



05-06-2020 ~4m Alejandro Mur

Cómo validar un formulario con CSS (sin JavaScript)

► Tabla de contenido

Validar formularios es una de las tareas más aburridas a las que nos solemos enfrentar como desarrolladores web, quizá porque es algo muy repetitivo o quizá porque los formularios son aburridos. **Son muy útiles pero muy aburridos.** Esto lo sabe todo el mundo. Lo sabe el usuario que lo tiene que rellenar y lo sabe el desarrollador que lo va a validar. De hecho lo saben los equipos de Marketing y por eso piden a UX encontrar fórmulas más flexibles para enganchar al usuario, como por ejemplo sólo solicitar un email para empezar a funcionar. El resto vendrá después.

Validar en el front

Lo primero que hay que aclarar es que **una validación correcta se ha de realizar en el front y en el back** porque son complementarias. Utilizaremos el back para garantizar que los datos son los esperados y no comprometer la seguridad de nuestro sistema y el front como herramienta al servicio del usuario para ayudarle a introducir los datos evitando que cometa errores, además le informaremos si alguno de los campos no cumple con el formato esperado (por ejemplo una fecha) y en definitiva buscaremos la manera de no ahuyentarlo.

Construir y validar un formulario con HTML

Con HTML podemos crear formularios autoexplicativos haciendo uso de etiquetas como `<fieldset>`, `<legend>` o `<label>`, y de atributos `[placeholder]`, `[autocomplete]` o `[aria-label]`. Pero además podemos crear reglas de validación **sin utilizar JavaScript** usando los atributos `[required]`, `[pattern]`, `[min]`, `[max]`, `[minlength]`, `[maxlength]`, etc.

Con todo esto tenemos un formulario fácil de entender y lo mejor de todo es que podemos delegar la validación en el navegador. Éste evaluará si la información introducida es correcta o no y mostrará un mensaje acorde a las reglas que hayamos establecido si no se cumple lo esperado, eso sí, el estilo no podemos modificarlo, así que tendremos que conformarnos.

Chrome

Nombre *

alejandro

Fecha de nacimiento *

unemail|



Utiliza un formato que coincida con el solicitado

Formato DD/MM/AAAA

Enviar

Firefox

Nombre *

alejandro

Fecha de nacimiento *

unemail

Ajústese al formato solicitado: Formato DD/MM/AAAA.

Enviar

Safari

Nombre *

alejandro

Fecha de nacimiento

unemail

Usa el formato solicitado

Enviar

Entonces no se valida con CSS

Pues es verdad, no se puede decir que se pueda validar un formulario con CSS pero lo que sí podemos hacer es aplicar estilos a los campos para que la validación sea pintona, porque puntera ya lo es. Es decir, podemos modificar la apariencia de los campos sobre los que se está interactuando o se ha interactuado y que el usuario tenga la sensación de que está tratando con un formulario reactivo construido con el último framework de moda.

Cómo lo conseguimos

En realidad sólo necesitamos saber 4 cosas:

- Qué campos son requeridos
- Si el [placeholder] no es visible (**ALERTA DE SPOILER:** el usuario ha introducido datos)
- Si los datos son válidos
- Si los datos son inválidos

```
input {  
  // Placeholder  
  &::placeholder {  
    color: $c-placeholder;  
  }  
  
  // Focus  
  &:focus:required {  
    border-color: $c-primary;  
    outline: 0 none;  
  }  
  
  // States  
  &:required:placeholder-shown {  
    border: 1px solid $c-border;  
  }  
  &:required:not(:placeholder-shown):invalid {
```

```
border: 1px solid $c-error;
color: $c-error;
box-shadow: 0 0 0 1px;
}
&:required:not(:placeholder-shown):valid {
border: 1px solid $c-success;
color: $c-success;
box-shadow: 0 0 0 1px;
}
&:required:focus:invalid {
border: 1px solid $c-error;
color: $c-error;
box-shadow: 0 0 0 1px;
}
}
```

Para **actuar sobre los campos requeridos** utilizaremos el selector de atributo `:required` porque no queremos afectar a los campos libres. Ahora bien, tenemos que localizar los campos que aún no han sido rellenados para dar un estilo por defecto al borde ya que al estar **vacío el resultado de la validación será incorrecto** y se aplicarían los estilos para marcar los errores. A continuación creamos estilos para los campos requeridos una vez rellenados y sean válidos o inválidos con `:valid` y `:invalid`. Y por último, aprovecharemos que tras la validación si hay campos incorrectos **se devuelve el foco** al primero de estos elementos, creamos reglas para marcar como error el campo requerido con foco e inválido.

Como los campos se evalúan constantemente al introducir cambios en ellos podemos pasar de un estado inválido y marcado como tal a uno válido con un efecto suave e interesante sin tener que estar escuchando eventos con JavaScript

Demo

En esta **demo** puedes ver un formulario de contacto sencillo validado con HTML5 y tuneado con CSS

Conclusión

Somos víctimas de las modas. Suele ocurrir que cargamos nuestras aplicaciones con montones de librerías o que las montamos con frameworks que son estupendos para hacer cosas complejas o para escribir menos, pero pierden sentido cuando son utilizadas para resolver problemas sencillos o existe una solución nativa que puede funcionar.

Somos víctimas de las modas porque seguimos las tendencias de las capas que envuelven la base de la tecnología con la que trabajamos pero no seguimos la evolución de esa base.

Al igual que al inicio de un proyecto valoramos si necesitamos una arquitectura compleja o un framework, cabe preguntarse si necesitamos una librería más para realizar una tarea que se puede completar sin ella, sólo utilizando las herramientas nativas. En el caso de los formularios está claro que hay formularios y formularios y también hay interfaces que te exigen más trabajo y una solución nativa no va a funcionar pero en otros casos sí, y es increíble que con 3 selectores podamos igualar el comportamiento de algunas librerías de JavaScript. Entonces, ¿no es más sencillo este enfoque?

0 Comentarios

 Acceder ▼

G

Sé el primero en comentar...

INICIAR SESIÓN CON

O REGISTRARSE CON DISQUS 

Nombre



Comparte

Mejores

Más recientes

Más antiguos

Sé el primero en comentar.

Suscríbete

Política de Privacidad

No vendan mis datos

Si te ha parecido interesante

Tanto si tienes alguna duda o quieres charlar sobre este tema, como si el contenido o nuestros perfiles te parecen interesantes y crees que pdemos hacer algo juntos, no dudes en ponerte en contacto con nosotros a través de [twitter](#) o en el email hola@mamutlove.com

[RSS](#)

hola@mamutlove.com

[@mamutlove.com](#)

© Mamutlove 2024