

Artículo que te resume rápidamente la técnica para validar un formulario en el lado del cliente con Javascript. Se comprueba si sus campos son correctos y en caso positivo, se envía el formulario.

## VALIDACIÓN de formularios



Vamos realizar un par de ejemplos completo de validación de formularios. Las validaciones se hacen en el propio navegador, con código Javascript, antes de enviarlo. Si hubo algún campo no rellenado o con información errónea, el formulario muestra el campo que está incorrecto y solicita al usuario que lo cambie. Si todos los datos del formulario son correctos se envía el formulario.

Hemos querido hacer un formulario sencillo, para que el ejercicio no se haga demasiado complicado. No obstante, se realizan validaciones en campos con distintos valores, para hacerlo más variado.

# Alternativas para validar formularios con Javascript

El ejercicio de validar un formulario es uno de los más típicos de Javascript y se puede hacer de diversas maneras, con alternativas de código distintas, que pueden funcionar perfectamente en todos los casos.

Básicamente el proceso es siempre el mismo: acceder campo a campo del formulario y comprobar que los valores introducidos tienen la forma que estamos esperando. En caso que todos los campos de formulario estén correctos, entonces se realiza el envío hacia el servidor donde se tenga que tratar el formulario. La diferencia de enfoques que podemos encontrar en la práctica y que os hemos traído a este artículo varía principalmente en cómo accedemos a los campos del formulario y hacemos en envío.

Hoy vamos a mostrar un par de alternativas de validación de formulario, en las que puedes ver esos distintos modos de trabajo para la validación con Javascript. Son estilos de acceso al DOM un poco diferentes, ya que las costumbres de codificación, así como el estándar de Javascript han ido cambiando con el tiempo. Comenzaremos por un código más actual, que nos gusta más y que es técnicamente un poco más complejo de mayor versatilidad. Luego veremos una alternativa un poco más antigua que se usaba en versiones de Javascript bastante más viejas. Las dos funcionan exactamente igual en la práctica y son capaces de ejecutarse actualmente sin problema en cualquier navegador.

# Validación capturando el evento submit del formulario

En este primer ejemplo vamos a capturar el evento submit del formulario con Javascript, ya que el momento de enviarlo es el adecuado para validar los datos. Realizaremos la validación y luego, si ha ido todo bien, lo enviaremos.

## Código HTML del formulario

Antes de ponernos con la parte del Javascript, veamos el formulario que tenemos:

```
<form action="/login.php" id="formulario">
  <p>
    Usuario: <input type="text" name="usuario"
id="usuario">
  </p>
  <p>
    Clave: <input type="text" name="clave"
id="clave">
  </p>
  <input type="submit" value="Entrar">
</form>
```

Quiero llamar la atención sobre varias cosas:

- El HTML formulario no tiene nada de especial, que lo pueda hacer diferente de cualquier otro formulario donde no exista validación (tiene un botón de submit normal y no hay código Javascript por ningún lado ni eventos declarados en manejadores en el HTML). Este hecho, que pueda parecer solamente anecdótico es importante, porque tal como está el formulario funcionará

perfectamente aunque el navegador tenga deshabilitado Javascript, lo que mejora la accesibilidad. Es la diferencia fundamental con el segundo ejemplo de validación de fomrulario Javascript que vamos a ver después.

- El formulario tiene un campo "id", en la etiqueta FORM, que nos servirá para acceder a él mediante Javascript.
- Todos los campos que necesitamos validar, también tienen su "id".

## El Javascript para la validación del formulario

Antes de pasar a ver el código quiero resumir los pasos que vamos a realizar para esta validación.

1. Definiremos un evento "submit" para capturar el instante en el que el usuario enviará el formulario.
2. Una vez capturado el evento submit del formulario con Javascript, vamos bloquear el envío, evitando que pueda enviar con datos incorrectos.
3. Validaremos todos los campos del formulario.
4. Si todos los campos se validan correctamente, entonces realizaremos el envío nosotros mediante Javascript.

Entendidos los pasos que tenemos que realizar, podemos pasar a ver el código. Consta de dos partes, uno para la definición de los eventos de Javascript necesarios para enganchar nuestros manejadores para procesar el envío del formulario y otra parte en la que realizamos la validación propiamente dicha.

```
document.addEventListener("DOMContentLoaded",  
function() {  
  
document.getElementById("formulario").addEventListener('submit', validarFormulario);
```

```
});

function validarFormulario(evento) {
    evento.preventDefault();
    var usuario =
document.getElementById('usuario').value;
    if(usuario.length == 0) {
        alert('No has escrito nada en el usuario');
        return;
    }
    var clave = document.getElementById('clave').value;
    if (clave.length < 6) {
        alert('La clave no es válida');
        return;
    }
    this.submit();
}
```

Comenzamos definiendo un manejador para el evento "DOMContentLoaded". Este manejador nos asegura que el resto del código que se va a ejecutar se realice cuando verdaderamente el navegador esté listo para ejecutarlo.

Luego realizamos un manejador para el evento "submit" del formulario. Como manejador para este segundo evento indicamos la función "validarFormulario()". Es interesante aquí el acceso al DOM de los elementos de la página, que estamos haciendo con "document.getElementById()" y enviando su identificador (atributo "id") como parámetro. Este es otro de los detalles que distinguen a este código del otro más sencillo que veremos más adelante en este artículo.

La función `validarFormulario()` recibe como parámetro el objeto evento. Esto es algo que ocurre con todos los manejadores de eventos Javascript. Este objeto "evento" nos sirve para hacer diversas cosas, como por ejemplo detener el comportamiento por defecto de un evento. Como se trata del evento `submit`, el comportamiento predeterminado es justamente el envío del formulario, que hemos detenido con la instrucción `"evento.preventDefault()"`.

A continuación vamos realizando el acceso a cada elemento del formulario, para estudiar si son correctos o no. En el caso que no lo sean, mostramos un mensaje en una caja de alerta y luego salimos de la función con `"return"`. Ese `return` permite que, una vez que se ha detectado un error, no se siga ejecutando el código de la función.

Por último, si no se salió por ninguna de las validaciones incorrectas, entonces enviamos el formulario. Esto lo realizamos con el método `submit()` del formulario. Como estamos dentro de un manejador de evento definido por el formulario, la variable `"this"` corresponde con el propio formulario. Por ello, el envío lo podemos realizar con un simple `"this.submit()"`.

Eso es todo. Espero que hayas podido entender las bases sobre esta alternativa para la validación del formulario. Ahora te vamos a presentar una diferente, más básica, que quizás también te guste y con la que podrás seguir aprendiendo.

## Segundo ejemplo de validación

Ahora vamos a ver un ejemplo de validación un poco más sencillo, con un estilo de codificación un poco más antiguo y que actualmente nos gusta menos, pero que puede ayudarte si estás

aprendiendo, para darte cuenta que podemos hacer las cosas mediante formas bastante distintas y obtener el mismo resultado.

En este ejemplo se comprueba un campo donde debe figurar un texto, otro donde debe introducirse un número mayor que 18 y un último con un campo select donde deben haber seleccionado un valor.

Recuerda que en Javascript existen diversas vías para validar formularios. En este ejemplo se han querido hacer las cosas fáciles para personas que están iniciando.

**Referencia:** para comprender este ejercicio hace falta conocer el [trabajo con formularios con Javascript](#). Podemos aprender también Javascript desde cero <http://www.desarrolloweb.com/javascript/>, si es que fuera necesario.

## El código del formulario

A la hora de construir un formulario que se desea validar, se realiza con un HTML igual que un formulario que se fuera a enviar sin validación. En nuestro ejemplo usaremos el DOM para acceder a los campos a partir de la jearquía de objetos, a partir del documento y del objeto formulario. En otros ejemplos en Internet podrás ver que para acceder a los campos se usan identificadores (atributos "id"). El resultado en realidad es el mismo.

El formulario con el que vamos a trabajar es el siguiente:

```
<form name="fvalida">
<table>
```

```

<tr>
    <td>Nombre: </td>
    <td><input type="text" name="nombre" size="30"
maxlength="100"></td>
</tr>
<tr>
    <td>Edad: </td>
    <td><input type="text" name="edad" size="3"
maxlength="2"></td>
</tr>
<tr>
    <td>Interés:</td>
    <td>
        <select name=interes>
            <option value="Elegir">Elegir
            <option value="Comercial">Contacto comercial
            <option value="Clientes">Atención al cliente
            <option value="Proveedores">Contacto de
proveedores
        </select>
    </td>
</tr>
<tr>
    <td colspan="2" align="center"><input
type="button" value="Enviar"
onclick="valida_envia()"></td>
</tr>
</table>
</form>

```

Es un formulario cualquiera. Los únicos puntos donde debemos prestar atención son:



- El nombre del formulario, "fvalida", que utilizaremos para referirnos al él mediante Javascript.
- El botón de enviar, que en lugar de ser un submit corriente, es un botón que llama a una función, que se encarga de validar el formulario y enviarlo si todo fue correcto.

## Función Javascript para validar el formulario

Ahora veremos la función que hemos creado para validar el formulario. Se llama valida\_envia(). Simplemente, para cada campo del formulario, comprueba que el valor introducido es correcto. Si no es correcto, muestra un mensaje de alerta, pone el foco de la aplicación en el campo que ha dado el error y abandona la función retornando el valor 0.

Si todos los campos eran correctos, la función continúa hasta el final, sin salirse, por no estar ningún campo incorrecto. Entonces ejecuta la sentencia última, que es el envío del formulario.

Veamos la función entera, aunque luego la expliquemos por partes.

```
function valida_envia(){
    //valido el nombre
    if (document.fvalida.nombre.value.length==0){
        alert("Tiene que escribir su
nombre")

        document.fvalida.nombre.focus()
        return 0;
    }
```

```
//valido la edad. tiene que ser entero mayor
que 18
edad = document.fvalida.edad.value
edad = validarEntero(edad)
document.fvalida.edad.value=edad
if (edad==""){
    alert("Tiene que introducir un
número entero en su edad.")
    document.fvalida.edad.focus()
    return 0;
}else{
    if (edad<18){
        alert("Debe ser mayor de 18
años.")
        document.fvalida.edad.focus()
        return 0;
    }
}

//valido el interés
if (document.fvalida.interes.selectedIndex==0){
    alert("Debe seleccionar un motivo de
su contacto.")
    document.fvalida.interes.focus()
    return 0;
}

//el formulario se envia
alert("Muchas gracias por enviar el
formulario");
document.fvalida.submit();
}
```

En el primer bloque se valida el campo nombre. La validación que se hace es simplemente si se ha escrito algo en el campo. Para ello comprueba si el número de caracteres escritos en el campo nombre es cero. En ese caso, muestra el mensaje de alerta, sitúa el foco en el campo de texto y se sale de la función devolviendo el valor 0.

**Nota:** el foco de la aplicación es el lugar donde está situado el cursor.

El foco puede estar en cualquier sitio. Por ejemplo en un campo de texto, en un select, en un enlace o en la propia página. Si presionamos una tecla del teclado afecta al lugar donde está situado el foco. Si, por ejemplo, el foco está en un campo de texto, al operar con el teclado estaremos escribiendo en ese campo de texto.

La validación de la edad mayor que 18 años tiene dos partes. Primero debemos comprobar que en el campo de texto hay escrito un valor entero. Luego, si teníamos un entero, habría que comprobar que es mayor que 18. Para hacer esta validación nos vamos a apoyar en una función que ya hemos visto en otro artículo de DesarrolloWeb.com, que habla sobre cómo [validar un entero en un campo de formulario](#). Esa función devuelve un string vacío en caso de que no sea un entero y el propio entero, si es que lo era.

Antes de realizar la validación de la edad propiamente dicha, se obtiene el valor introducido en el campo de formulario edad y se guarda en una variable llamada edad. Luego se ejecuta la función pasando esta edad. Su resultado lo utilizamos para volcarlo otra vez al campo de texto. Entonces, se comprueba si el valor devuelto por la función es un string vacío. En ese caso, es que el valor escrito en el formulario no era un entero, por lo que se muestra el mensaje de error, se sitúa el foco y se sale de la función.

En caso de que el campo edad contuviese un entero, se debe comprobar a continuación si es mayor que 18. En caso de que sea menor, se muestra el error y se sale. En caso contrario - entonces el valor sería mayor o igual que 18-, se continúa con las comprobaciones.

Por último se valida el campo select, donde aparece el interés del supuesto visitante, que le motiva para enviarnos el formulario. En ese campo se debe haber seleccionado cualquier opción menos la primera. Para asegurarnos, simplemente se comprueba si el atributo `selectedIndex` del campo select tiene el valor 0. Ese atributo almacena el índice seleccionado en el menú desplegable. El primer campo tiene el índice 0, el segundo el índice 1...

Si se comprueba que `selectedIndex` vale 0, se muestra un mensaje de alerta, se pone el foco en el campo del formulario y se sale de la función.

Si hemos llegado hasta este punto sin salirnos de la función es que todos los campos del formulario estaban rellenos correctamente. En ese caso se debe enviar el formulario. Antes de enviar el formulario se muestra un mensaje de alerta, agradeciendo que se haya rellenado correctamente. Este mensaje se puede suprimir si se desea.

Para enviar el formulario se hace una llamada al método `submit()` de dicho formulario.

## Conclusión a la validación de formularios con Javascript

Este ejercicio es de lo más básico y útil que se puede hacer en Javascript. Requiere ciertos conocimientos, ya ligeramente avanzados, pero en el fondo no resulta complicado. Incluso ampliarlo es bastante sencillo, siempre que sigamos un esquema similar para cada uno de los campos.

Recuerda que en DesarrolloWeb.com tienes mucho material para complementar y expandir estas explicaciones. Tenemos un manual completo dedicado al [trabajo con formularios](#). Pero también encontrarás muchos otros manuales para hacer validaciones apoyado en librerías y frameworks Javascript avanzados. Te recomendamos acceder a la [sección dedicada a Javascript](#).