

Validar formularios en HTML5



Atributos básicos de validación

En nuestros campos de entrada de datos, se pueden utilizar ciertos atributos para realizar validaciones sencillas. Algunos de estos atributos ya los hemos visto en apartados anteriores, sin embargo, vamos a comentarlos:

Validar formularios en HTML5



Atributo	Valor	Puede actuar sobre...	Descripción
minlength	<u>número</u>	Campos de texto	Establece la longitud mínima del texto requerida.
maxlength	<u>número</u>	Campos de texto	No permite escribir textos superiores a <u>número</u> caracteres.
min	<u>número</u>	Campos numéricos	Establece el número mínimo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha mínima permitida.
	<u>hora</u>	Campos de hora	Establece la hora mínima permitida.
max	<u>número</u>	Campos numéricos	Establece el número máximo permitido.
	<u>fecha</u>	Campos de fecha	Establece la fecha máxima permitida.
	<u>hora</u>	Campos de hora	Establece la hora máxima permitida.
step	<u>número</u>	Campos numéricos	Establece el salto de números permitido. Por defecto, 1.
	<u>fecha</u>	Campos de fecha	Establece el salto de días permitido. Por defecto, 1.
	<u>hora</u>	Campos de hora	Establece el salto de segundos permitido. Por defecto, 1.
required		Campos en general	Campo obligatorio. Se debe rellenar para enviar formulario.
disabled		Campos en general	Campo desactivado. No se puede modificar. No se envía.
readonly		Campos en general	Campo de sólo lectura. No se puede modificar. Se envía.

Validar formularios en HTML5



Atributos básicos de validación

Con estos atributos, podemos crear validaciones básicas en nuestros campos de entrada de datos, obligando al usuario a rellenar un campo obligatorio, forzando a indicar valores entre un rango numérico o permitiendo sólo textos con un tamaño específico, entre otros.

Además, utilizando las **pseudoclases** CSS de validación **:valid** e **:invalid** podemos aplicar estilos a los campos `<input>` y `<textarea>` teniendo en cuenta su validación.

HTML

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Nombre de usuario. Campo obligatorio, entre 5-40 caracteres -->
  Usuario:
  <input type="text" name="nombre" placeholder="Por ejemplo, Juan" minlength="5" maxlength="40" required>
  <!-- Contraseña. Campo obligatorio, mínimo 11 caracteres -->
  Password: <input type="password" name="pass" placeholder="Contraseña" minlength="11" required>
</form>
```

CSS

```
input:valid, textarea:valid { background:green;}
input:invalid, textarea:invalid { background:red;}
```

Validar formularios en HTML5

patrones: pattern



Patrones de validación HTML5

No obstante, aunque los atributos de validación básicos son muy interesantes y pueden facilitarnos la tarea de validación, en muchos casos son insuficientes. Para ello tenemos los patrones de validación HTML5, mucho más potentes y flexibles, que nos permitirán ser mucho más específicos utilizando expresiones regulares para validar datos.

Una expresión regular es una cadena de texto que representa un posible patrón de coincidencias, que aplicaremos mediante el atributo pattern en los campos que queramos validar.

Para ello hay que conocer algunas características básicas de las expresiones regulares:

Validar formularios en HTML5

patrones: **pattern**



Ejemplos de patrones HTML5

Para clarificar estos conceptos y entenderlos mejor, vamos a mostrar algunos ejemplos de campos con validaciones HTML5 en casos reales, y así afianzar conceptos sobre expresiones regulares y su aplicación en los atributos **pattern**:

- Tipo de campo: Nombre de usuario
- Campo obligatorio: required.
- Entre 5-40 caracteres: minlength="5" maxlength="40"
- Sólo se permiten letras (mayúsculas y minúsculas) y números:
pattern="[A-Za-z0-9]+"

```
<form name="formulario" method="post" action="http://pagina.com/send.php">  
  <!-- Nombre de usuario. Campo obligatorio, entre 5-40 caracteres y  
  Sólo se permiten letras y números -->  
  <input type="text" name="nombre" placeholder="Su nombre de usuario"  
    minlength="5" maxlength="40" required pattern="[A-Za-z0-9]{5,40}"  
    title="Letras y números. Tamaño mínimo: 5. Tamaño máximo: 40">  
</form>
```

Validar formularios en HTML5

patrones: pattern



Expresión regular	Carácter especial	Denominación	Descripción
.	Punto	Comodín	Cualquier carácter (o texto de tamaño 1)
A B	<i>Pipe</i>	Opciones lógicas	Opciones alternativas (o A o B)
C (A B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB)
[0-9]	Corchetes	Rangos de caracteres	Un dígito (del 0 al 9)
[A-Z]			Una letra mayúscula de la A a la Z
[^A-Z]	^ en corchetes	Rango de exclusión	Una letra que no sea mayúscula de la A a la Z
[0-9] *	Asterisco	Cierre o clausura	Un dígito repetido 0 ó más veces (vacío incluido)
[0-9] +	Signo más	Cierre positivo	Un dígito repetido 1 ó más veces
[0-9] { 3 }	Llaves	Coincidencia exacta	Cifra de 3 dígitos (dígito repetido 3 veces)
[0-9] { 2, 4 }		Coincidencia (rango)	Cifra de 2 a 4 dígitos (rep. de 2 a 4 veces)
b ?	Interrogación	Carácter opcional	El carácter b puede aparecer o puede que no
\ .	Barra invertida	Escape	El carácter . literalmente (no como comodín)

Validar formularios en HTML5



En el ejemplo anterior, no se limitará al usuario a la hora de escribir, como hace `maxlength`, sino que permitirá al usuario escribir la información que desee y en caso de no pasar la validación, mostrará un mensaje de advertencia y no lo dejará continuar hasta que termine. Además, hemos ampliado el mensaje de advertencia incluyendo el texto en el atributo `title`.

A screenshot of a web form. It features a text input field containing the text "Mar-o\$". To the right of the input field is a button labeled "Enviar". Below the input field, a validation error message is displayed in a light gray box with a thin border. The message starts with an orange exclamation mark icon, followed by the text "Utiliza un formato que coincida con el solicitado". Below this, it specifies "Letras y números. Tamaño mínimo: 5." and "Tamaño máximo: 40".

Validar formularios en HTML5



En el siguiente caso, se pide al usuario que indique el modelo de coche que posee, en un posible formulario de servicio técnico. Los modelos posibles son A1, A3, A4 y A15. En lugar de mostrar una lista de selección, podemos mostrar un campo de texto y colocar una validación como la siguiente:

- Tipo de campo: Modelo de coche
- Campo obligatorio: required.
- Sólo se permiten las opciones: A1, A3, A4 y A15

```
<form name="formulario" method="post" action="http://pagina.com/send.php">
  <!-- Modelo de coche Campo obligatorio, opciones posibles: A1, A3, A4 y A15
       Sólo se permiten letras y números -->
  <input type="text" name="coche" placeholder="Su modelo de coche"
    required pattern="A(1|3|4|15)|a(1|3|4|15)" title="Modelos posibles: A1, A3,
A4 y A15">
</form>
```


Envío de formularios por mail



Si queremos incluir un formulario de contacto en nuestra página web tenemos que tener en cuenta que el atributo **action** trabaja con el lenguaje de programación PHP, por lo tanto:

- Sólo funcionará si nuestra web está alojada en un servidor que lo soporte (tenéis que ver las características de vuestro alojamiento).
- Si trabajáis en un servidor local (en vuestra máquina con Wampserver o XAMPP), tenéis que tener instalado un servidor smtp, para que envíe el email.
- Necesitáis tener unos conocimientos mínimos de PHP. Al menos, saber que el archivo en el que queráis que se ejecute el php tiene que tener extensión .php (no .html), por ejemplo: formulario.php.

Envío de formularios por mail



Teniendo en cuenta estos tres puntos básicos, ya podemos ver el código completo de nuestro archivo de formulario HTML y del envío del formulario PHP:

Nombre:

Teléfono:

Email:

Mensaje:

```
<?php
$nombre = $_POST['nombre'];
$mail = $_POST['mail'];
$empresa = $_POST['empresa'];

$header = 'From: ' . $mail . " \r\n";
$header .= "X-Mailer: PHP/" . phpversion() . " \r\n";
$header .= "Mime-Version: 1.0 \r\n";
$header .= "Content-Type: text/plain";

$mensaje = "Este mensaje fue enviado por " . $nombre . ",
de la empresa " . $empresa . " \r\n";
$mensaje .= "Su e-mail es: " . $mail . " \r\n";
$mensaje .= "Mensaje: " . $_POST['mensaje'] . " \r\n";
$mensaje .= "Enviado el " . date('d/m/Y', time());

$para = 'santiagomartindejesus@gmail.com';
$asunto = 'Datos del formulario: ';

mail($para, $asunto, utf8_decode($mensaje), $header);

echo 'El mensaje fue enviado correctamente';

//header('Location: http://mardeasa.es');
?>
```

Envío de formularios por mail



Un poco de explicación:

El atributo action de la etiqueta form indica a donde tiene que dirigirse a la hora de enviar el formulario, con el código puesto simplemente le indicamos que recargue la página.

Con la variable \$_POST recogemos los datos del formulario que escribió el usuario y los enviamos, el nombre de cada variable corresponde con el name de cada input.

La **función MAIL** es la encargada de, con todos los datos, enviar el formulario, las variables que necesita son:

- Email de destino
- Asunto del mensaje
- Mensaje
- Remitente

Con esto ya debería funcionar, tenéis que acordaros de cambiar el email de la variable destino con el email al que queráis que lleguen los mensajes.

Algunos servidores pueden reconocer como spam los correos enviados mediante la función MAIL, lo que impide que lleguen al destino.