Validación Nativa de Formularios en Javascript utilizando el API Constraint Validation

Juan Manuel Valcárcel Sánchez
Juan Bautista Muñoz Ruiz

Tabla de contenidos

Introducción

- Proporciona opciones de personalización que pueden mejorar o modificar la verificación HTML5 estándar. Es controlada por el programador.
- Utilidades:
  - Validación después de la interacción del usuario.
  - Mensajes de error personalizados.
  - Posibilidad de implementar una validación personalizada: Ej: Coincidencia de campos.
- Establece unas propiedades y unos métodos comunes a todos los elementos.

`willValidate`

`validationMessage`

`validity`

checkValidity()

reportValidity()

setCustomValidity()

Propiedades

`willValidate`

- Verdadero si un formulario está siendo validado.
- Falso en caso contrario. `disabled`

```
<div id="one"></div>
<input type="text" id="two" />
<input type="text" id="three" disabled />
<script>
    document.getElementById('one').willValidate; //undefined
    document.getElementById('two').willValidate; //true
    document.getElementById('three').willValidate; //false
</script>
```

`validationMessage`

- Devuelve un mensaje con las restricciones de validación que no cumple.
- Si `willValidate` es falso o el valor del elemento no incumple ninguna restricción, devolverá una cadena vacía.

```
document.getElementById('foo').validationMessage;
//Chrome  --> 'Please fill out this field.'
//Firefox --> 'Please fill out this field.'
//Safari  --> 'value missing'
//IE10    --> 'This is a required field.'
//Opera   --> ''
```

Propiedades

`validity`

- Devuelve un ValidityState con el estado de validez del elemento.
- Cada restricción es un valor booleano.
- Las restricciones que necesitan atributos los obtienen de la etiqueta input.

`type="email"`

```
<input class="form-control form-control-lg" type="email" id="mail"
       name="mail"/>
```

ValidityState	description
`.badInput`	the browser cannot understand the input
`.customError`	a custom validity message has been set
`.patternMismatch`	the value does not match the specified `pattern` attribute
`.rangeOverflow`	the value is greater than the `max` attribute
`.rangeUnderflow`	the value is less than the `min` attribute
`.stepMismatch`	the value does not fit `step` attribute rules
`.tooLong`	the string length is greater than the `maxlength` attribute
`.tooShort`	the string length is less than the `minlength` attribute
`.typeMismatch`	the value is not a valid email or URL
`.valueMissing`	a `required` value is empty

Propiedades

`validity`

- ***patternMismatch***: Devuelve si el valor no coincide con el `pattern` especificado
- ***tooLong***: Devuelve si el valor es más largo que el atributo `maxlength`.
- ***tooShort***: Devuelve si el valor es más corto que el atributo `minlength`.
- ***rangeOverflow***: Devuelve si el valor es mayor que el atributo `max`.
- ***badInput***: Devuelve si el navegador no entiende la entrada.
- ***customError***: Mensaje customizado.

ValidityState	description
`.badInput`	the browser cannot understand the input
`.customError`	a custom validity message has been set
`.patternMismatch`	the value does not match the specified `pattern` attribute
`.rangeOverflow`	the value is greater than the `max` attribute
`.rangeUnderflow`	the value is less than the `min` attribute
`.stepMismatch`	the value does not fit `step` attribute rules
`.tooLong`	the string length is greater than the `maxlength` attribute
`.tooShort`	the string length is less than the `minlength` attribute
`.typeMismatch`	the value is not a valid email or URL
`.valueMissing`	a `required` value is empty

Propiedades

`validity`

- ***rangeUnderflow***: Devuelve si el valor es menor que el atributo `min`.
- ***typeMismatch***: Devuelve si el valor no está en la sintaxis requerida. No cumple el atributo `type` (email o url).
- ***valid:*** Devuelve si el elemento cumple con todas sus restricciones de validación.
- ***valueMissing***: Devuelve si el elemento con atributo `required` está vacío.
- ***stepMismatch***: Devuelve si valor no se adecua a la reglas del atributo `step`.

ValidityState	description
`.badInput`	the browser cannot understand the input
`.customError`	a custom validity message has been set
`.patternMismatch`	the value does not match the specified `pattern` attribute
`.rangeOverflow`	the value is greater than the `max` attribute
`.rangeUnderflow`	the value is less than the `min` attribute
`.stepMismatch`	the value does not fit `step` attribute rules
`.tooLong`	the string length is greater than the `maxlength` attribute
`.tooShort`	the string length is less than the `minlength` attribute
`.typeMismatch`	the value is not a valid email or URL
`.valueMissing`	a `required` value is empty

Métodos

checkValidity()

Solo es verdadero si todos los controles hijos que posee satisfacen las restricciones impuestas.

```
// validate form on submission
function validateForm(e) {

  const form = e.target;

  if (!form.checkValidity()) {

    // form is invalid - cancel submit
    e.preventDefault();
    e.stopImmediatePropagation();

  }

};
```

reportValidity()

Solo es verdadero si como mínimo un control no pasó alguna restricción. Además reporta qué campos son inválidos.

```
// validate form on submission
function validateForm(e) {

    const form = e.target;

    if (form.reportValidity()) {

      // form is invalid - cancel submit
      e.preventDefault();
      e.stopImmediatePropagation();

    }

  };
```

Restricciones personalizadas

setCustomValidity()

```
if (document.getElementById('password1').value != document.getElementById('password2')
    document.getElementById('password1').setCustomValidity('Passwords must match.');
} else {
    document.getElementById('password1').setCustomValidity('');
}
```

Personalización CSS

```
.valid {
    box-shadow: none; /* FF */
    outline: 0;       /* IE 10 */
}
```

```
.invalid .help {
  display: block;
}

.invalid label, .invalid input {
  color: #c00;
  border-color: #c00;
}
```

Conclusiones:

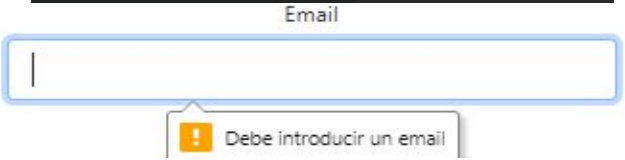
- Ventajas:
  - Mensajes de error personalizados.
  - Restricciones personalizadas.
  - Compatibilidad con la gran mayoría de navegadores.
- Desventajas:
  - Carga de trabajo para el programador.
  - Más sitios donde puede “explotar” el código.
  - Limitaciones en restricciones más complejas
- No es recomendable sustituir la validación en el servidor por la validación en el cliente. Mejor combinarlas.
- Herramientas y tecnologías relacionadas:

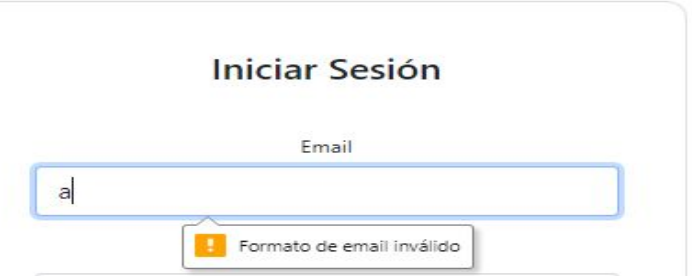
Entidad Usuario: Login

```
const email = document.getElementById( elementId: "mail");
const contr = document.getElementById( elementId: "contr");
const tarjeta = document.getElementById( elementId: "tarjeta");
new *
email.addEventListener( type: "input", listener: (event : Event ) => {
    if (email.validity.valueMissing){
        email.setCustomValidity( error: "Debe introducir un email");
    } else {
        if (email.validity.typeMismatch) {
            email.setCustomValidity( error: "Formato de email inválido");
        } else {
            email.setCustomValidity( error: "");
        }
    }
});
contr.addEventListener( type: "input", listener: (event : Event ) => {
    if (contr.validity.valueMissing) {
        contr.setCustomValidity( error: "Debe introducir una contraseña");
    } else {
        contr.setCustomValidity( error: "");
    }
});
```

Atributos en el <input>:

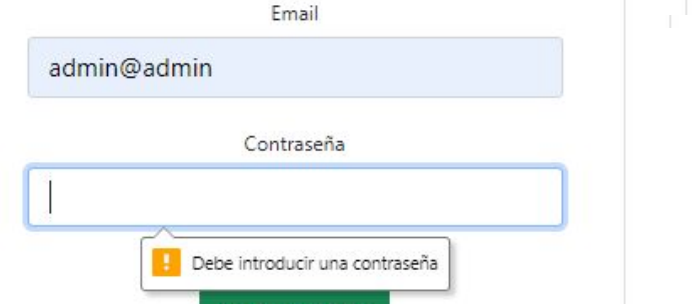
```
type="email" required="true"
```





Atributos en el <input>:

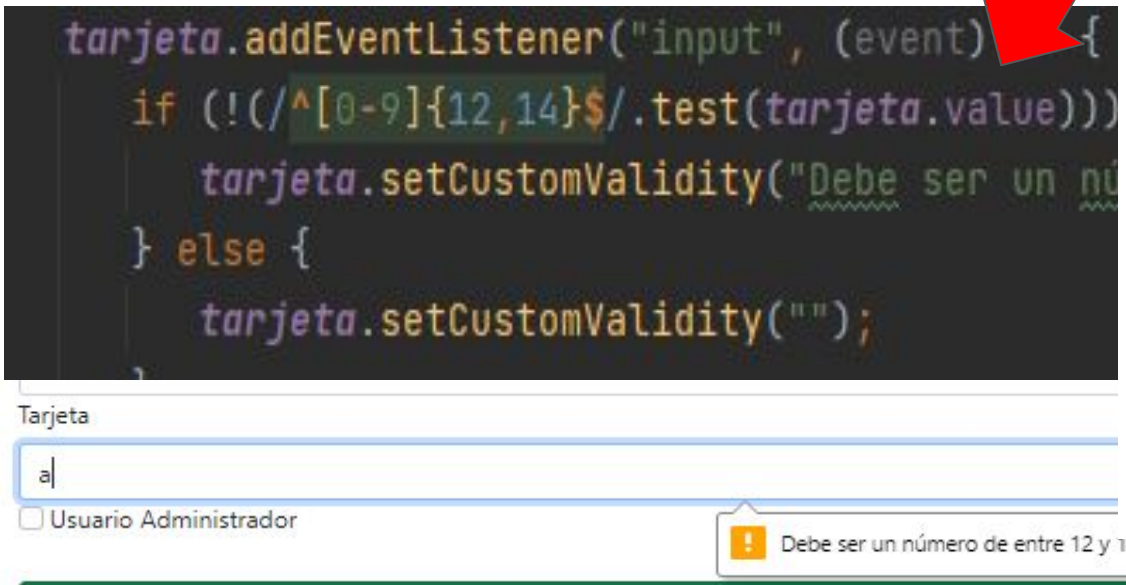
```
required="true"
```



Entidad Usuario: Registro, Insertar y Modificar

La misma validación de la anterior diapositiva (Login), pero incluimos una validación personalizada extra para la tarjeta

```
tarjeta.addEventListener("input", (event) => {
    if (!(/^[0-9]{12,14}$/.test(tarjeta.value))) {
        tarjeta.setCustomValidity("Debe ser un nú
    } else {
        tarjeta.setCustomValidity("");
    }
```



(También se puede realizar esta validación con el atributo pattern y el tarjeta.validity.patternMismatch)

Entidad Producto: Insertar y Modificar

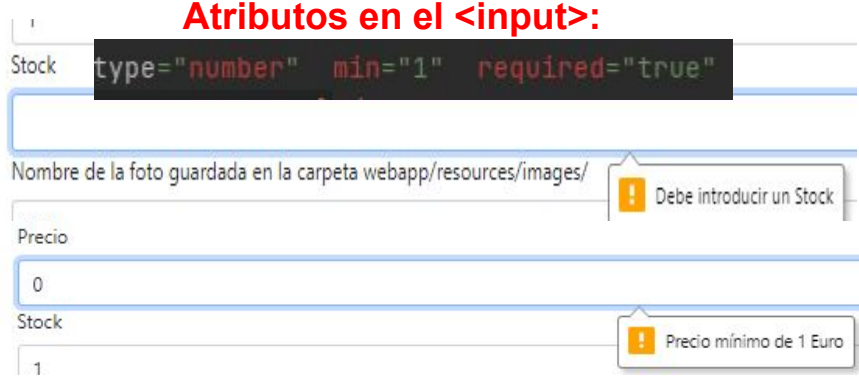
```
nombre.addEventListener("input", (event) => {
    if (nombre.validity.valueMissing){
        nombre.setCustomValidity("Debe introducir un Nombre");
    }else{
        if (nombre.validity.tooShort){
            nombre.setCustomValidity("Logitud minima de 10 carácteres");
        }else{
            nombre.setCustomValidity("");
        }
    }
});
precio.addEventListener("input", (event) => {
    if (precio.validity.valueMissing){
        precio.setCustomValidity("Debe introducir un Precio");
    }else{
        if (precio.validity.rangeUnderflow){
            precio.setCustomValidity("Precio minimo de 1 Euro");
        }else{
            precio.setCustomValidity("");
        }
    }
});
```

Atributos en el <input>:

```
minlength="10" required="true"/>
```

Atributos en el <input>:

```
type="number"  min="1"  required="true"
```



Entidad Venta: Insertar y Modificar

Atributos en el <input>:

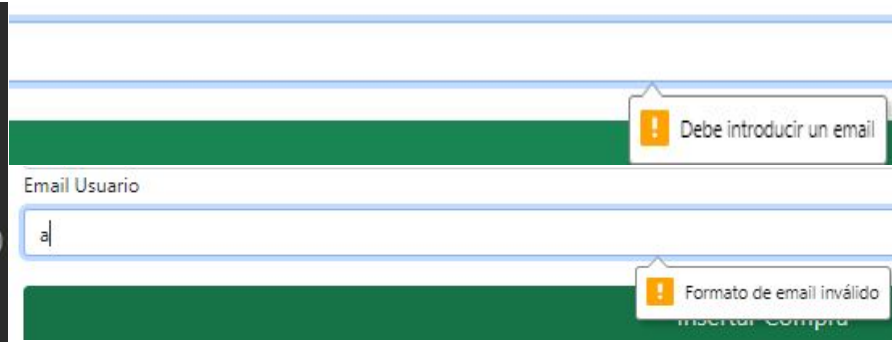
```
type="number"  min="1"  required="true"
```

```
const id = document.getElementById( elementId: "id");
const idProducto = document.getElementById( elementId: "idProducto");
const email = document.getElementById( elementId: "email");
Juan
id.addEventListener( type: "input", listener: (event :Event ) => {
    if (id.validity.valueMissing) {
        id.setCustomValidity( error: "Debe introducir un Id");
    } else {
        if (id.validity.rangeUnderflow) {
            id.setCustomValidity( error: "Debe ser superior a 1");
        } else {
            id.setCustomValidity( error: "");
        }
    }
});
```

Atributos en el <input>:

```
type="email"  required="true"
```

```
email.addEventListener( type: "input", listener: (event :Event ) => {
    if (email.validity.valueMissing) {
        email.setCustomValidity( error: "Debe introducir un email");
    } else {
        if (email.validity.typeMismatch) {
            email.setCustomValidity( error: "Formato de email inválido")
        } else {
            email.setCustomValidity( error: "");
```



Bibliografía

https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation

https://blog.openreplay.com/form-validation-using-javascripts-constraint-validation-api/

https://web.dev/constraintvalidation/