

estado de validez

La **ValidityState** interfaz representa los *estados de validez* en los que puede estar un elemento, con respecto a la validación de restricciones. Juntos, ayudan a explicar por qué el valor de un elemento no se valida, si no es válido.

Propiedades

Para cada una de estas propiedades booleanas, un valor de `true` indica que la razón especificada por la que la validación puede haber fallado es verdadera, con la excepción de la `valid` propiedad, que es `true` si el valor del elemento obedece todas las restricciones.

[badInput](#)

Solo lectura

Un valor booleano que es `true` si el usuario ha proporcionado una entrada que el navegador no puede convertir.

[customError](#)

Solo lectura

Un valor booleano que indica si el mensaje de validez personalizado del elemento se ha establecido en una cadena no vacía llamando al [setCustomValidity\(\)](#) método del elemento.

[patternMismatch](#)

Solo lectura

Un valor booleano que es `true` si el valor no coincide con el especificado [pattern](#) y `false` si coincide. Si `true`, el elemento coincide con la [:invalid](#) pseudoclase CSS.

[rangeOverflow](#)

Solo lectura

Un valor booleano que es `true` si el valor es mayor que el máximo especificado por el [max](#) atributo, o `false` si es menor o igual que el máximo. Si `true`, el elemento coincide con las pseudoclases [:invalid](#) and [:out-of-range](#) y CSS.

[rangeUnderflow](#)

Solo lectura

A boolean value that is `true` if the value is less than the minimum specified by the [min](#) attribute, or `false` if it is greater than or equal to the minimum. If `true`, the element matches the [:invalid](#) and [:out-of-range](#) CSS pseudo-classes.

[stepMismatch](#)

Read only

A boolean value that is `true` if the value does not fit the rules determined by the [step](#) attribute (that is, it's not evenly divisible by the step value), or `false` if it does fit the step rule. If `true`, the element matches the [:invalid](#) and [:out-of-range](#) CSS pseudo-classes.

[tooLong](#)

Read only

A boolean value that is `true` if the value exceeds the specified `maxLength` for [HTMLInputElement](#) or [HTMLTextAreaElement](#) objects, or `false` if its length is less than or equal to the maximum length. *Note: This property is never `true` in Gecko, because elements' values are prevented from being longer than `maxLength`.* If `true`, the element matches the [:invalid](#) and [:out-of-range](#) CSS pseudo-classes.

[tooShort](#)

Read only

A boolean value that is `true` if the value fails to meet the specified `minLength` for [HTMLInputElement](#) or [HTMLTextAreaElement](#) objects, or `false` if its length is greater than or equal to the minimum length. If `true`, the element matches the [:invalid](#) and [:out-of-range](#) CSS pseudo-classes.

[typeMismatch](#)

Read only

A boolean value that is `true` if the value is not in the required syntax (when `type` is `email` or `url`), or `false` if the syntax is correct. If `true`, the element matches the `:invalid` CSS pseudo-class.

`valid` Read only

A boolean value that is `true` if the element meets all its validation constraints, and is therefore considered to be valid, or `false` if it fails any constraint. If `true`, the element matches the `:valid` CSS pseudo-class; the `:invalid` CSS pseudo-class otherwise.

`valueMissing` Read only

A boolean value that is `true` if the element has a `required` attribute, but no value, or `false` otherwise. If `true`, the element matches the `:invalid` CSS pseudo-class.

Specifications

Specification
HTML Standard # the-constraint-validation-api:validitystate-3

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	
<code>ValidityState</code>	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
badInput	Chrome 25	Edge 12	Firefox 29	Opera 15	Safari 7	Chrome 25 Android	Firefox 64 for Android	<i>,</i>
<code>customError</code>	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
patternMismatch	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
rangeOverflow	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
rangeUnderflow	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
stepMismatch	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	<i>A</i>
tooLong	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 64 for Android	<i>A</i>

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	
tooShort	Chrome 40	Edge 17	Firefox 51	Opera 27	Safari 10	Chrome 40 Android	Firefox 64 for Android	A
typeMismatch	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	A
valid	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	A
valueMissing	Chrome 3	Edge 12	Firefox 4	Opera 12.1	Safari 5	Chrome 18 Android	Firefox 4 for Android	A

Tip: you can click/tap on a cell for more information.

[Full support](#)
[See implementation notes.](#)

See also

- [Guide: Constraint validation](#)
- [Tutorial: Form data validation](#)

Last modified: Sep 9, 2022, [por colaboradores de MDN](#)