Validación de datos HTML

Uso de los atributos HTML en la validación de datos

Al crear un formulario en HTML, debemos ser conscientes de un detalle ineludible: **los usuarios se equivocan al rellenar un formulario**. Ya sea por equivocación del usuario, ambigüedad del formulario, o error del creador del formulario, el caso es que debemos estar preparados y anticiparnos a estos errores, para intentar que los datos lleguen correctamente a su destino y evitar cualquier tipo de moderación o revisión posterior.

Para evitar estos casos, se suele recurrir a un tipo de proceso automático llamado **validación**, en el cuál, establecemos unas pautas para que si el usuario introduce alguna información incorrecta, deba modificarla o en caso contrario no podrá continuar ni enviar el formulario correctamente.

Uso de los atributos HTML en la validación de datos

Cada vez que creamos un formulario, la **validación** de los datos introducidos estará situada en uno de los siguientes casos (*colocados de peor a mejor*):

1. En este primer caso, **el formulario no tiene validación** de ningún tipo. El usuario puede escribir la información y el sistema no comprobará los datos, ni realizará ningún tipo de validación. Es el peor escenario posible, puesto que el usuario podría enviar desde información incorrecta, hasta datos malintencionados que podrían comprometer la seguridad de la página.

2. Otro caso podría ser que **el formulario tiene validación sólo en el front-end** (*cliente*). De esta forma, los datos son verificados en el navegador del usuario antes de enviarse, pero carecen de validación en el **back-end**, por lo que un usuario malintencionado podría eliminar la validación del front-end y saltársela, enviando datos malintencionados que comprometan la seguridad de la página.

Uso de los atributos HTML en la validación de datos

3. El tercer caso posible es uno donde **el formulario tiene validación sólo en el back-end**. De esta forma, garantizamos que un usuario malintencionado no podrá eliminar el proceso de validación, y los datos siempre se comprobarán. Sin embargo, la desventaja de este método es que el usuario puede rellenar un formulario y es necesario que lo envíe (*con la tardanza que eso puede acarrear*), se procese en el back-end y al devolver un error, el usuario tenga que retroceder al formulario y en algunos casos, incluso tener que volver a rellenar todos los campos de nuevo.

4. Por último, tendríamos el **caso ideal**, donde el formulario **tiene validación en el front-end y en el back-end**, también denominado **doble validación**. En este caso, el formulario es sometido a un proceso de validación en la parte del front-end, y si lo supera, vuelve a pasar otro proceso de validación en el back-end. La desventaja de este método es que conlleva más trabajo de validación, pero es el sistema recomendado, puesto que es más estricto y sobre todo, más seguro.

Uso de los atributos HTML en la validación de datos

•**required:** Indica que un campo debe ser completado antes de enviar el formulario

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

•**pattern:** Especifica un patrón que el valor introducido en un campo debe cumplir

```
<input type="text" name="codigo_postal" pattern="[0-9]{5}"
title="Introduzca un código postal válido de 5 dígitos">
```

•**min:** Especifica el valor mínimo permitido para un campo numérico

```
<input type="number" name="edad" min="18">
```

•**max:** Especifica el valor máximo permitido para un campo numérico

```
<input type="number" name="edad" max="99">
```

Uso de los atributos HTML en la validación de datos

- **minlength:** Especifica el número mínimo de caracteres permitido para un campo de texto

<input type="text" name="nombre" minlength="3">

- **maxlength:** Especifica el número mínimo de caracteres permitido para un campo de texto

<input type="text" name="apellido" maxlength="50">

- **step:** Especifica el intervalo entre valores numéricos permitidos para un campo numérico

<input type="number" name="precio" step="0.01">

Patrones de validación HTML5

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.

Patrones de validación HTML5

Expresión regular	Carácter especial	Significado	Descripción
.	Punto	Comodín	Cualquier carácter (o texto de tamaño 1)
A|B	Pipe	Opciones lógicas	Opciones alternativas (o A o B)
C(A|B)	Paréntesis	Agrupaciones	Agrupaciones alternativas (o CA o CB)
[0-9]	Corchetes	Rangos de carácteres	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)

Uso de '*pattern*' en la validación de datos

- Patrón de código postal de 5 dígitos:

```
<input type="text" pattern="\d{5}" title="Introduzca un código postal válido de 5 dígitos" required>
```

- Patrón de número de teléfono en formato internacional:

```
<input type="tel" pattern="^\+(?:[0-9] ?){6,14}[0-9]$" title="Introduzca un número de teléfono válido en formato internacional" required>
```

- Patrón de contraseña que requiere al menos 8 caracteres, una letra mayúscula, una letra minúscula y un número:

```
<input type="password" pattern="^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).{8,}$" title="La contraseña debe tener al menos 8 caracteres, una letra mayúscula, una letra minúscula y un número" required>
```

Uso de '*pattern*' en la validación de datos

- Patrón de dirección de correo electrónico:

```
<input type="email" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$"
title="Introduzca una dirección de correo electrónico válida" required>
```

- Patrón de número de tarjeta de crédito Visa:

```
<input type="text" pattern="^4[0-9]{12}(?:[0-9]{3})?$" title="Introduzca un
número de tarjeta de crédito Visa válido" required>
```

- Patrón de contraseña que requiere al menos 8 caracteres, una letra mayúscula, una letra minúscula y un número:

```
<input type="password" pattern="^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).{8,}$" title="La
contraseña debe tener al menos 8 caracteres, una letra mayúscula, una letra
minúscula y un número" required>
```

Uso de '*pattern*' en la validación de datos

^[A-Za-z]+$

Este patrón solo permite letras mayúsculas y minúsculas en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de nombres de usuario, nombres de personas, nombres de lugares, etc.

^\d+$

Este patrón solo permite números en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de edades, números de teléfono, códigos postales, números de identificación, etc.

^[A-Za-z0-9]+$

Este patrón permite letras mayúsculas y minúsculas, así como números en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de contraseñas, nombres de usuario, etc.

Uso de '*pattern*' en la validación de datos

^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$

Este patrón es específico para validar direcciones de correo electrónico en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de una cadena de texto que se ajusta a la estructura de una dirección de correo electrónico válida.

^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[^\w\d\s:])([^\s]){8,}$

Este patrón es un ejemplo de un patrón de validación de contraseña seguro. Su semántica se refiere a la validación de contraseñas que contienen al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.

^.{10,}$

Este patrón permite cualquier carácter y requiere al menos 10 caracteres de longitud en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de campos de descripción, comentarios, etc

Uso de '*pattern*' en la validación de datos

^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$

Este patrón es específico para validar direcciones de correo electrónico en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de una cadena de texto que se ajusta a la estructura de una dirección de correo electrónico válida.

^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[^\w\d\s:])([^\s]){8,}$

Este patrón es un ejemplo de un patrón de validación de contraseña seguro. Su semántica se refiere a la validación de contraseñas que contienen al menos una letra mayúscula, una letra minúscula, un número y un carácter especial.

^.{10,}$

Este patrón permite cualquier carácter y requiere al menos 10 caracteres de longitud en el campo de formulario en el que se aplica. Su semántica se refiere a la validación de campos de descripción, comentarios, etc

Ejemplo de uso de 'pattern'

```
<input type="password" pattern="^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[^\w\d\s:])([^\s]){8,}$"
title="Introduzca una contraseña segura con al menos una letra mayúscula, una letra minúscula, un
número y un carácter especial" required>
```

^: Indica el comienzo de la cadena a validar.

(?=.*\d): Es una expresión de "búsqueda positiva anticipada" que indica que la cadena debe contener al menos un dígito (\d), que puede estar en cualquier parte de la cadena.

(?=.*[a-z]): Es otra expresión de búsqueda positiva anticipada que indica que la cadena debe contener al menos una letra minúscula ([a-z]), que también puede estar en cualquier parte de la cadena.

(?=.*[A-Z]): Es otra expresión de búsqueda positiva anticipada que indica que la cadena debe contener al menos una letra mayúscula ([A-Z]), que también puede estar en cualquier parte de la cadena.

(?=.*[^\w\d\s:]): Es otra expresión de búsqueda positiva anticipada que indica que la cadena debe contener al menos un carácter especial ([^\w\d\s:]) que no sea un espacio en blanco (\s) o un carácter alfanumérico (\w).

([^\s]){8,}: Es la expresión regular que indica que la cadena debe tener al menos 8 caracteres ({8,}) y que no debe contener espacios en blanco ([^\s]).

$: Indica el final de la cadena a validar.