APRENDERAPROGRAMAR.COM

JAVASCRIPT VALIDAR CAMPO TEXTO NO VACÍO. QUE VALOR SEA NUMÉRICO. EMAIL O CORREO ELECTRÓNICO EN FORMULARIOS. EJEMPLOS (CU01182E)

Sección: Cursos
Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº82 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

VALIDACIÓN DE FORMULARIOS CON JAVASCRIPT

Un uso habitual e importante de JavaScript es servir para realizar la validación de formularios. Cuando un usuario rellena un formulario y pulsa en el botón enviar, comprobaremos con JavaScript si ha introducido los datos mínimos requeridos y con el formato requerido. Si no es así, abortaremos el envío del formulario y mostraremos los avisos correspondientes evitando enviar la información al servidor.

El envío de la información al servidor supondría una demora en la respuesta (enviar la información, realizar la validación en el servidor, enviar la respuesta…) y daría la impresión de que la página web "está lenta". Con JavaScript podemos lograr una respuesta inmediata, haciendo que la navegación para el usuario de nuestra página web sea satisfactoria.

La validación normalmente consiste en detectar el evento submit del formulario y derivar la gestión del mismo a una función manejadora que se encarga de revisar el contenido del formulario. Si en esa revisión se detectan errores en los formatos o falta de datos, el envío se aborta y se muestra un mensaje de aviso al usuario. Si en la revisión se considera que está todo conforme, se procede al envío del formulario.

La validación no supone más que una aplicación práctica de los conocimientos que hemos adquirido durante el curso (acceso a elementos del DOM, manejo de objetos, expresiones regulares, etc.).

La invocación a la validación se puede enfocar de distintas maneras. La forma quizás preferible es usar addEventListener y preventDefault:

```
document.nombreDelFormulario.addEventListener('submit', validarFormulario);

function validarFormulario(evObject) {
evObject.preventDefault(); //Evita el envío del formulario hasta comprobar
}
```

Otra forma que ha venido siendo bastante utilizada es capturar el evento en el código HTML para que devuelva true (en cuyo caso se enviará el formulario) o false (en cuyo caso no se enviará):

```
function validarFormulario() {
if (… ) { return true; // Si se verifican las condiciones enviar formulario
} else { return false; // Si no se verifican las condiciones no se enviará el formulario
}
---
}
---
<form action="" method="" id="" name="" onsubmit="return validarFormulario ()">
```

VALIDAR QUE UN CAMPO NO ESTÁ VACÍO

En muchos formularios se requerirá que uno o varios campos no se encuentren vacíos. Por ejemplo si se trata de una reserva de hotel, será necesario que datos como el nombre de la persona que realiza la reserva no se encuentren vacíos.

También podemos comprobar que no se hayan introducido sólo espacios en blanco (que son caracteres pero no aportan información), u otro tipo de comprobaciones más elaboradas usando expresiones regulares.

Escribe este código que es un ejemplo de comprobación de que un campo no esté vacío o contenga sólo espacios en blanco y comprueba sus resultados:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css">body {background-color:yellow; font-family: sans-serif;} label{color: maroon; display:block;
padding:5px;}
</style>
<script type="text/javascript">
window.onload = function () {
document.formularioContacto.nombre.focus();
document.formularioContacto.addEventListener('submit', validarFormulario);
}

function validarFormulario(evObject) {
evObject.preventDefault();
var todoCorrecto = true;
var formulario = document.formularioContacto;
for (var i=0; i<formulario.length; i++) {
        if(formulario[i].type =='text') {
                if (formulario[i].value == null || formulario[i].value.length == 0 || /^\s*$/.test(formulario[i].value)){
                alert (formulario[i].name+ ' no puede estar vacío o contener sólo espacios en blanco');
                todoCorrecto=false;
                }
        }
        }
if (todoCorrecto ==true) {formulario.submit();}
}

</script></head>
<body><div id="cabecera"><h1>Portal web aprenderaprogramar.com</h1><h2>Didáctica y divulgación de la
programación</h2>
</div>
<!-- Formulario de contacto -->
<form name ="formularioContacto" class="formularioTipo1" method="get" action="http://aprenderaprogramar.com">
<p>Si quieres contactar con nosotros envíanos este formulario relleno:</p>
<label for="nombre"><span>Nombre:</span> <input id="nombre" type="text" name="nombre" /></label>
<label for="apellidos"><span>Apellidos:</span> <input id="apellidos" type="text" name="apellidos" /></label>
<label for="email"><span>Correo electrónico:</span> <input id="email" type="text" name="email" /></label>
<label><input type="submit" value="Enviar"><input type="reset" value="Cancelar"></label>
</form></body></html>
```

Recordar lo que explicamos en los apartados del curso dedicados a expresiones regulares. /^\s*$/ define una expresión regular donde \s es el espacio en blanco, ^ indica comienzo de cadena con ese carácter, * indica repetición cualquier número de veces y $ señala el carácter de terminación de cadena. Por tanto la expresión regular indica “las cadenas que empiezan con espacio en blanco, siguen con cualquier número de espacios en blanco y terminan con un espacio en blanco”.

El método test devuelve true si la cadena hace match con la expresión regular o false si la cadena no hace match.

VALIDAR QUE UN CAMPO CONTIENE UN VALOR NUMÉRICO

En ocasiones se puede requerir que un campo contenga un valor numérico. Las comprobaciones se pueden hacer basándonos en expresiones regulares, basándonos en la función isNaN, o basándonos en que el valor numérico debe ser mayor, menor, mayor o igual, menor o igual que otro valor numérico de referencia.

Recordar que isNaN(x) devuelve true si x es un valor no asimilable a un número o false si el valor es asimilable a un número. La función isNaN es “laxa” en el sentido de que tratará de asimilar a un número todo lo que sea posible. Por ejemplo alert(isNaN(-3.23)); devuelve false (se considera -3.23 un número). Esto puede ser interesante en algunas ocasiones pero no interesante en otras.

Ejemplo: pedimos la altura respecto al nivel del mar, siendo 0 el nivel del mar, un valor positivo una altura sobre el nivel del mar y un valor negativo una altura por debajo del nivel del mar (por ejemplo la altura de un submario). En este caso nos interesa usar isNaN.

Otro ejemplo: pedimos la edad de una persona. En este caso será preferible usar expresiones regulares, por ejemplo comprobar que el valor del campo se corresponda con uno ó dos dígitos, lo que haría válido 5, 44, 88, pero haría inválidos 148, -3.23 ó 5.25.

Escribe este código que es un ejemplo de validaciones con campos numéricos y comprueba sus resultados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>
<title>Ejemplo aprenderaprogramar.com</title>
<meta charset="utf-8">
<style type="text/css">
body {background-color:yellow; font-family: sans-serif;}
label{color: maroon; display:block; padding:5px;}
</style>
<script type="text/javascript">

window.onload = function () {
document.formularioContacto.nombre.focus();
document.formularioContacto.addEventListener('submit', validarFormulario);
}
```

```
function validarFormulario(evObject) {
evObject.preventDefault();
var todoCorrecto = true;
var formulario = document.formularioContacto;
if (isNaN(formulario.edad.value)==true || /^[1-9]\d$/.test(formulario.edad.value)==false ) {alert ('Edad no valida');
todoCorrecto=false;}
if (isNaN(formulario.altura.value)==true || formulario.altura.value<=0 || formulario.altura.value>=2.50) {
alert ('Altura no valida'); todoCorrecto=false;}
if (todoCorrecto ==true) {formulario.submit();}
}
</script></head>
<body><div id="cabecera"><h1>Portal web aprenderaprogramar.com</h1><h2>Didáctica y divulgación de la
programación</h2></div>
<!-- Formulario de contacto -->
<form name ="formularioContacto" class="formularioTipo1" method="get" action="http://aprenderaprogramar.com">
<p>Si quieres contactar con nosotros envíanos este formulario relleno:</p>
<label for="nombre"><span>Nombre:</span> <input id="nombre" type="text" name="nombre" /></label>
<label for="apellidos"><span>Edad:</span> <input id="edad" type="text" name="edad" /></label>
<label for="email"><span>Altura:</span> <input id="altura" type="text" name="altura" /></label>
<label><input type="submit" value="Enviar"><input type="reset" value="Cancelar"></label>
</form></body></html>
```

El resultado esperado es que el formulario no se envíe si la edad no está formada por dos dígitos (no sería válido 124 ni -3.23) o si la altura es un valor negativo, mayor o igual a 2.50 ó no tiene formato numérico.

VALIDAR UNA DIRECCIÓN DE EMAIL

Para comprobar que el usuario introduce una dirección de email válida podemos hacer distintos tipos de comprobaciones. La más básica es comprobar que la cadena contiene cualquier número de caracteres seguido del carácter @ segudio de cualquier número de caracteres, seguido de un punto, y seguido de cualquier número de caracteres. Esta validación es amplia y no aceptaría la mayor parte de direcciones no válidas, por ejemplo no se aceptaría <<andres@gmailcom>> porque le falta el punto.

En general será suficiente usar una expresión regular simple del tipo:

```
/\S+@\S+\.\S+/
```

En muchas páginas de internet se encuentran expresiones más complejas que tratan de tener en cuenta posibilidades menos frecuentes de formato de correo electrónico, por ejemplo esta:

```
/ [a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-
9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?/
```

Las comprobaciones se harían de la misma forma a como hemos visto en los ejemplos anteriores.

EJERCICIO

Dado el siguiente código HTML que contiene un formulario con tres campos (nombre, apellidos y email):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Ejemplo aprenderaprogramar.com</title>
<meta charset="utf-8">
<style type="text/css">body {background-color:yellow; font-family: sans-serif;}
label{color: maroon; display:block; padding:5px;}
</style>
</head>
<body>
<div id="cabecera"><h1>Portal web aprenderaprogramar.com</h1><h2>Didáctica y divulgación de la programación</h2>
</div>
<!-- Formulario de contacto -->
<form name ="formularioContacto" class="formularioTipo1" method="get" action="http://aprenderaprogramar.com">
<p>Si quieres contactar con nosotros envíanos este formulario relleno:</p>
<label for="nombre"><span>Nombre:</span> <input id="nombre" type="text" name="nombre" /></label>
<label for="apellidos"><span>Apellidos:</span> <input id="apellidos" type="text" name="apellidos" /></label>
<label for="email"><span>Correo electrónico:</span> <input id="email" type="text" name="email" /></label>
<label><input type="submit" value="Enviar"><input type="reset" value="Cancelar"></label>
</form>
</body>
</html>
```

Crear el código JavaScript que cumpla con las siguientes funciones:

a) Si la longitud (número de caracteres) del campo nombre es mayor de 15 o igual a cero, el formulario no se enviará.

b) Si la longitud (número de caracteres) del campo apellidos es mayor de 30 o igual a cero, el formulario no se enviará.

c) Si la longitud (número de caracteres) del campo email es mayor de 35 o igual a cero, el formulario no se enviará. Si el email no contiene el carácter @ el formulario no se enviará.

d) Si se produce cualquiera de las circunstancias anteriores, debe aparecer un recuadro con color de fondo naranja y texto negro a la derecha de la casilla de introducción de datos, informando del problema detectado en ese campo (si es que ese campo presenta algún problema). Nota: estos mensajes se deben mostrar sólo si el campo es erróneo después de pulsado el botón enviar, y deben desaparecer si el usuario realiza un nuevo intento y el campo es correcto. Los mensajes se incorporarán al DOM (no serán mensajes usando alert).

Ejemplo de ejecución. El usuario deja el nombre, apellidos y correo electrónico vacíos. A la derecha de las casillas de introducción de datos aparecerá: El nombre no puede estar vacío. Los apellidos no pueden estar vacíos. El correo electrónico no puede estar vacío.

Ahora el usuario introduce nombre: <<Juan Manuel de Todos los Santos Efímeros Ecuánimes de Todos los días de la Tercera Edad>>. Como apellidos introduce: <<Suárez>>. Y como correo electrónico introduce: <<juanmanueldetodoslosantosefimerosecuanimesdetodoslosdias@gmail.com>>. Pulsa enviar. A la derecha de la casilla de introducción del nombre debe aparecer: "Nombre demasiado largo. Máximo 15 caracteres". A la derecha de apellidos no aparecerá nada porque el apellido es correcto. A la derecha del correo electrónico debe aparecer: "Correo electrónico demasiado largo. Máximo 35 caracteres". Ahora el usuario escribe como nombre: Juan Manuel. Como apellidos: Suárez. Y como correo electrónico juanmanuel@gmail.com. Pulsa enviar y el formulario es enviado.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01183E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206