

Nuevos controles de formularios HTML5

Atributos HTML5 (placeholder, autofocus, required)

Buscador (Search)

Teléfonos

URL

Email

Number

Rangos (range)

Date

Month

Week

Time

Datetime

Color

Datalist

Atributos HTML5

En HTML5, se agregaron varios tipos de controles nuevos para formularios, y atributos que permiten validar los datos de forma muy sencilla.

Placeholder (texto por defecto)

La funcionalidad de mostrar un texto por defecto dentro de un campo se llama placeholder.

La forma de implementar un texto de placeholder es muy sencilla:

```
<input type="text" placeholder="DD/MM/AAAA">
```

Fecha de Nacimiento:

Debería usarse para mostrar un ejemplo del formato requerido, y nunca como reemplazo del label. Por ejemplo, si el label dice "Fecha de nacimiento" el placeholder puede mostrar un ejemplo de cuántos dígitos, barras separadoras, etc. debe contener esa fecha.

Notemos que al escribir en el campo, desaparecerá ese texto, y si escribimos algo ya no volverá a aparecer; pero si borramos lo escrito, dejando vacío el campo, nuevamente se visualizará el texto del placeholder para volver a advertir al usuario acerca de la tarea pendiente.

También puede aplicarse a un textarea, no solo a un input.

Foco automático

Otra funcionalidad típica es la del atributo ***autofocus***, que posiciona el cursor automáticamente en el campo donde lo apliquemos. Por supuesto, solo podemos aplicar este atributo a un solo campo por formulario (ya que no puede haber dos cursores haciendo foco simultáneamente!). Y si por descuido lo aplicáramos a dos o más controles dentro de un mismo formulario, solo quedará activo el último de los campos, ya que el navegador lee secuencialmente nuestro código.

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

Ejemplo de autofocus

Veremos que el cursor queda parpadeando, a la espera de que utilicemos ese campo.

Una advertencia útil es que evitemos colocar a un mismo campo tanto un placeholder como autofocus, ya que al producirse el foco con autofocus, el placeholder no se visualizará, solo quedará el cursor parpadeando mostrando vacío el campo (aunque puede ser usada esta combinación justamente para que el mensaje se visualice una vez perdido el foco, si es lo que queremos).

Campos requeridos

Si queremos que en un campo sea obligatorio ingresar algún texto (y en caso contrario, no permitir el envío del formulario), debemos usar el atributo required

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

Aquí veremos en acción por primera vez la validación automática realizada por el navegador: si pulsamos el botón de “Enviar” del formulario dejando vacío un campo requerido, se evitará el envío del formulario, y aparecerán carteles avisando que el campo es obligatorio.

Cada navegador mostrará el texto en el idioma de ese navegador, y con ciertos efectos decorativos propios de cada navegador: la idea es que las personas se acostumbren a los textos y diseño que su navegador habitual les muestra, derivando a éste la funcionalidad de validación, uniformizando en todos los sitios que un usuario visita los mensajes de validación.

Nuevos elementos de interfaz

Además de los atributos que acabamos de conocer, otro de los objetivos detrás de los formularios HTML5 **es el de poder generar interfaces de interacción complejas** (por ejemplo: **selectores de calendario para fechas, horas, rangos de valores y hasta colores**) sin la necesidad de utilizar librerías JavaScript para eso. Y con el foco puesto en facilitar la carga de datos para usuarios de pantallas táctiles (móviles, tabletas, etc.), donde sea más fácil “arrastrar” o “elegir” que escribir.

Además, en HTML5, el marcado de formularios directamente permite asignarle a un campo el tipo de dato que recopilará, ocupándose a partir de ese instante el propio navegador de realizar la validación.

HTML5 agrega 13 nuevos tipos de input: es decir, no son etiquetas nuevas, sino que reutilizamos la misma etiqueta input, pero agregando nuevos tipos (nuevos valores del atributo type).

Estos 13 nuevos tipos de input son los siguientes:

Search (para búsquedas)

Tel (para teléfonos)

URL (para direcciones web)

Email (para correos electrónicos)

Number (para elegir numeros)

Range (para indicar rangos)

Date (para elegir fechas de un calendario)

Month (para elegir un mes)

Week (para elegir una semana del año)

Time (para elegir un horario)

Datetime (para elegir fecha y hora)

Color (para elegir un color)

Veamos cuál es la sintaxis y usos de cada uno de ellos.

Search (para búsquedas)

Este tipo de input es el indicado para un formulario donde el usuario escribirá un texto que se enviará hacia un buscador (interno o externo a nuestro sitio).

La sintaxis es:

```
<input type="search">
```

Principio del formulario

Buscar:

Final del formulario

Entre las funcionalidades que incorpora este tipo de campo ideales para pantallas touch podemos mencionar que agrega al final del campo un botón en forma de "x" para "limpiar" el contenido del campo; y que no permite saltos de línea (si pegamos en el campo un texto de varios párrafos, solo tomará la palabra o frase previa al primer salto de línea, ya que supone que un buscador es para buscar una palabra o frase, no más).

En el caso de que un navegador antiguo no conozca este tipo de input, degrada en un input de tipo "text", por lo que no afecta en nada la posibilidad de ingresar un texto en el campo. Podemos utilizarlo sin riesgos.

Tel (teléfonos)

Se debe utilizar este tipo de input para campos donde solicitamos que el usuario escriba su teléfono (fijo o móvil).

La sintaxis es:

```
<input type="tel">
```

Teléfono:

Se visualiza igual que un input de texto, pero su uso permite que en un móvil aparezca el teclado numérico, facilitando el ingreso del número.

Con respecto a la validación, es imposible unificar un criterio internacional que permita realizar una validación de un número telefónico (la cantidad de caracteres, los separadores –guiones, espacios, corchetes, paréntesis- varían de país en país, por lo que nunca será posible validar la sintaxis de un teléfono de manera automática).

En el caso de que el navegador no lo soporte, degrada en un input de texto, por lo que no existen problemas para usarlo, se podrá insertar el teléfono en cualquier circunstancia.

Además de la semántica que agrega, tiene la ventaja de permitir desde la hoja de estilos seleccionar todos los inputs de este tipo, mediante selectores de valor de atributo, con los que podríamos decorar (por ejemplo, con una imagen de fondo de un teléfono) este tipo de campo, sin necesidad de agregar ni identificadores ni clases.

Url (direcciones web)

Cuando necesitemos solicitar a nuestros usuarios el ingreso de una URL, podemos colocarle este valor de type al input:

```
<input type="url">
```

De esta manera, el navegador validará que se ingrese una URL absoluta, es decir, que comience con un protocolo (típicamente, http:// pero acepta cualquier carácter delante de los dos puntos y la doble barra, lo que la vuelve compatible hacia el futuro con nuevos protocolos que puedan surgir, como ws:// y otros). En caso de error, nos mostrará un mensaje.

En caso de no ser soportado por el navegador, degrada en un input de texto, por lo que también se podrá ingresar la URL sin problemas.

Email (casillas de correo)

Este tipo de input verifica que la casilla ingresada sea válida (utiliza un estándar internacional para verificar la validez: RFC 5322) pero a pesar de eso no garantiza la calidad de la validación (puede ser mejor utilizar un pattern, algo que veremos más adelante).

Su uso es simple:

```
<input type="email">
```

Una variante de este tipo de input es que podemos especificarle el atributo múltiple, y en ese caso, aceptará varias casillas separadas por comas. Al igual que todos los anteriores, degrada en un input de texto en caso de no ser soportado, por lo que no genera ningún problema su uso.

Number (números)

Este tipo de input es para permitir el rápido ingreso o la selección de números, gracias al agregado de un par de flechas (ascendente y descendente) que facilitan mucho el ingreso de datos numéricos, sobre todo en el caso de una pantalla táctil de un móvil. Además, se le puede definir un rango de valores con su mínimo y su máximo permitido.

Su sintaxis es:

```
<input type="number" min="0" max="0" step="10" value="50">
```

Donde min es el valor mínimo que será permitido ingresar, max es el valor más alto permitido, y step es no solo el salto que dará al pulsar las flechas, sino que también limita los valores permitidos (si step es 10, solo permitirá ingresar múltiplos de 10). Desde ya que value es el valor que se mostrará seleccionado de entrada.

No solo permite ingresar dígitos del 0 al 9, sino también punto decimal, signo negativo y exponente en la forma de: -12345.678e+90

Degrada en input de texto en caso de no ser soportado, por lo que no hay inconvenientes en utilizarlo.

Tiene gran utilidad al facilitar el ingreso de números en móviles táctiles, permitiendo elegir el número con un par de toques en vez de tener que escribirlo con el teclado.

Range (slider para deslizar y elegir rangos)

Es similar a number, pero debemos usarlo cuando no es importante la precisión del número ingresado, sino la usabilidad (la facilidad de arrastrar con el dedo en una pantalla táctil).

Su sintaxis es:

```
<input type="range" min="0" max="100" step="10" value="50">
```

Lo que envía hacia el servidor como valor es un número, proporcional al rango avanzado con la perilla del selector dentro del recorrido total definido entre min y max.

>Degrada en un input de texto.

Date (calendario para elegir una fecha)

Se encarga de mostrar un calendario completo, con solo especificar:

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

Puede llevar los atributos min y max para definir un rango permitido de fechas, fuera del cual no permitirá ingresar fechas.

Un problema para usuarios de países que no siguen el formato norteamericano de fechas, es que el formato del texto que queda escrito en el input es 2012-09-21 (año, mes y día, en ese orden), lo cual puede dar lugar a errores y confusiones por parte de los usuarios.

El formato que envía al servidor como valor de este campo es ese mismo que acabamos de mostrar (2012-09-21, año, guión, mes, guión, y día). Luego deberemos utilizar PHP o algún otro lenguaje de servidor para descomponer en partes esa cadena de caracteres y poder utilizarla en formatos más legibles para nosotros.

Month (mes)

Similar al anterior, pero al pulsar sobre alguno de los días del mes elegido, selecciona el mes completo. El formato del texto que envía al servidor (y que muestra una vez seleccionada la fecha) es 2012-09 (año, guión, mes). Otra vez, el orden no es familiar para nuestra cultura.

Esto se genera con el código:

```
<input type="month">
```

Soporta los mismos atributos min y max que el resto de elementos de fecha y hora.

Puede ser útil para ingresar mes de vencimiento de una tarjeta de crédito.

Week (semana del año)

Seguramente el menos usado de todos, permite elegir una semana del año:

```
<input type="week">
```

Selecciona un renglón entero (una semana) y posee una columna extra al principio, que indica el número de semana dentro de las semanas del año. Envía al servidor y muestra una vez elegida la semana, el texto en formato:

2012-W34

(año, guión, una "W" y el número de semana).

Time (hora)

Se utiliza para indicar una hora. Su sintaxis es:

```
<input type="time" min="11:30" max="23:30" step="600">
```

Lleva los mismos atributos que date. El valor del step está indicado en segundos (y define de a cuánto saltará al pulsar las flechitas, además de limitar las horas permitidas a las que sean múltiplo del step especificado).

El formato que muestra y envía al servidor esta vez sí sirve en nuestra cultura: 11:59

Datetime (fecha y hora)

Permite seleccionar a la vez una fecha y un horario. El formato que enviará está especificado en formato UTC:

2012-09-21T10:55:59.001-03:00

```
<input type="datetime">
```

Selectores de color

Si en una aplicación web deseamos dar a elegir un color, podemos utilizar este tipo de control.

```
<input type="color">
```

Datalist

Este elemento nuevo se utiliza en combinación con un input de texto (o similar, puede ser de tipo URL, email, etc.), donde se espera que el usuario escriba. Funciona a la manera de un auto-suggest, es decir: a medida que el usuario escribe una letra y se encuentra coincidencias entre lo ingresado por el usuario y algunos de los valores previstos, los muestra desplegados debajo del campo, para que podamos seleccionarlos con el mouse o con el dedo en una pantalla táctil.

Su uso principal es facilitar la carga de valores mayoritarios, es decir, que un 80 o 90 por ciento de los usuarios seguramente elegirá. Por ejemplo, si damos a elegir “bebida para el desayuno”, seguramente muchos elegirán café, otros muchos elegirán té, y así con unas 6 o 7 opciones comunes, mientras que algunos pocos ingresarán alguna otra bebida no incluida en la lista. En este sentido, esta etiqueta reemplaza a un select con options, que a su vez tuviera al lado un campo de texto extra con la opción de ingresar “Otros” datos no contemplados en la lista del Select. Es decir, el usuario no está obligado a elegir una de las opciones previstas, solo sirven de ayuda para facilitar la carga en la mayoría de los casos, pero puede escribirse algo diferente a lo previsto y lo aceptará.

Primero creamos el campo de texto:

```
<input type="text" name="diario" list="sugerencias">
```

Y una vez creado, armamos la lista de options dentro de una etiqueta datalist:

```
<datalist id="sugerencias">  
<option label="La Nación" value="La Nación">  
<option label="Clarín" value="Clarín">  
<option label="Infobae" value="Infobae">  
<option label="Perfil" value="Perfil">  
</datalist>
```

En esa etiqueta, es fundamental que le coloquemos un id (identificador) ya que es el que debemos especificar en el campo de texto anterior dentro del atributo "list" para poder vincular a ese campo con esta lista.

Como vemos, la estructura de cada option no es igual a la de los option de un select: en un select, cada option envuelve el valor visible y puede llevar un value que es el dato que se enviará al servidor como valor de esa opción elegida. En cambio, en los option de un datalist no se envuelve ningún valor, sino que se especifican dos atributos: label y value. Dentro de los label es donde se buscará la coincidencia, pero lo que diga en el value es lo que se enviará al servidor. Desde ya que podemos hacer que ambos sean iguales.