

UF1304: ELABORACIÓN DE PLANTILLAS Y FORMULARIOS.

Manual teórico.

INDICE

1.	Formularios en la construcción de páginas web.....	3
1.1.	Introducción.	3
1.2.	Elementos y atributos.....	5
1.2.1.	El formulario.	6
1.2.2.	Elementos de entrada.....	6
1.2.3.	Botones.....	14
1.2.4.	Listas desplegables	16
1.2.5.	Etiquetas.....	18
1.2.6.	Grupos de elementos.	20
1.3.	Eventos.....	21
1.4.	Creación de formularios.	22
1.5.	Elementos de formulario adicionales del HTML5.	24
1.6.	Accesibilidad y Usabilidad en los formularios.....	32
1.6.1.	Criterios de accesibilidad	33
2.	Plantillas en la construcción de páginas web.	33
2.1.	Introducción.	33
2.2.	Repositorios libres de plantillas.....	34

1. Formularios en la construcción de páginas web.

1.1. Introducción.

Los formularios interactivos permiten a los autores de páginas Web poner elementos interactivos en sus páginas, por ejemplo, para recibir mensajes de sus lectores, de forma similar a las cartas de respuestas que se encuentra en algunas revistas.

El lector escribe la información rellenando campos o haciendo clic sobre botones, y luego presiona un botón de envío para enviarla a una dirección URL que se suele dirigir a una dirección de correo electrónico o a un script dinámico Web como PHP, ASP o CGI, que reciben los datos, los almacenan en una base de datos y los procesan.

Aunque los formularios se utilizan para interactuar con el servidor, sin duda alguna, su uso más extendido es el de recopilar información de los usuarios (por ejemplo: datos personales, aficiones, ...).

A continuación un ejemplo de un formulario, maquetado por medio de una tabla:

```
<form action="mailto:direccion@correo.com" method="post"
      enctype="text/plain">

<table border="0">
<tr height="50">
  <td>Nombre</td>
  <td>
<input name="nombre" type="Text" size="18" maxlength="18">
  </td>
</tr>
<tr height="50">
  <td>apellidos</td>
  <td>
<input type="text" name="apellidos" size="48" maxlength=48">
  </td>
</tr>
<tr height="50">
  <td>f. nacimiento </td>
  <td>
<input name="f_n" type="text" size="18" maxlength="18">
  </td>
</tr>
<tr height="50">
  <td>calle y número</td>
  <td>
```

[illegible]

```
</tr>
</table>
</form>
```

The diagram illustrates a web form with the following fields and their corresponding input types:

- Nombre**: Input type text
- Apellidos**: Input type text
- F. Nacimiento**: Input type text
- Calle y Número**: Input type text
- Código postal**: Input type text
- Provincia**: Input type text
- Teléfono**: Input type text
- Sexo**: Input type radio (Hombre, Mujer)
- Estado Civil**: select lista/menú (Soltero)
- Idiomas**: Input type checkbox (Español, Francés, Inglés, Alemán)
- Comentarios Personales**: Input type textarea
- enviar los datos**: Input type submit envío de datos
- borrar los datos**: Input type reset restaurar datos

1.2. Elementos y atributos.

Un formulario se estructura en cuatro secciones:

- **Inicio del formulario:** Los formularios están delimitados con la etiqueta `<form> ... </form>`, que permite reunir varios elementos de formulario, como botones y casillas de texto y es donde se indica cómo se envía la información y su destino.
- **Elementos del formulario:** Es la parte más importante del formulario y es dónde se indican los datos que se quieren registrar. Los elementos de

un formulario se indican con las etiquetas **<input>**, **<textarea>** o **<select>**.

- **Botones de acción:** Permiten enviar o resetear el formulario. Los botones se pueden definir con las etiquetas **<input>** o **<button>**.
- **Fin del formulario:** El fin del formulario se establece con la etiqueta **</form>**.

1.2.1. El formulario.

Los formularios deben poseer los siguientes atributos:

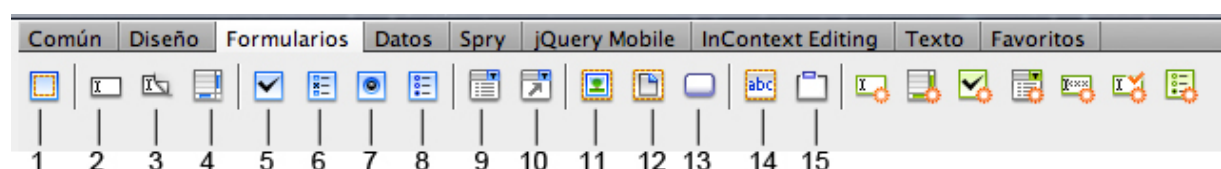
- **Name:** Es el nombre del formulario y es muy importante indicarlo en el caso de utilizar más de un formulario en la misma página o si se utilizan lenguajes scripts para verificar los datos.
- **Method:** Indica cómo se enviarán las respuestas
 - "GET" envía los datos agregándolos a la dirección URL y separándolos de la dirección con un signo de interrogación. Por ejemplo, en la URL `http://estaesmiweb.es/formulario.html?nombre=Luis` se envía a variable nombre con el valor "Luis".
 - "POST" Esta opción es más segura que la anterior, ya que los datos se envían directamente al servidor y no por la URL.
- **Action:** Indica la dirección a la que se enviará la información del formulario (un script CGI o dirección de correo electrónico (`mailto:dirección_de_correo_e@equipo`))
- **Enctype:** Es un parámetro opcional aunque recomendable que especifica cómo se codifican los datos del formulario. De cualquier forma, esto no necesita especificarse, ya que el valor predeterminado (**application/x-www-form-urlencoded**) codifica los caracteres antes de enviar el formulario, (**multipart/form-data**), se utiliza para enviar fichero sin realizar ningún tipo de codificación y (**text/plain**) envía la información en texto plano.

1.2.2. Elementos de entrada.

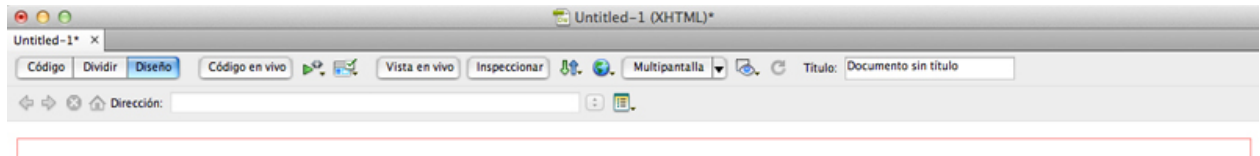
La etiqueta INPUT es una etiqueta esencial para los formularios, ya que se usa para crear muchos elementos "interactivos". La sintaxis de esta etiqueta es la siguiente:

```
<INPUT type="Nombre de campo" value="Valor predeterminado"
name="Nombre de elemento">
```

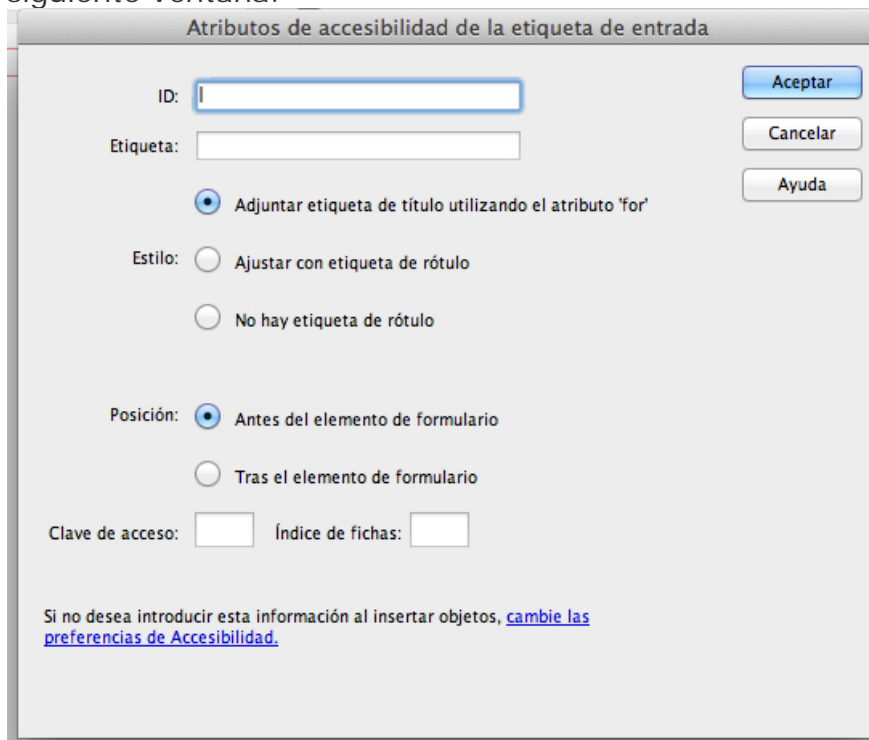
- El atributo **name** es esencial, ya que permite al script CGI reconocer qué campo está asociado con un par nombre/valor, lo que significa que el nombre del campo estará seguido de un signo igual ("=") seguido de un valor que el usuario introdujo, o si el usuario no introdujo ningún valor, por el valor predeterminado de la etiqueta value.
- El atributo **type** se usa para especificar qué tipo de elemento se representa con la etiqueta INPUT. Estos son los valores posibles:
 - **checkbox**: Las casillas de elección pueden adoptar uno de dos estados: **checked** (seleccionado) o **unchecked** (no seleccionado). Cuando la casilla es seleccionada, el par nombre/valor se envía al CGI.
 - **hidden**: Este campo, que el navegador no muestra, es para definir una configuración única que se enviará al CGI como par nombre/valor.
 - **file**: Un campo que permite al usuario especificar una ruta de archivo que lleva al archivo que se enviará con el formulario. Los tipos de archivo que pueden ser enviados deben especificarse utilizando el atributo ACCEPT de la etiqueta FORM.
 - **image**: Un botón de envío personalizado que aparece cuando se ubica una imagen en la ubicación definida por el atributo SRC.
 - **password**: Una casilla de texto donde los caracteres escritos aparecen como asteriscos para camuflar el texto de entrada.
 - **radio**: Un botón que permite al usuario elegir entre varias opciones. Cada uno de estos botones debe tener el mismo atributo name. El par nombre/valor del botón radio seleccionado se enviará al CGI. Al aplicar el atributo checked para uno de estos botones se definirá como seleccionado de forma predeterminada.
 - **reset**: Un botón de restauración para quitar todos los elementos en el formulario y restablecer sus valores predeterminados.
 - **submit**: Un botón de envío para enviar el formulario. El texto en el botón puede definirse usando el atributo value.
 - **text**: Una casilla de texto para escribir una línea de texto. El tamaño de la casilla puede definirse usando el atributo **size** y la extensión máxima del texto con el atributo **maxlength**.
- Dreamweaver nos da la posibilidad de crear un formulario de contacto sencillo y fácil de programar son tener demasiados conocimientos de html y php.
- Para insertar el área de formulario debemos dirigirnos a la barra de solapas superior y clicar en la de "Formularios".
En esta imagen se muestra los nombres y descripción de las opciones dentro de la barra de iconos de "Formularios":



1) FORMULARIO: Sirve para definir el área que va a ocupar el formulario dentro del html. Primero debemos posicionarnos dentro del html donde queremos insertar el formulario y luego hacer click en este icono. Dentro de esas “líneas rojas” vamos a colocar los elementos de nuestro formulario. Una vez insertada el área de formulario debería verse más o menos así:



2) CAMPO DE TEXTO: Define un campo de datos para que el usuario ingrese información, por ejemplo: Nombre y Apellido. Al hacer clic en el icono se abrirá la siguiente ventana:



ID: es el nombre que le podremos a este campo de datos. Este nombre no será visible para el usuario, solo será visible en el modo de visualización de código y como etiqueta. Debemos ingresar el ID (identificación) del campo con letras minúsculas, sin espacios ni caracteres espaciales; esto luego nos será útil para definir el código de validación del formulario.

Etiqueta: el nombre del campo que será visible para el usuario, por ejemplo: Nombre y Apellido

Posición: debemos elegir si deseamos que el nombre del campo (el que es visible para el usuario) queremos que esté antes del cuadro de texto o detrás. Las demás opciones la dejaremos tal cual están por el momento.

Una vez completemos el ID y la ETIQUETA el campo de texto insertado se verá así:

Nombre y Apellido:

Si por algún motivo deseamos cambiar los atributos del campo de texto lo podremos hacer desde la barra de propiedades, ubicada en la parte inferior de nuestra pantalla.

3) AREA DE TEXTO: Sirve para insertar un área de texto mayor al “campo de texto”. Generalmente se usa para definir el lugar donde el usuario puede dejar un mensaje o comentario. Esta área de texto puede tener varias líneas (renglones) y usar una barra de scroll:

Mensaje:

Al insertar un AREA DE TEXTO la barra de propiedades se modificará. A continuación veremos las propiedades que necesitamos modificar según la necesidad del formulario:



“Ancho car”: Ancho caracteres, los caracteres son letras, números, comas, etc. En este cuadro pondremos la cantidad de caracteres de ancho que deseamos que tenga el área de texto. Si quiero que en un línea de texto entren hasta 20 caracteres, entonces introduciremos ese valor ahí.

“Líneas num”: Cantidad de líneas que deseamos que el usuario pueda tipear. Si dejamos ese valor vacío el usuario será capaz de escribir sin restricciones. Lo siguiente que nos importará modificar en esta instancia es alguna de las opciones de “Tipo”, tenemos 3 opciones Una línea, Varias líneas y Contraseña. Si estamos creando un área de texto para un mensaje nos convendrá seleccionar la opción “Varias líneas”. La opción “contraseña” puede ser usada para cuando

nuestro sitio tiene la posibilidad de dar al usuario un nombre de usuario y contraseña.

4) CASILLA DE VERIFICACION: Son típicamente usadas para darle al usuario a elegir entre una o varias opciones preestablecidas, como por ejemplo cuáles son sus intereses (turismo, gastronomía, deportes, etc.) o simplemente si desea suscribirse al newsletter.

Al insertar una casilla de verificación también se debe establecer un ID y una etiqueta, como el resto de los elementos dentro del formulario.

5) GRUPO DE CASILLAS DE VERIFICACIÓN: Esta opción nos permite insertar un grupo de opciones en lugar de una sola e individual. Se debe proporcionar un nombre al grupo de opciones y establecer una Etiqueta (label) y un Valor (value) para cada una de las opciones:



Grupo de casillas de verificación

Nombre:

Casillas de verificación:

Label	Value
Deportes	deportes
Turismo	turismo

Disponer utilizando: ☒ Saltos de línea (etiquetas
)

☐ Tabla

El "Label" será la palabra que verá el usuario, sobre la cual hará su elección y el "Value" será el equivalente de cada Label para el código html e indicará la opción seleccionada por el usuario. Los "values" deben estar escritos con minúsculas, sin caracteres espaciales ni espacios, en cambio los "labels" al ser texto común pueden usar cualquier tipo de caracteres.

6) BOTÓN DE OPCIÓN: cumple la misma función de una casilla de verificación, pero el formato del área donde el usuario puede clicar es circular en lugar de un cuadrado.

7) GRUPO DE OPCIÓN: a diferencia de las 2 anteriores herramientas, el grupo de opción me permite ingresar varias opciones de una sola vez, usando un sólo cuadro de diálogo para establecer los valores o nombres de esas opciones. Por ejemplo puede ser usado para definir si el usuario es Masculino o Femenino. En la columna introduciremos los nombre que le queremos dar a las opciones. El usuario sólo será capaz de seleccionar una sola opción.

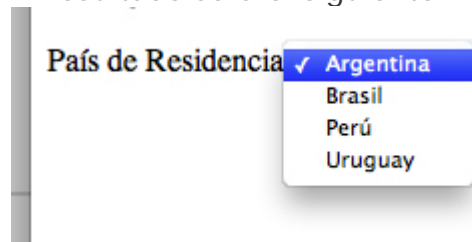
8) LISTA/MENÚ: con esta herramienta podremos crear una lista desplegable para que el usuario seleccione una opción, por ejemplo para indicar el país de residencia.

Al hacer clic en la herramienta deberemos, al igual que en las herramientas anteriores, definir un ID y un nombre visible para el usuario. Al hacer clic en Aceptar veremos que se ha insertado un cuadro blanco con una flecha de menú de opción, sin valores. Si seleccionamos esto podemos ver que la barra de propiedades nos deja introducir los valores que deseemos estén disponibles para el usuario.

Si hacemos clic en el botón "Valores de lista" se nos abrirá una nueva ventana de diálogo. Bajo la columna "Etiqueta de elemento" introduciremos los valores mencionados.

Podemos introducir tantos valores como queramos. Para añadir nuevos valores simplemente haremos clic en el botón con el signo "suma", y si deseamos borrar una de las opciones deberemos seleccionarla y hacer clic en el botón con el signo "restar". Esta lista de opciones por lo general está ordenada alfabéticamente, por lo cual tenemos la posibilidad de alterar el orden de los mismos haciendo clic en las flechas hacia arriba y abajo que aparecen cerca del botón "Aceptar".

El resultado será el siguiente:



9) MENÚ DE SALTO: este tipo de menú desplegable de opciones es muy similar al anterior, con la diferencia que el usuario, al hacer clic sobre una de las opciones será re-direccionado a otro html dentro del sitio, o bien a un URL externo (otro sitio web).

Aquí también podremos definir los valores de la lista, ponerles nombre, agregar o restar valores, y re-ordenarlos según conveniencia.

Donde dice "Al seleccionarse, ir a URL:" podremos ingresar el link a donde deseamos que esa opción lleve al usuario.

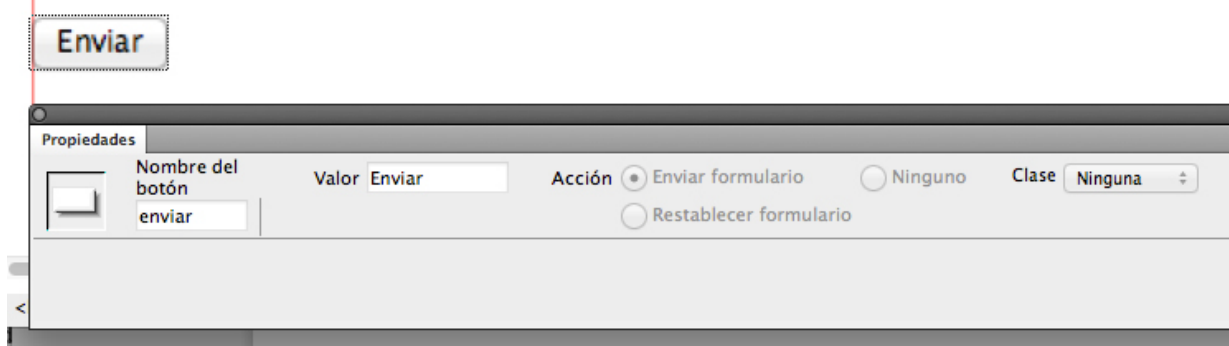
El aspecto final de esta lista de opciones será el mismo que "LISTA/MENÚ".

10) CAMPO DE IMAGEN: nos servirá para incrustar una imagen en nuestro formulario.

11) CAMPO DE ARCHIVO: con esta herramienta podremos insertar un botón tipo "Examinar", donde, al hacer clic, el usuario podrá adjuntar un archivo de cualquier tipo. Tal y como lo hacemos al adjuntar un archivo al enviar un email.

12) BOTÓN: con esta herramienta podremos insertar un botón en nuestro formulario. Este botón puede servir para enviar el contenido del formulario o para borrar lo que se ha escrito en el mismo, o sea, para re-establecer el formulario y que quede en blanco otra vez.







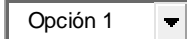

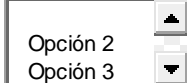
Una vez que le dimos el nombre al botón y hacemos clic en Aceptar, la barra de propiedades se verá así:



Donde dice "Valor" va el texto que queremos sobre el botón, por ejemplo "Enviar".

También tenemos 3 opciones correspondientes a la acción que deseamos que el botón cumpla, simplemente debemos hacer clic en la opción deseada.

- Atributos de etiquetas FORM y tipos de entrada

• Etiqueta	Atributo	Valor	Resultado	Efecto visual
<FORM> ... </FORM>	METHOD	POST GET		
	ACTION		Envía a la dirección mostrada	
	ENCTYPE		Especifica el tipo de código	
<INPUT>	TYPE	submit	realiza la ACTION de la etiqueta <FORM>	
		text	línea simple de texto cuya longitud se especifica por el atributo SIZE	
		Reset	Elimina el contenido del formulario	
		Radio	botón de radio	
		Checkbo x	casilla de selección	
	NAME		Nombre	
	SIZE		Tamaño del texto	
<TEXTAREA> ... </TEXTAREA>	NAME		Casilla de texto	
	ROWS			
	COLS			
<SELECT> <OPTION> ... </OPTION> </SELECT>	NAME			
	MULTIPLE		Múltiples selecciones posibles	
<OPTION> ... </OPTION>	SELECTED	Elección predeter minada		
	VALUE	Valor		

		forzado		
--	--	---------	--	--

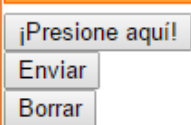
1.2.3. Botones.

La etiqueta **<button>** se utiliza para crear botones de un formulario.

Los botones creados con esta etiqueta funcionan de igual manera que los botones creados con la etiqueta **<input>**, pero permiten mejores posibilidades de presentación. Por ejemplo, podemos asociar una imagen y un texto al mismo botón, podemos colocar dentro del botón cualquier contenido que se desee, etc.

El elemento HTML button inserta un control de botón en un formulario o documento. A diferencia del elemento HTML input con el valor "button" en su atributo "type", este botón puede contener a otros elementos (es decir, código HTML).

Ejemplo

Código	Resultado
<pre><button type="button">¡Presione aquí!</button> <button type="submit">Enviar</button> <button type="reset">Borrar</button></pre>	

Las opciones que permite el elemento **<button>** son las siguientes:

- **Name:** Nombre del botón.
- **Value:** Valor inicial del botón.
- **Disabled:** El valor es opcional, y si establecemos disabled=false entonces el botón está deshabilitado.
- **Type:** Indica el tipo de botón que se va a crear. Los posibles valores son:
 - **button:** Crea un botón pulsador. En principio no realiza ninguna acción y se utiliza para asociarle acciones mediante lenguajes scripts.
 - **submit:** Crea un botón de envío del formulario.
 - **reset:** Crea un botón de restablecimiento del formulario.

- **Atributos Estándar de la Etiqueta <button>**

Atributos	Valor	Descripción
class	nombre de la clase	Asigna un nombre de clase. El atributo class actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS), cuando se asigna información de estilo a un conjunto de elementos. • Para procesos generales por parte del usuario.
id	nombre	Asigna un nombre a un elemento. El atributo id actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS). • Como vínculo destino para vínculos de hipertexto. • Como medio de hacer referencia a un elemento en particular desde un script. • Como nombre de un elemento object declarado. • Para procesos generales por parte del usuario.
style	estilo	Este atributo especifica información de estilo para el elemento actual.
title	nombre	Este atributo ofrece información consultiva sobre el elemento para el cual se establece.
dir	ltr o rtl	Especifica la dirección del texto. Valores posibles: <ul style="list-style-type: none"> • ltr : De izquierda a derecha (left to right). • rtl : De derecha a izquierda (right to left).
lang	código de lenguaje	Especifica el idioma base de los valores de los atributos y del texto contenido en un elemento. El atributo lang es útil para: <ul style="list-style-type: none"> • Ayudar a los motores de búsqueda. • Ayudar a los sintetizadores de voz. • Ayudar al agente de usuario a hacer decisiones sobre separación de palabras, ligaduras, y espaciado. • Ayudar a los verificadores de ortografía y gramática.
accesskey	caracter	Asigna una tecla de acceso a un elemento. Al pulsar esa tecla el foco se dirige hacia ese elemento.
tabindex	número	Especifica el orden de tabulación del elemento dentro del documento actual

- Eventos de la Etiqueta **<button>**

Evento	Valor	Descripción
onblur	script	El script corre cuando el elemento pierde el foco ya sea con el mouse o por navegación con tabulador.
onfocus	script	El script corre cuando el foco se dirige hacia un elemento, ya sea con el mouse o por navegación con tabulador.
onkeydown	script	El script corre cuando se pulsa una tecla.
onkeypress	script	El script corre cuando se pulsa y se suelta una tecla.
onkeyup	script	El script corre cuando una tecla se suelta.
onclick	script	El script corre cuando se hace clic con el mouse.
ondblclick	script	El script corre cuando se hace doble clic con el mouse.
onmousedown	script	El script corre cuando el botón del mouse se pulsa cuando está encima de un elemento.
onmousemove	script	El script corre cuando el mouse se mueve mientras está sobre un elemento.
onmouseout	script	El script corre cuando el mouse se aparta de un elemento.
onmouseover	script	El script corre cuando el mouse se sitúa sobre un elemento.
onmouseup	script	El script corre cuando el botón del mouse se suelta cuando está encima de un elemento.

¿Y QUE BOTON VOY A UTILIZAR?

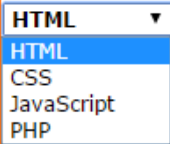
La directiva **<input>** es más de formularios, es decir se identifica mucho con la directiva **<form>**, por eso cuando quieres un botón para el envío o reseteo de un formulario **<input>** es la solución. En cambio cuando quieres un botón que te abra alguna aplicación, pagina Web, que cambie colores, que inicie la reproducción de un video la directiva **<button>** es la solución.

1.2.4. Listas desplegables

La etiqueta **<select>** se utiliza para crear los elementos **combobox** o **listas desplegables**. Crea un menú desplegable.

Cada opción ofrecida por dicho menú se representa por un elemento **option**.

Ejemplo

Código	Resultado
<pre> <select> <option value="html">HTML</option> <option value="css">CSS</option> <option value="js">JavaScript</option> <option value="php">PHP</option> </select> </pre>	

- Posibles Atributos de la Etiqueta <select>

Atributos	Valor	Descripción
disabled	disabled	Deshabilita el control para la entrada de datos.
multiple	multiple	Si está activado, permite selecciones múltiples.
name	nombre	Especifica un nombre para el menú de desplazamiento.
size	número	Especifica el número de filas de la lista que deberían ser visibles al mismo tiempo.

- Atributos Estándar de la Etiqueta <select>

Atributos	Valor	Descripción
class	nombre de la clase	Asigna un nombre de clase. El atributo class actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS), cuando se asigna información de estilo a un conjunto de elementos. • Para procesos generales por parte del usuario.
id	nombre	Asigna un nombre a un elemento. El atributo id actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS). • Como vínculo destino para vínculos de hipertexto. • Como medio de hacer referencia a un elemento en particular desde un script. • Como nombre de un elemento object declarado. • Para procesos generales por parte del usuario.
style	estilo	Este atributo especifica información de estilo para el elemento actual.
title	nombre	Este atributo ofrece información consultiva sobre el elemento para el cual se establece.
dir	ltr o rtl	Especifica la dirección del texto. Valores posibles: <ul style="list-style-type: none"> • ltr : De izquierda a derecha (left to right). • rtl : De derecha a izquierda (right to left).
lang	código de lenguaje	Especifica el idioma base de los valores de los atributos y del texto contenido en un elemento. El atributo lang es útil para: <ul style="list-style-type: none"> • Ayudar a los motores de búsqueda. • Ayudar a los sintetizadores de voz. • Ayudar al agente de usuario a hacer decisiones sobre separación de palabras, ligaduras, y espaciado. • Ayudar a los verificadores de ortografía y gramática.
tabindex	número	Especifica el orden de tabulación del elemento dentro del documento actual

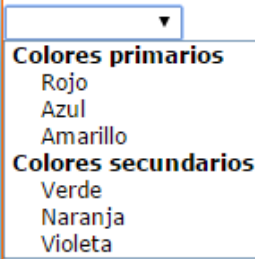
- **Eventos de la Etiqueta <select>**

Evento	Valor	Descripción
onkeydown	script	El script corre cuando se pulsa una tecla.
onkeypress	script	El script corre cuando se pulsa y se suelta una tecla.
onkeyup	script	El script corre cuando una tecla se suelta.
onclick	script	El script corre cuando se hace clic con el mouse.
ondblclick	script	El script corre cuando se hace doble clic con el mouse.
onmousedown	script	El script corre cuando el botón del mouse se pulsa cuando está encima de un elemento.
onmousemove	script	El script corre cuando el mouse se mueve mientras está sobre un elemento.
onmouseout	script	El script corre cuando el mouse se aparta de un elemento.
onmouseover	script	El script corre cuando el mouse se sitúa sobre un elemento.
onmouseup	script	El script corre cuando el botón del mouse se suelta cuando está encima de un elemento.

- **Etiqueta <optgroup>**

La etiqueta **<optgroup>** se utiliza para agrupar opciones relacionadas, por medio de un título común, en los menús desplegables.

Ejemplo

Código	Resultado
<pre> <select> <optgroup label="Colores primarios"> <option value="rojo">Rojo</option> <option value="azul">Azul</option> <option value="amarillo">Amarillo</option> </optgroup> <optgroup label="Colores secundarios"> <option value="verde">Verde</option> <option value="naranja">Naranja</option> <option value="violeta">Violeta</option> </optgroup> </select> </pre>	

1.2.5. Etiquetas.

Las etiquetas proporcionan información al usuario sobre el tipo de información que guarda el elemento de entrada.

La etiqueta **<label>** se utiliza para adjuntar información a los controles. El valor del atributo **for** debe ser igual al valor id del elemento input para que los reconozca como asociados.

Ejemplo

Código	Resultado
<pre><form> <input type="radio" name="internet" id="html" /> <label for="html">HTML</label>
 <input type="radio" name="internet" id="css" /> <label for="css">CSS</label>
 <input type="radio" name="internet" id="js" /> <label for="js">JavaScript</label> </form></pre>	<input type="radio"/> HTML <input type="radio"/> CSS <input type="radio"/> JavaScript

• Posibles Atributos de la Etiqueta <label>

Atributos	Valor	Descripción
class	nombre de la clase	Asigna un nombre de clase. El atributo class actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS), cuando se asigna información de estilo a un conjunto de elementos. • Para procesos generales por parte del usuario.
id	nombre	Asigna un nombre a un elemento. El atributo id actúa: <ul style="list-style-type: none"> • Como selector para las hojas de estilo(CSS). • Como vínculo destino para vínculos de hipertexto. • Como medio de hacer referencia a un elemento en particular desde un script. • Como nombre de un elemento object declarado. • Para procesos generales por parte del usuario.
style	estilo	Este atributo especifica información de estilo para el elemento actual.
title	nombre	Este atributo ofrece información consultiva sobre el elemento para el cual se establece.
dir	ltr o rtl	Especifica la dirección del texto. Valores posibles: <ul style="list-style-type: none"> • ltr : De izquierda a derecha (left to right). • rtl : De derecha a izquierda (right to left).
lang	código de lenguaje	Especifica el idioma base de los valores de los atributos y del texto contenido en un elemento. El atributo lang es útil para: <ul style="list-style-type: none"> • Ayudar a los motores de búsqueda. • Ayudar a los sintetizadores de voz. • Ayudar al agente de usuario a hacer decisiones sobre separación de palabras, ligaduras, y espaciado. • Ayudar a los verificadores de ortografía y gramática.
accesskey	caracter	Asigna una tecla de acceso a un elemento. Al pulsar esa tecla el foco se dirige hacia ese elemento.

- Eventos de la Etiqueta <label>

Evento	Valor	Descripción
onblur	script	El script corre cuando el elemento pierde el foco ya sea con el mouse o por navegación con tabulador.
onfocus	script	El script corre cuando el foco se dirige hacia un elemento, ya sea con el mouse o por navegación con tabulador.
onkeydown	script	El script corre cuando se pulsa una tecla.
onkeypress	script	El script corre cuando se pulsa y se suelta una tecla.
onkeyup	script	El script corre cuando una tecla se suelta.
onclick	script	El script corre cuando se hace clic con el mouse.
ondblclick	script	El script corre cuando se hace doble clic con el mouse.
onmousedown	script	El script corre cuando el botón del mouse se pulsa cuando está encima de un elemento.
onmousemove	script	El script corre cuando el mouse se mueve mientras está sobre un elemento.
onmouseout	script	El script corre cuando el mouse se aparta de un elemento.
onmouseover	script	El script corre cuando el mouse se sitúa sobre un elemento.
onmouseup	script	El script corre cuando el botón del mouse se suelta cuando está encima de un elemento.

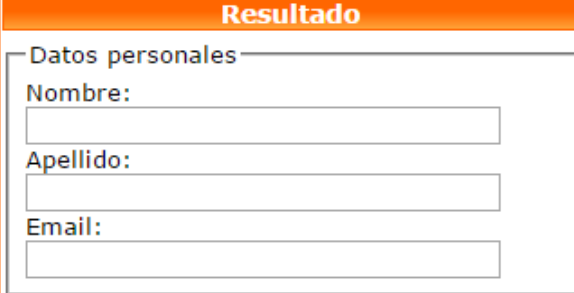
1.2.6. Grupos de elementos.

Las etiquetas **<fieldset>** y **<legend>** permiten la creación de un grupo de elementos agrupados por un área visible dentro del formulario, facilitando al usuario su comprensión y usabilidad, ya que permite desplazarse de un control a otro a través de la tecla TAB y le proporciona una etiqueta a modo de leyenda con una descripción global del conjunto de elementos.

La etiqueta **<fieldset>** se utiliza para agrupar los controles de un formulario por medio de un recuadro. De esta forma, el formulario es más fácil de entender para los usuarios.

La etiqueta **<legend>** se utiliza para asignar un título a un fieldset.

Ejemplo

Código	Resultado
<pre> <form> <fieldset> <legend>Datos personales</legend> Nombre: <input type="text" size="30" />
 Apellido: <input type="text" size="30" />
 Email: <input type="text" size="30" /> </fieldset> </form> </pre>	

1.3. Eventos.

Los eventos son una característica de los documentos HTML (presente en otros lenguajes de programación también) que permite a los autores agregar interactividad entre el sitio web y el visitante, al ejecutar programas del lado cliente cuando el visitante (u otro programa) realiza una acción. Por ejemplo, el autor puede hacer que un párrafo cambie de color de su texto cuando el visitante pose el puntero del mouse sobre el mismo.

Así como los atributos, los eventos pueden ser definidos fácilmente en el tag de apertura del elemento, con el mismo formato:

<nombretag evento="codigo">

El contenido del evento es el código que se ejecutará, y debe ser creado utilizando un lenguaje del lado cliente (por ejemplo, JavaScript) que debe ser soportado por el navegador para que funcione. En el ejemplo siguiente, definimos un párrafo que cambia el color de su texto a rojo cuando el mouse pasa por encima, y lo devuelve a negro cuando se retira.

<p onmouseover="this.style.color='red'" onmouseout="this.style.color='black'">Este es un texto que cambia de color. Pruébalo!</p>

Evento	Descripción
Eventos de ventana	
onLoad:	Permiten lanzar un script a la carga del documento
onUnload:	Todo lo contrario, permitirán lanzar un script al cierre del documento.
Eventos de formulario	
onChange	Detecta cuando un elemento cambia, por ejemplo cuando se inserta un carácter, o se pulsa sobre un botón.
onSubmit	Lanza un script cuando el formulario haya sido enviado.
onReset	Revela cuándo un formulario ha sido reseteado.
onSelect	Ejecuta un script cuando un elemento es seleccionado.
onBlur	Se puede ver cuando el elemento pierde el foco.
onFocus	Detecta cuando el elemento obtiene el foco.
Eventos de teclado	
onKeyDown	Lanza un script cuando una tecla es pulsada.
onKeyPress	En este caso cuando la tecla es pulsada y devuelta a su posición.
onKeyUp	Lanza el evento al soltar una tecla.

Eventos de ratón	
onClick	Lanza un script cuando se hace un click de ratón.
onDbIClick	Lanza un script cuando se hace un dobleclick de ratón.
onMouseDown	Detecta cuando el botón del ratón es presionado.
onMouseMove	Ejecuta el script cuando movemos el puntero del ratón.
onMouseOut	Obtiene cuando el puntero del ratón se mueve fuera de la zona de un elemento.
onMouseOver	Revela que el puntero del ratón se está desplazando sobre el área de un elemento.
onMouseUp	Lanza el script cuando el botón del ratón es liberado.

1.4. Creación de formularios.

Crear formularios puede ser una tarea bastante sencilla, debido a la existencia de numerosas herramientas online que permiten generar formularios en cuestión de minutos.

- **Creación de formularios online gratuitos:**

<http://www.suformulario.com/>



Configurador de formularios web 10128 formularios

Formularios web on-line gratuito.

1º Elija el tipo de formulario que más se adapta a sus necesidades

2º Marque los campos que quiera usar en su formulario.

3º Marque los campos de obligatoria cumplimentación.

4º Elija una apariencia.

5º Inserte su correo electrónico.

6º Acepte la política de privacidad.

7º Guarde la configuración y obtenga el código HTML

Seleccione un tipo: Formulario de Contacto

Selección Obligatorio

Tratamiento	<input type="checkbox"/>	<input type="checkbox"/>
Nombre	<input type="checkbox"/>	<input type="checkbox"/>
Apellidos	<input type="checkbox"/>	<input type="checkbox"/>
E-mail	<input type="checkbox"/>	<input type="checkbox"/>
Organización / Empresa	<input type="checkbox"/>	<input type="checkbox"/>
Teléfono	<input type="checkbox"/>	<input type="checkbox"/>
Código postal	<input type="checkbox"/>	<input type="checkbox"/>
Listado de motivos	<input type="checkbox"/>	<input type="checkbox"/>
Asunto	<input type="checkbox"/>	<input type="checkbox"/>
Página web	<input type="checkbox"/>	<input type="checkbox"/>
Comentario	<input type="checkbox"/>	<input type="checkbox"/>

Estilo: Blanco Bord. Redondos

Tu correo:

☐ [Acepto la política de privacidad](#)

[Donar](#) | [Accede a la aplicación](#) | [Ayuda](#) | [Contacto](#) | [Blog](#) | Ver. 2.4



Foxyform.com

Formulario de contacto para su sitio web con protección anti-spam integrada!

Cree su propio formulario de contacto en tan sólo unos segundos. Por supuesto, es gratuito e incluye capacidad anti-spam integrada.

1 Configuración

¿Qué campos desearía incluir en su formulario de contacto personal?
Por favor especifique qué campos son obligatorios. Los campos obligatorios deben ser rellenados por el usuario sinó el formulario no será enviado.

SUS OPCIONES:

<input type="checkbox"/> Saludo	obligatorio: <input type="checkbox"/>
<input checked="" type="checkbox"/> Apellido	obligatorio: <input type="checkbox"/>
<input checked="" type="checkbox"/> E-Mail	obligatorio: <input checked="" type="checkbox"/>
<input type="checkbox"/> Asunto	obligatorio: <input type="checkbox"/>

Mostrar más opciones

2 Configuración avanzada

Ajuste el formulario al diseño que mejor se adapte a su sitio web personal.

SUS OPCIONES:

Color de fondo:	# ffffff	(Colores)
Color de fuente:	# 000000	(Colores)
Fuente:	Arial	▼
Tamaño de la fuente:	normal	▼

- **Creación de formularios de pago con planes free:**

Wufoo cuenta con la posibilidad de adaptar su diseño a nuestros colores, logos y todo lo relacionado con la presentación visual de nuestro sitio. Su versión gratuita tiene menos opciones que la de pago, permitiendo un máximo de 100 entradas al mes.

<http://www.wufoo.com.mx/>

FormSite es una herramienta potente en cuanto a su flexibilidad y cantidad de opciones. Podemos crear formularios completamente personalizados con hasta 40 campos a completar. El mismo puede incluirse en nuestro propio sitio o alojarse en el sitio de FormSite incluyendo un enlace en nuestra web.

<https://www.formsite.com/>

FormAssembly tiene una amigable interfaz del tipo arrastrar y soltar. Cuenta también con una gran variedad de opciones para adaptarlo a nuestras necesidades, incluyendo variedad de idiomas.

<http://www.formassembly.com/>

JotForm está especialmente diseñado para implementarse a través de comercio electrónico. Viene con varias opciones de pago incluidas como PayPal y Google Checkout.

<https://spanish.jotform.com/>

1.5. Elementos de formulario adicionales del HTML5.

El HTML5 incluye muchas nuevas características para hacer los formularios web más fáciles de crear, más poderosos y consistentes a través de la Web. A continuación se presenta un breve resumen acerca de algunos de los nuevos controles y funcionalidades de formulario que han sido introducidos.

Como los formularios son la principal herramienta para la introducción de datos en las aplicaciones web y los datos que deseamos recopilar se han vuelto más complejos, ha sido necesario crear un elemento input con más capacidades, para recolectar estos datos con más semántica y una mejor definición, además de permitir un más fácil y eficaz manejo de errores y validación.

- **`<input type="number">`**

El primer nuevo tipo de campo de entrada que analizaremos es `type="number"`:

```
<input type="number" ... >
```

Este crea un tipo especial de campo de entrada para la introducción de un número. En la mayoría de los navegadores que lo soportan, aparece como un campo de entrada de texto con un control de número, que le permite aumentar o disminuir su valor.



Figura 1: Un campo de entrada de tipo `number`.

- **`<input type="range">`**

Crear un control deslizante que le permita elegir entre un rango de valores solía ser una propuesta complicada, semánticamente dudosa, pero con el HTML5 es fácil: sólo tiene que utilizar el tipo de campo de entrada `range`:

```
<input type="range" ... >
```



Figura 2: Un campo de entrada `range`.

Tengamos en cuenta que, por defecto, este campo de entrada no suele mostrar el valor seleccionado en el momento actual, o incluso el rango de valores que

abarca. Los desarrolladores deben proporcionarlos mediante otros medios; por ejemplo, para mostrar el valor actual, podríamos utilizar un elemento `<output>` junto con algo de JavaScript para actualizar la representación del control cuando el usuario ha interactuado con el formulario:

```
<output onforminput="value=weight.value"></output>
```

- **`<input type="date">` y otros controles de fecha/hora**

HTML5 tiene un número de campos de entrada diferentes para crear selectores complejos de fecha/hora; por ejemplo, el tipo de selector de fecha que aparece en prácticamente todo sitio de reserva de vuelo/tren existente. Estos suelen crearse utilizando trucos no semánticos, por lo que es estupendo que ahora tengamos formas estandarizadas fáciles para hacerlo. Por ejemplo:

```
<input type="date" ... >
```

```
<input type="time" ... >
```

Estos crean, respectivamente, un selector de fecha totalmente funcional y un campo de entrada de texto que contiene un separador para horas, minutos y segundos (dependiendo del atributo `step` especificado) que sólo le permite introducir un valor de tiempo.

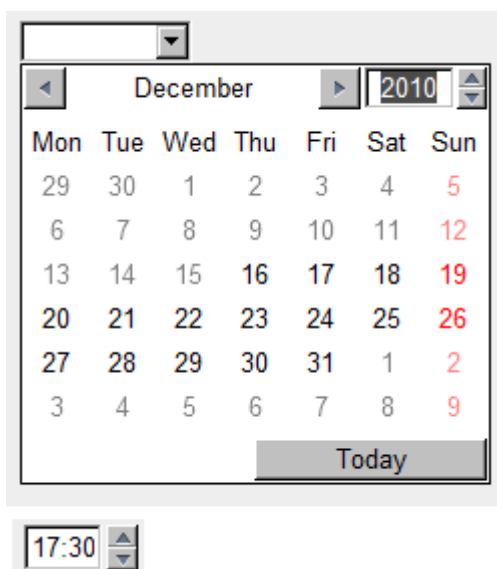


Figura 3: Campos de entrada `date` y `time`.

Pero aquí no termina todo — hay un número de otros campos de entrada relacionados disponibles:

- `datetime`: combina la funcionalidad de los dos que hemos analizado anteriormente, permitiéndole elegir una fecha y una hora.
- `month`: le permite elegir un mes, almacenado internamente como un número entre 1-12; aunque los diferentes navegadores pueden

proporcionarle mecanismos de selección más elaborados, como una lista desplegable con los nombres de los meses.

- o week: le permite elegir una semana, almacenada internamente en el formato 2010-W37 (semana 37 del año 2010) y elegida mediante un selector de fecha similar a los que ya hemos visto.

- **<input type="color">**

Este campo de entrada muestra un selector de color. La implementación de Opera le permite al usuario elegir entre una selección de colores, introducir valores hexadecimales directamente en un campo de texto o invocar el selector de colores nativo del Sistema Operativo.

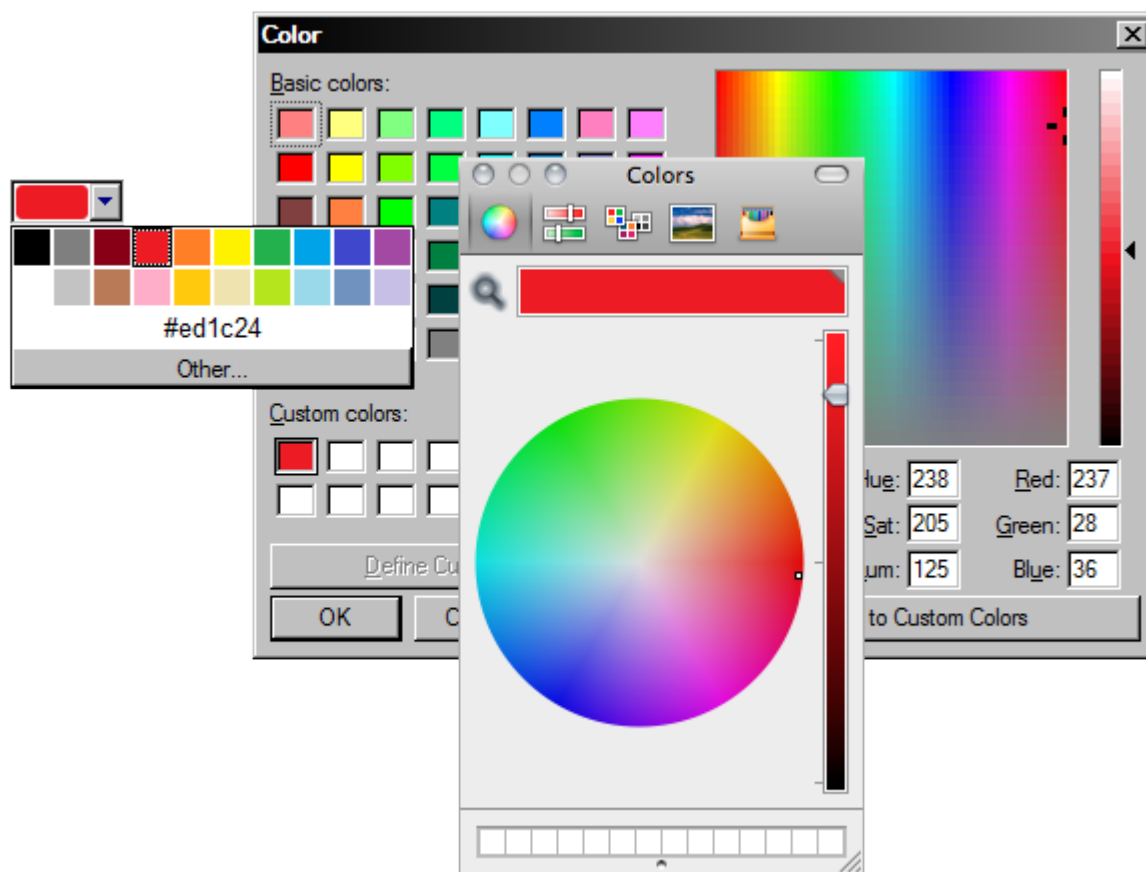


Figura 4: un campo de entrada `color` y los selectores de color nativos de Windows y OS X.

- **<input type="search">**

El campo de entrada `search` podría decirse que no es nada más que un campo de entrada de texto con un estilo diferente. Los navegadores deberían aplicarle a estos campos de entrada el mismo estilo que a cualquier funcionalidad de búsqueda específica del sistema operativo. Sin embargo, más allá de esta consideración puramente estética, sigue siendo importante tener en cuenta que el etiquetar explícitamente campos de búsqueda abre la posibilidad para los navegadores, ayudas técnicas o rastreadores automatizados, de hacer algo inteligente con estas entradas en el futuro; por ejemplo, un navegador podría,

posiblemente, ofrecer al usuario una opción para crear automáticamente una búsqueda personalizada para un sitio específico.



Figura 5: Un campo de entrada `search` como se ve en Opera en OS X.

- **El elemento `<datalist>` y el atributo `list`**

Hasta ahora hemos estado acostumbrados a utilizar los elementos `<select>` y `<option>` para crear listas desplegadas de opciones donde nuestros usuarios pueden elegir. Pero, ¿y si quisiéramos crear una lista que le permita a los usuarios elegir entre una lista de opciones sugeridas, así como la posibilidad de introducir su propia opción? Esto solía requerir secuencias de comandos complicadas, pero ahora puede simplemente usar el atributo `list` para conectar un campo de entrada corriente a una lista de opciones, definidas dentro de un elemento `<datalist>`.

```
<input type="text" list="misdatos" ... >
<datalist id="misdatos">
  <option label="Sr." value="Señor">
  <option label="Sra." value="Señora">
  <option label="Srta." value="Señorita">
</datalist>
```

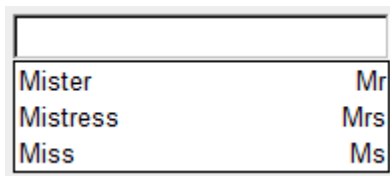


Figura 6: Creación de un campo de entrada con sugerencias usando `datalist`.

- **`<input type="tel">`, `<input type="email">` e `<input type="url">`**

Como sus nombres lo indican, estos nuevos campos de entrada se refieren a números de teléfono, direcciones de correo electrónico y URLs. Los navegadores los mostrarán como campos de entrada de texto normales, pero señalar claramente qué tipo de texto estamos esperando en estos campos juega un papel importante en la validación de formularios del lado del cliente. Además, en ciertos dispositivos móviles el navegador cambiará su teclado en pantalla para entrada de texto común por sus variantes más relevantes al contexto. Una vez más, es concebible que en el futuro los navegadores aprovecharán mejor estas entradas explícitamente definidas para ofrecer funcionalidades adicionales, como el autocompletado de direcciones de correos electrónicos y números telefónicos basándose en la lista de contactos o la libreta de direcciones del usuario.



- **Nuevos atributos**

Además de nuevos campos de entrada explícitos, el HTML5 define una serie de nuevos atributos para los controles de formulario que ayudan a simplificar algunas tareas comunes y precisar los valores esperados para determinados campos de entrada.

- **placeholder**

Un truco común de usabilidad para los formularios web es tener un contenido predeterminado en los campos de entrada de texto; por ejemplo, para dar sugerencias acerca del tipo de información que esperamos que el usuario introduzca, y que desaparecerá cuando dicho control de formulario reciba el foco del usuario. Aunque esto solía requerir algo de JavaScript (borrar el contenido del campo de formulario al recibir el foco y restablecer el texto predeterminado si el usuario retira el foco del campo sin introducir nada), ahora podemos simplemente utilizar el atributo placeholder:

```
<input type="text"... placeholder="John Doe">
```

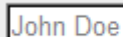


Figura 7: Un campo de entrada de texto con texto predeterminado mediante el uso de placeholder.

- **autofocus**

Otra característica común que anteriormente dependía de secuencias de comandos es la colocación automática del foco en un campo de formulario cuando se carga una página. Esto puede lograrse ahora con el atributo autofocus:

```
<input type="text" autofocus ... >
```

Tenga en cuenta que no debe tener más de un control de formulario con el atributo autofocus en una misma página. Además, debe utilizar este tipo de funcionalidad con cautela, en situaciones donde un formulario representa la principal área de interés en una página. Una página de búsqueda es un buen ejemplo, siempre que no haya mucho contenido y texto descriptivo, tiene sentido establecer el foco automáticamente en el campo de entrada de texto del formulario de búsqueda.

- **min y max**

Como su nombre sugiere, este par de atributos le permiten establecer un límite inferior y superior para los valores que se pueden introducir en un campo de entrada numérico; por ejemplo, campos de entrada de número, rango (control deslizante), hora o fecha (sí, incluso se puede utilizar para establecer los límites superior e inferior de las fechas, por ejemplo, en un formulario de reserva de vuelos podría limitar el selector de fechas para que sólo permita que el usuario seleccione fechas futuras). Para campos de entrada range, min y max son realmente necesarios para definir los valores que se devuelven cuando se envía el formulario. El código es bastante simple y autoexplicativo:

```
<input type="number" ... min="1" max="10">
```

- **step**

El atributo step puede utilizarse con un valor de campo de entrada numérico para establecer la precisión de los valores que se pueden ingresar. Por ejemplo, puede que desee que los usuarios introduzcan un tiempo determinado, pero sólo en incrementos de 30 minutos. En este caso, podemos utilizar el atributo step, teniendo en cuenta que para campos de entrada time el valor de este atributo se establece en segundos:

```
<input type="time" ... step="1800">
```

- **Nuevos mecanismos de salida**

Más allá de los nuevos controles de formulario con los que los usuarios pueden interactuar, HTML5 define una serie de nuevos elementos específicamente destinados a mostrar y visualizar información para el usuario.

- **<output>**

Ya hemos mencionado el elemento <output> cuando hablamos del campo de entrada range. Este elemento sirve como una forma de mostrar el resultado de

un cálculo, o más generalmente, para proporcionar una salida explícitamente identificada a una secuencia de comandos (en lugar de simplemente colocar un texto dentro de un span o div al azar). Para hacer aún más explícito a que controles de formulario en particular está vinculado el `<output>`, podemos, de una manera similar al elemento `<label>`, pasar una lista de IDs en el atributo opcional `for` del elemento.

```
<input type="range" id="ejemplorango" ... >
<output onforminput="value=ejemplorango.value"
for="ejemplorango"></output>
```

- **`<progress>` y `<meter>`**

Estos dos nuevos elementos son muy similares. Ambos muestran un indicador/barra al usuario como resultado. Lo que los distingue es su propósito. Como su nombre sugiere, `<progress>` tiene el fin de representar una barra de progreso para indicar el porcentaje de finalización de una tarea determinada, mientras que `<meter>` es un indicador más genérico de una medida escalar o valor fraccionario.

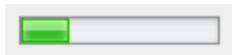


Figura 8: Una barra indicadora de progreso.

- **Validación**

La validación de formularios es muy importante tanto en el lado cliente, como en el lado servidor, para ayudar a los usuarios reales a evitar y corregir los errores y para evitar que usuarios maliciosos envíen datos que pudieran causar daños a nuestra aplicación. Dado que los navegadores pueden ahora tener una idea de qué tipo de valores se esperan para los diversos controles de formulario (ya sea mediante su `type`, o cualquier límite superior/inferior establecido en valores numéricos, fechas y horas), pueden ofrecer adicionalmente validación de formulario nativa; otra tarea tediosa, que, hasta ahora, requería que los desarrolladores crearan resmas de JavaScript o utilizaran algún script/biblioteca de validación ya existente.

Nota: para que los controles de formulario sean validados, necesitan tener un atributo `name`, ya que sin el no serán enviados como parte del formulario.

- **`required`**

Uno de los aspectos más comunes de la validación de formularios es la implementación de campos obligatorios, no permitiendo que un formulario sea enviado hasta que ciertas piezas de información hayan sido introducidas. Esto puede hacerse ahora simplemente añadiéndole el atributo `required` a un elemento `input`, `select` o `textarea`.

```
<input type="text" ... required>
```



Figura 9: La validación en lado cliente de Opera en acción, mostrando un error para un campo requerido que ha sido dejado vacío.

- **type y pattern**

Como hemos visto, los desarrolladores ahora pueden especificar los tipos de entradas que esperan en sus campos de formulario. En lugar de simplemente definir campos de entrada de texto, los desarrolladores pueden crear de forma explícita campos de entrada para cosas como números, direcciones de correo electrónico y URLs. Como parte de su validación en el lado cliente, los navegadores pueden ahora comprobar que los datos introducidos por el usuario en estos campos más específicos coinciden con la estructura prevista. En esencia, los navegadores evalúan los valores de la entrada en base a un patrón integrado que define como deben ser las entradas válidas en esos tipos de entrada y le advertirá al usuario cuando su entrada no coincida con los criterios.

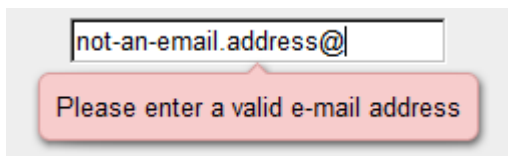


Figura 10: Mensaje de error de Opera para direcciones de correo-e inválidas en un campo de entrada email.

Para otros campos de entrada de texto que, no obstante, deban seguir una cierta estructura (por ejemplo, formularios de inicio de sesión donde los nombres de usuario sólo puedan contener una secuencia específica de letras minúsculas y números), los desarrolladores pueden utilizar el atributo pattern para especificar su propia expresión regular personalizada.

```
<input type="text" ... pattern="[a-z]{3}[0-9]{3}">
```

- **Soporte en los navegadores**

Por diseño, los navegadores que no entienden los nuevos tipos de entrada (como date o number) simplemente recurrirán a tratarlos como entradas de texto estándar; no tan fáciles de usar como sus contrapartes avanzadas de HTML5, pero al menos permiten que un formulario sea rellenado.

1.6. Accesibilidad y Usabilidad en los formularios

A la hora de diseñar un formulario es imprescindible seguir un conjunto de pasos para construir un formulario usable y accesible. En primer lugar se debe analizar la información que desea obtener del usuario y establecer la forma óptima la información.

Para ello se debe analizar las funciones y características que debe tener el formulario, así como cumplir una serie de requisitos básicos.

- Pedir exclusivamente la información necesaria, pedir información innecesaria hará dudar al usuario, así como incrementará el tiempo de dedicación del usuario a rellenarlo, minimizando el éxito de recolección de datos.
- Reutilizar los campos siempre que sea posible. Por ejemplo se podría utilizar el email, como nombre de usuario. De esta forma ya se elimina un dato.
- Si se necesitan formularios muy extensos, dividir estos en distintas páginas. Indicando siempre el estado de proceso del formulario, por ejemplo paso 2 de 5.
- Si se divide el formulario por páginas la información por página debe ser agrupada coherentemente por funcionalidad.
- Si el formulario tiene más de una página, siempre se debe permitir al usuario volver atrás.
- Evitar el uso de pestañas para el diseño de formularios extensos.
- Establecer un título para el formulario, que describa de forma breve y precisa la función del mismo.
- Organizar campos en una sola columna.
- Organizar los campos por grupos lógicos.
- Agrupar, si es posible, los campos obligatorios al principio del formulario.
- Identificar claramente los campos obligatorios y opcionales, mediante etiquetas. Identificando los obligatorios siempre y cuando sean menos que los opcionales o viceversa.
- Establecer un formulario limpio, alineando verticalmente todas las etiquetas y campos.
- Evitar la posibilidad de que en determinados campos se introduzca información incorrecta, mediante el uso de lenguajes scripts. Por ejemplo, evitar que en un campo numérico se pueda introducir texto.
- Cuando se produzca un error al rellenar el formulario, identificar el error indicando el campo que lo ha provocado y porque motivo.

1.6.1. Criterios de accesibilidad

Para cumplir con los criterios de accesibilidad en formularios basta con cumplir tan solo cuatro de las pautas establecidas por WCAG que son:

- Asociar explícitamente las etiquetas con sus controles mediante la etiqueta `<label>`.
- Asociar implícitamente los controles y etiquetas, es decir, todo control debe tener una etiqueta significativa.
- Comprobar que todos los campos del usuario están accesibles por el tabulador, en el mismo orden que el establecido visualmente.
- No establecer tiempos límites para completar el formulario.
- Utilizar la opción `tabindex` para indicar el orden de los elementos para que el usuario pueda utilizar la tecla de tabulación para moverse entre los diferentes elementos de entrada.

2. Plantillas en la construcción de páginas web.

2.1. Introducción.

Una forma fácil y sencilla de crear una página web es la utilización de plantilla.

Una plantilla web está compuesta por un conjunto de páginas, imágenes, hoja de estilo, javascript, etc, que permiten la reutilización y diseño de un sitio web.

Las principales ventajas que nos ofrece el uso de plantillas son:

- **Reducir el tiempo de desarrollo:** Gracias a la plantilla tenemos desarrollado el aspecto gráfico y la creación de zonas fijas, por lo tanto podemos centrarnos exclusivamente en la creación de contenido.
- **Mantenimiento fácil:** Un sitio web desarrollado a partir de una plantilla permite realizar cualquier cambio global de manera muy rápida y fácil. Por

ejemplo, si se requiere cambiar el logo, modificándolo en la plantilla, se actualizan automáticamente todas las páginas del sitio.

- **Flexibilidad:** Separamos el diseño de la web del contenido, siendo mucho más flexible, ya que un cambio de diseño: color, fuente... se realiza de forma rápida y sencilla.
- **Coherencia:** Al utilizar plantillas se consigue que todas las páginas del sitio mantengan el mismo formato, distribución, tipo de fuentes. Mostrando al usuario una imagen corporativa, sólida y consistente.
- **Facil acceso e indexación por los buscadores:** Con las plantillas el contenido de la página queda mejor estructurado, facilitando a los robots de los buscadores analizar con facilidad, mejorando el posicionamiento.

Pero no todo son ventajas, el abuso del uso de plantillas produce sitios con apariencia similar y poco personalizados a los requerimientos reales de usuario.

2.2. Repositorios libres de plantillas.

Recurso
Plantillas web
http://www.tooplate.com/
http://www.dreamtemplate.com/
http://www.freewebsitetemplates.com/
http://www.misplantillasgratis.com/
25 lugares para descargar plantillas web gratuitas: http://www.cosassencillas.com/articulos/25-lugares-para-descargar-plantillas-web-gratuitas
49 plantillas libres y responsivas: http://designscrazed.org/free-responsive-html5-css3-templates/
Plantillas web + hosting
http://es.wix.com/
http://www.webnode.es/