al hacer clic despliega el calendario, y un campo para la hora que nos permite aumentar o disminuir los minutos. La diferencia aparece en que el primero es la hora UTC, y la segunda la hora local, importante a efectos de cambios de hora y países.

Elemento de entrada time

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir un elemento de hora.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos *<input>*.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento *<datalist>* que predefine una lista para dicho elemento.

Ejemplo

Si utilizamos el siguiente código:

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control TIME</title>
5 </head>
6 <body>
7   <form>
8     <label for="elem">Selecciona una hora:</label>
9     <input type="time" id="elem" max="02:00:00" min="00:00:00" step="10">
10  </form>
11 </body>
12 </html>
```

El resultado en el navegador sería el siguiente:

Selecciona una hora:

Vemos como hemos creado un campo de entrada de tipo *time*, y una etiqueta para dicho control. Junto al campo de entrada aparecen dos botones para aumentar y disminuir la cantidad de tiempo. Hemos definido que la hora máxima sean las 02:00:00 y la mínima las 00:00:00, por tanto no nos dejará salirnos de dichos márgenes. Por otro lado, hemos definido un salto de 10 segundos, por lo tanto, cada vez que pulsemos los botones se aumentará o disminuirá en 10 y no en 1.

Elemento de entrada week

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir una semana del año desde un calendario.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos *<input>*.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento *<datalist>* que predefine una lista para dicho elemento.

NAVEGADOR	FUNCIONA
Chrome	
Safari	
Opera	
Firefox	
Internet Explorer	

Elemento de entrada month

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir un mes del año desde un calendario.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos *<input>*.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

NAVEGADOR	FUNCIONA
Chrome	
Safari	
Opera	
Firefox	
Internet Explorer	

Ejemplo

Si utilizamos el siguiente código:

```
weekmonth.html X
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Control WEEK/MONTH</title>
5  </head>
6  <body>
7      <form>
8          <label for="elem">Selecciona semana:</label>
9          <input type="week" id="elem" min="2012-W20">
10         <br><br><label for="elem2">Selecciona mes:</label>
11         <input type="month" id="elem2" max="2012-10" step="2">
12     </form>
13 </body>
14 </html>
```

El resultado en el navegador sería el siguiente:

Selecciona semana: 2012-W20 ▾

Selecciona mes:

Mayo

2012

Sem	Lun	Mar	Mie	Jue	Vie	Sab	Dom
18	30	1	2	3	4	5	6
19	7	8	9	10	11	12	13
20	14	15	16	17	18	19	20
21	21	22	23	24	25	26	27
22	28	29	30	31	1	2	3
23	4	5	6	7	8	9	10

Hoy

Selecciona semana: 2012-W20 ▾

Selecciona mes: 2012-08 ▾

Agosto

2012

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

Hoy

Vemos como hemos creado en primer lugar, un campo de entrada de tipo *week*, y una etiqueta para dicho control, y en segundo lugar, un campo de entrada de tipo *month*, y otra etiqueta para dicho control. Vemos que son similares, en ambos, al hacer clic sobre el elemento se despliega el calendario, pero en el primero de ellos podemos seleccionar solo semanas y en el segundo sin embargo meses enteros.

El formato del valor que se pone en el campo de entrada también es diferente, en el primero de ellos, 2012-W20 hace referencia a la semana número 20 de 2012, y en el segundo 2012-08 hace referencia al mes 8 de 2012. Si nos fijamos en los atributos que hemos puesto, en el caso de la semana, no nos va a dejar seleccionar una semana inferior a la 20, y en el caso de los meses, nos dejará seleccionar únicamente hasta el mes número 10, pero además, solo podremos escoger un mes sí y otro no, ya que el salto que hemos definido es de 2.

Elemento de entrada number

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario para introducir un número. Aparecen al lado, unos botones para aumentar o disminuir el número seleccionado.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

Elemento de entrada range

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario para seleccionar un valor de manera gráfica desde una barra de desplazamiento.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

min: Atributo que especifica el valor mínimo que podemos introducir.

max: Atributo que especifica el valor máximo que podemos introducir.

step: Atributo que define el salto entre los valores permitidos.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✓
Firefox	✗
Internet Explorer	✗

Ejemplo

Si utilizamos el siguiente código:

```
numberrange.html x
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Control NUMBER/RANGE</title>
5  </head>
6  <body>
7      <form>
8          <label for="elem">Selecciona un número:</label>
9          <input type="number" id="elem" min="5" max="21">
10         <br><br><label for="elem2">Selecciona un número del rango:</label>
11         <input type="range" id="elem2" min="0" max="10" step="2">
12     </form>
13 </body>
14 </html>
```

El resultado en el navegador sería el siguiente:

Selecciona un número:

Selecciona un número del rango:

Vemos como hemos creado en primer lugar, un campo de entrada de tipo *number*, y una etiqueta para dicho control, y en segundo lugar, un campo de entrada de tipo *range*, y otra etiqueta para dicho control. Vemos que en el primero de ellos, aparecen unas flechas para aumentar o disminuir el número que aparece en el cuadro de entrada, como hemos limitado el valor entre 5 y 21, no podremos elegir ningún valor fuera de estos límites. En segundo lugar, aparece una barra que el usuario puede mover para seleccionar un valor de la misma, y como hemos limitado los valores de 0 a 10 y con un salto de 2, únicamente podemos elegir entre esos 6 valores disponibles.

Elemento de entrada tel

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario para introducir un número de teléfono. De momento este control no está en funcionamiento para los principales navegadores. Es de gran importancia en los dispositivos móviles.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos *<input>*.

maxlength: Número máximo de caracteres que permitimos introducir en el elemento.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

pattern: Atributo que especifica una expresión regular que debe cumplir el elemento, se compara con el contenido antes de ser enviado el formulario.

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

size: Atributo que define el tamaño que ocupará el elemento.

No está en funcionamiento en ningún navegador

Elemento de entrada email

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir un correo electrónico. Solo se dará por bueno el contenido que tenga la estructura que define un correo electrónico, hasta entonces no se enviará el formulario.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

maxlength: Número máximo de caracteres que permitimos introducir en el elemento.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

multiple: Atributo que especifica que se puedan introducir más de un valor en el elemento.

pattern: Atributo que especifica una expresión regular que debe cumplir el elemento, se compara con el contenido antes de ser enviado el formulario.

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

size: Atributo que define el tamaño que ocupará el elemento.

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✗
Opera	✓
Firefox	✓
Internet Explorer	✗

Ejemplo


Si utilizamos el siguiente código:

```
email.html X
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control EMAIL</title>
5 </head>
6 <body>
7   <form action="#">
8     <label for="elem">Introduce un email:</label>
9     <input type="email" id="elem" placeholder="Introduce email" multiple>
10    <input type="submit" value="enviar">
11  </form>
12 </body>
13 </html>
```

El resultado en el navegador sería el siguiente:

Introduce un email:

Introduce un email:

 Introduce una lista de direcciones de correo electrónico separada por comas

Vemos como hemos creado un campo de entrada de tipo *email*, una etiqueta para dicho control, y un botón de envío de formulario. Mediante el atributo *placeholder*, definimos el texto que debe de aparecer en el elemento de entrada mientras no se escriba nada. En el momento que escribimos un contenido que no se corresponde con una dirección de correo electrónico y pulsamos en enviar, vemos que nos aparece un mensaje de error, y el formulario no se envía. En este caso, como hemos puesto el atributo *multiple* en el elemento de correo, en el mensaje de error nos deja constancia de ello. Cuando el contenido sea correcto, se enviará el formulario.

Elemento de entrada url

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario para introducir una dirección Web. Solo se dará por bueno, el contenido que tenga la estructura que define una página Web, hasta entonces no se enviará el formulario.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

maxlength: Número máximo de caracteres que permitimos introducir en el elemento.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

pattern: Atributo que especifica una expresión regular que debe cumplir el elemento, se compara con el contenido antes de ser enviado el formulario.

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

size: Atributo que define el tamaño que ocupará el elemento.

Ejemplo

Si utilizamos el siguiente código:

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✗
Opera	✓
Firefox	✓
Internet Explorer	✗

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control URL</title>
5 </head>
6 <body>
7   <form action="#">
8     <label for="elem">Introduce una dirección Web:</label>
9     <input type="url" id="elem" required>
10    <input type="submit" value="enviar">
11  </form>
12 </body>
13 </html>
```

El resultado en el navegador sería el siguiente:

Introduce una dirección Web:

! Completa este campo

Introduce una dirección Web:

! Introduce una URL

Vemos como hemos creado un campo de entrada de tipo *url*, una etiqueta para dicho control, y un botón de envío de formulario. Mediante el atributo *required*, no permitimos que este campo sea enviado totalmente vacío, por lo tanto, al enviar el formulario con dicho campo sin rellenar, nos salta un error como vemos en la imagen de arriba. En el momento que escribimos un contenido que no se corresponde con una dirección Web y pulsamos en enviar, vemos que nos aparece un mensaje de error diferente al anterior, pero el formulario tampoco se envía. Cuando el contenido sea correcto, se enviará el formulario.

Elemento de entrada color

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario y seleccionar un color desde la paleta de colores.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

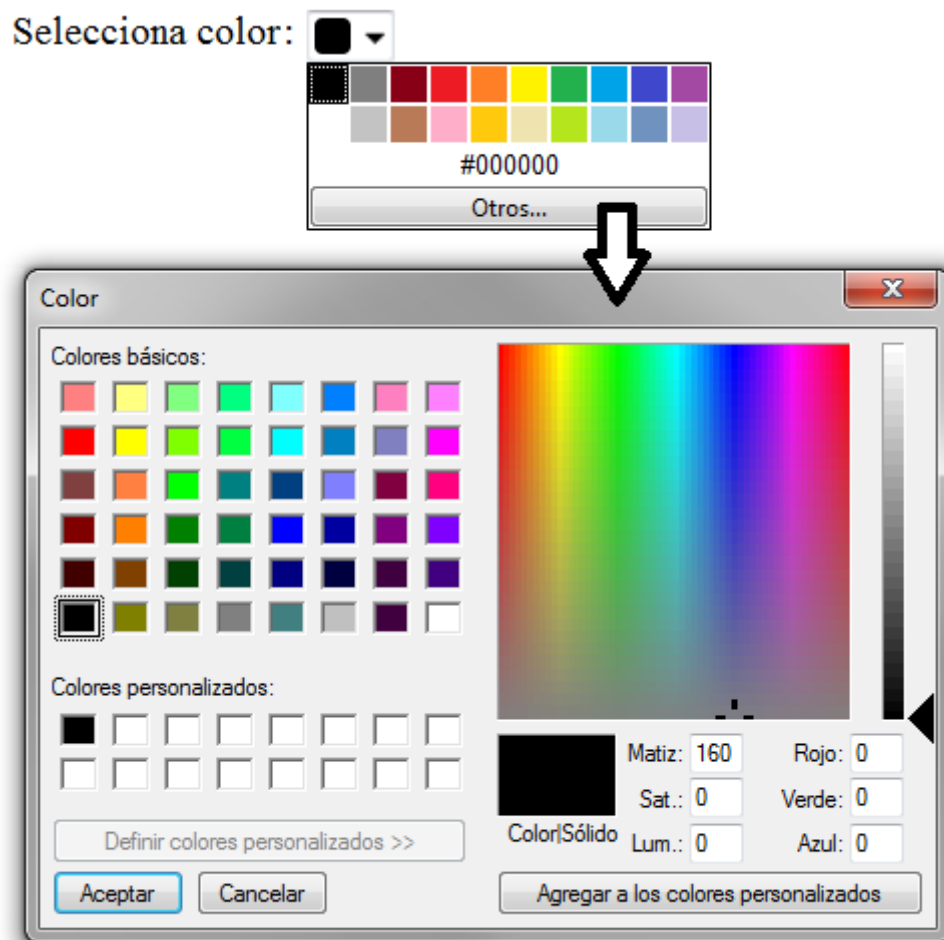
NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✗
Opera	✓
Firefox	✗
Internet Explorer	✗

Ejemplo

Si utilizamos el siguiente código:

```
color.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control COLOR</title>
5 </head>
6 <body>
7   <form>
8     <label for="elem">Selecciona color:</label>
9     <input type="color" id="elem">
10  </form>
11 </body>
12 </html>
```

El resultado en el navegador sería el siguiente:



Vemos como hemos creado un campo de entrada de tipo *color* y una etiqueta para dicho control. Vemos que este tipo de control muestra un elemento de lista para poder seleccionar un color desde la paleta de colores. El valor que enviará el formulario, será el color seleccionado en código hexadecimal como se muestra en la imagen.

Elemento de entrada search

Este nuevo control que aparece en HTML5, permite crear un campo de entrada para el usuario e introducir un contenido a buscar. Visualmente, es un tipo de control muy similar a un campo de texto, pero su importancia reside en el significado semántico del mismo.

Aparte del atributo *type* que define el tipo de control, los atributos que podemos utilizar para este tipo de control son:

Atributos globales.

Atributos comunes de todos los elementos `<input>`.

maxlength: Número máximo de caracteres que permitimos introducir en el elemento.

autocomplete: Atributo que activa o no la función de autocompletado en función de los valores introducidos con anterioridad por el usuario.

pattern: Atributo que especifica una expresión regular que debe cumplir el elemento, se compara con el contenido antes de ser enviado el formulario.

placeholder: Atributo que especifica una descripción del elemento, y que aparece cuando el campo está vacío o cuando no está focalizado.

required: Atributo que especifica que es un elemento requerido y que debe de estar completado antes de enviar el formulario.

list: Atributo que contiene el identificador del elemento `<datalist>` que predefine una lista para dicho elemento.

size: Atributo que define el tamaño que ocupará el elemento.

NAVEGADOR	FUNCIONA
Chrome	✓
Safari	✓
Opera	✗
Firefox	✗
Internet Explorer	✗

Ejemplo

Si utilizamos el siguiente código:

```
search.html X
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Control SEARCH</title>
5 </head>
6 <body>
7   <form>
8     <label for="elem">Introduce el contenido a buscar:</label>
9     <input type="search" id="elem" placeholder="buscar" pattern="[a-z]">
10   </form>
11 </body>
12 </html>
```

El resultado en el navegador sería el siguiente:

Introduce el contenido a buscar:

Introduce el contenido a buscar:

! Utiliza un formato que coincida con el solicitado

Vemos como hemos creado un campo de entrada de tipo *search* y una etiqueta para dicho control. Visualmente, es un campo de texto normal, pero en el momento que escribimos algo en él nos aparece la opción de eliminarlo. Hemos utilizado el atributo *placeholder* para definir el texto que aparece cuando no se ha escrito nada, y mediante el atributo *pattern*, le indicamos una expresión regular que debe de cumplir el contenido introducido. En este caso, la expresión que hemos puesto hace referencia a una sola letra, por lo tanto al introducir un contenido que no se corresponde nos salta el error.

Atributos comunes de todos los <input>	Definición
autofocus	Atributo que focaliza el elemento en el momento de cargar la página.
form	Atributo que contiene los identificadores de los formularios a los que pertenece dicho elemento. <i>(Nuevo en HTML5)</i>
Name	Atributo que especifica el nombre del elemento.
value	Atributo que define el valor o contenido del elemento.
disabled	Atributo que desactiva dicho elemento.
readonly	Atributo que convierte al elemento en un elemento de solo lectura.

<i>Atributos globales</i>	<i>Definición</i>
accesskey	Para crear un atajo de teclado y acceder al elemento con él.
class	Clase a la que pertenece el comando.
contenteditable	Para hacer que el contenido sea editable desde el propio navegador.
contextmenu	Contiene el nombre del menú que será menú contextual del elemento.
draggable	Para permitir que el elemento sea arrastrable.
dropzone	Para convertir el elemento en zona donde se puede arrastrar otro elemento.
Hidden	Para convertir el elemento en oculto.
Id	Definir el identificador para el elemento.
Lang	Especifica el idioma principal del contenido del elemento.
Spellcheck	Para permitir la corrección automática, se usa cuando el elemento es editable o cuando se puede escribir en él.
style	Atributo para definir el estilo del elemento.
tabindex	Para controlar cuándo se focaliza el elemento. El nombre viene de la tecla tabulador. Muy común en los formularios.
title	Define el título del elemento.
translate	Para definir si se traduce o no el elemento cuando se traduce la página desde el navegador.

Ejercicios

Ejercicio 1: Crear un formulario completo con los nuevos controles de HTML5

Vamos a crear un formulario completo que podría ser válido para el registro en cualquier página Web. Vamos a utilizar para ello, tanto los elementos de formulario que ya existían, como los nuevos que aparecen a partir de HTML5. Lo que tenemos que conseguir es más o menos lo que aparece en la siguiente imagen.

FORMULARIO DE REGISTRO	
Nombre y apellidos	
Nombre y apellidos*:	<input type="text" value="Introduce nombre y apellidos"/>
Nombre de usuario*:	<input type="text" value="Nombre de usuario"/>
Password*:	<input type="password" value="Introduce password"/>
Dirección	
Tipo de Vía*:	<input type="radio"/> Calle <input type="radio"/> Avenida <input type="radio"/> Paseo
Nombre de la vía*:	<input type="text"/>
Número:	<input type="text"/>
Código Postal*:	<input type="text"/>
Municipio:	<input type="text"/>
Provincia:	La Rioja ▼
Datos personales	
DNI*:	<input type="text"/>
Fecha de Nacimiento*:	<input type="text" value="dd/mm/aaaa"/>
Email*:	<input type="text"/>
Espacio Web:	<input type="text"/>
Teléfono:	<input type="text"/>
Fotos:	<input type="button" value="Elegir archivos"/> Ningún archivo seleccionado
Intereses	
Seleccione sus preferencias:	<input type="checkbox"/> Deportes <input type="checkbox"/> Música <input type="checkbox"/> Cine <input type="checkbox"/> Informática
Color preferido:	<input type="color" value="black"/>
<div>POST!</div> <div>Enviar por GET sin validar</div>	

Pasos a seguir

1. Todos los campos con un * son campos obligatorios.
2. Cada elemento debe de ser de un tipo acorde con el dato que vamos a introducir en él.
3. El elemento para introducir el número no puede ser ni cero ni negativo.
4. El código postal debe de estar compuesto por 5 dígitos numéricos.
5. El DNI deberá estar compuesto por 8 dígitos numéricos y una letra mayúscula.
6. La imagen, envía el formulario validado y mediante el método POST.
7. El botón, envía el formulario mediante el método GET y sin validación alguna.