



Validación básica de formularios con Javascript

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Ene
- iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

———— Índice ————



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Identificar el concepto de BOM con los objetos que lo componen y saber utilizarlo en el desarrollo de aplicaciones web.

GLOSARIO



BOM (Browser Object Model). Objeto en el que se representa el navegador, no solo el documento.

Mediante el BOM se puede acceder a datos como el historial de navegación, dimensiones de la ventana, etcétera.

DOM. Plataforma e interfaz de un documento que permite a los scripts y programas acceder y modificar dinámicamente su contenido, estructura y estilo.

Tag. Término en inglés que significa “etiqueta” y hace referencia a una palabra clave que sirve para describir un documento.

INTRODUCCIÓN



En esta lección estudiamos una de las partes más interesantes y más útiles de la programación con Javascript: **la validación de formularios desde el lado del cliente (navegador).**

En primer lugar hay que tener en cuenta que **la validación con Javascript no garantiza que los datos enviados son correctos.** Es, simplemente, una cuestión de **experiencia de usuario** que evita mandar datos erróneos al servidor y hacer operaciones innecesarias. Pero es posible **deshabilitar el Javascript del navegador** o utilizar la consola del desarrollador para **cambiar estos datos**, por lo que no se puede asegurar su validez.

Por eso, **la seguridad en la validación de un formulario debe realizarse en el lado del servidor y en la base de datos.**

INTRODUCCIÓN



Como se ha comentado anteriormente, la seguridad en la validación de un formulario debe realizarse en el lado del servidor y en la base de datos.

Para ello podríamos realizar esta validación mediante algún lenguaje de servidor como PHP, JSP o ASP.

Recuerda: No es seguro validar un formulario únicamente con #Javascript. La validación en el lado del cliente sólo garantiza la experiencia de usuario. Hay que validar los datos en el servidor y en la base de datos para asegurarnos que son correctos.

INTRODUCCIÓN



Hay que tener en cuenta que es posible validar un formulario del lado del cliente de varias maneras.

Para facilitar la comprensión, se implementa un formulario con nombre, teléfono, fecha de nacimiento, sexo y un checkbox “Mayor de 18”, al cual, validaremos su información únicamente utilizando Javascript y sin tener en cuenta las características de los campos de los formularios de HTML5 o utilizar expresiones regulares.

Adicionalmente, el formulario implementa un botón de tipo `submit` cuya acción es ejecutar el archivo `procesar.php`. Este sería el encargado de validar el formulario en el lado del servidor.

A continuación, se muestra el código HTML del formulario:

FORMULARIO



CÓDIGO HTML

```
<h1>Formulario</h1>
<form action="procesar.php" method="post" id="miFormulario">
  <fieldset>
    <legend>Datos personales</legend>
    <p>Nombre: <input type="text" name="nombre" id="nombre" /></p>
    <p>Teléfono: <input type="text" name="telefono" id="telefono" /></p>
    <p>Fecha de nacimiento:
      <input type="text" name="dia" size="2" id="dia" /> <b>/</b>
      <input type="text" name="mes" size="2" id="mes" /> <b>/</b>
      <input type="text" name="ano" size="4" id="ano" />
    </p>
```


FORMULARIO



CÓDIGO HTML

```
<p>Sexo :  
  <input type="radio" name="sexo" value="H" checked /> Hombre  
  <input type="radio" name="sexo" value="M" /> Mujer  
</p>  
<p>Mayor de 18: <input type="checkbox" name="mayor" id="mayor" checked  
</p>  
</fieldset>  
<p>  
  <input type="submit" value="Enviar" id="enviar" />  
  <input type="reset" value="Borrar" id="borrar" />  
</p>  
</form>
```

VALIDACIÓN DEL FORMULARIO

Acceso al elemento formulario



Por último, vinculamos el documento HTML con un fichero JS donde se implementará la lógica de validación del formulario mediante:

```
<script src="validacion.js"></script>
```

A continuación, se desarrolla la lógica de validación en dicho fichero JS externo.

En primer lugar, seleccionamos el formulario obteniendo el elemento HTML mediante su identificador, para poder acceder a sus campos.

```
let formulario = document.getElementById("miFormulario");
```

VALIDACIÓN DEL FORMULARIO

Acceso al elemento formulario



Otra alternativa de selección mediante identificador, sería haciendo uso de la propiedad forms del objeto `Document`.

```
let formulario2 = document.forms["miFormulario"];
```

También sería posible seleccionar el formulario sabiendo su número (ubicación) en la página web.

```
let formulario3 = document.getElementsByTagName("form")[0];  
//Seleccionamos todos los elementos de tipo form y nos quedamos con el primero  
let formulario4 = document.forms[0];
```

VALIDACIÓN DEL FORMULARIO

Acceso a los elementos del formulario



A continuación, accedemos a los elementos del formulario en sí. Para ello emplearemos los siguientes métodos:

Recuerda: El valor del atributo **id** es único en el documento, por lo que hasta ahora se empleaba el método **getElementById()** para acceder al elemento deseado mediante este atributo.

- `.elements[]` : Devuelve un *array* con todos los **input** del formulario
- `getElementById("idElemento")` : Devuelve un elemento con un **id** determinado
- `getElementsByTagName("etiqueta")` : Devuelve un *array* con elementos de un tipo de etiqueta (**input**, **select**, etc.)
- `getElementsByName("nombre")` : Devuelve un array con elementos que tienen el mismo nombre (por ejemplo, **radiobutton**).

VALIDACIÓN DEL FORMULARIO

Acceso a los elementos del formulario



Es importante tener en cuenta que hasta que no se haya cargado (creado) la página web, no podremos acceder a los elementos de la misma, para ello podríamos implementar:

- El atributo defer, a la hora de vincular el fichero JS.
- Asignar una función al evento `window.onload`.

```
window.onload = iniciar; //Sin paréntesis ya que si no se ejecutaría automáticamente
```

```
//Asocia un evento a la pulsación del boton enviar
function iniciar() {
    document.getElementById("enviar").addEventListener('click', validar, false);}
```

VALIDACIÓN DEL FORMULARIO

validar()



La función `validar()` comprueba que todos los campos del formulario son correctos.

```
if (validaNombre() && validaTelefono() && validaFecha() && validaCheck() &&
confirm("Pulsa aceptar si deseas enviar el formulario")) {
    return true;
}
```

A continuación, se implementan cada una de las funciones incluidas en el `if`, de la manera más básica posible por ahora.

VALIDACIÓN DEL FORMULARIO

validaNombre()



La función `validaNombre()`:

- Accede al elemento correspondiente al nombre:

```
let elemento = document.getElementById("nombre");
```

- Comprobamos que el campo no esté vacío, en tal caso devolveremos `false`:

```
if (elemento.value == "") {  
  alert("El campo está vacío. Por favor, introduzca un nombre.");  
  return false;  
}
```

- En caso contrario devolvemos `true`:

```
return true;
```

VALIDACIÓN DEL FORMULARIO

validaNombre()



Esta implementación es muy limitada, entre otras mejoras, podríamos implementar un parámetro con un nombre de campo y así comprobar, que dicho campo no esté vacío.

Esto resulta de utilidad en caso de tener muchos campos de tipo **input**, ya que así podemos validar dependiendo del campo que sea.

También podríamos comprobar, que el campo en cuestión, contenga un número de caracteres determinado.

VALIDACIÓN DEL FORMULARIO

validaTelefono()



La función `validaTelefono()`:

- Accede al elemento correspondiente al teléfono:

```
let elemento = document.getElementById("telefono");
```

- En este caso comprobamos que el contenido no sea un número, en tal caso devolveremos `false`:

```
if (isNaN(elemento.value)) {  
    alert("El campo teléfono tiene que ser numérico");  
    return false;
```

- En caso contrario devolvemos `true`:

```
return true;
```

VALIDACIÓN DEL FORMULARIO

validaFecha()



La función `validaFecha()`, para este caso, en el cual tenemos tres campos:

- Accede a los elementos correspondientes a la fecha:

```
let dia = document.getElementById("dia").value;  
let mes = document.getElementById("mes").value;  
let ano = document.getElementById("ano").value;
```

- A continuación, creamos un objeto `Fecha`, con esos valores:

```
let fecha = new Date(ano, mes, dia);
```

- Si ese objeto devuelve un número, se trata de una fecha válida (véase el objeto Date):

```
if (isNaN(fecha)) {  
    alert("Los campos de la fecha son incorrectos");  
    return false;  
}
```

VALIDACIÓN DEL FORMULARIO

validaSexo()



En este caso, al darle al usuario las opciones entre las que puede elegir, estamos limitando que pueda introducir un dato erróneo.

Por lo tanto, en este caso no tenemos que validar nada.

Adicionalmente, al establecer una selección predeterminada, mediante el atributo **checked**, en uno de los campos, aseguramos que no se envíe vacío.

VALIDACIÓN DEL FORMULARIO

validaCheck()



La función `validaCheck()`:

- Accede al elemento correspondientes al *check*:

```
let campoCheck = document.getElementById("mayor");
```

- A continuación, comprueba que este campo no esté marcado, en tal caso devolveremos `false`:

```
if (!campoCheck.checked) {  
    alert("Debes ser mayor de edad");  
    return false
```

- En caso contrario devolvemos `true`:

```
return true;
```

VALIDACIÓN DEL FORMULARIO

validar()



La función `validar()` comprueba que todas las funciones de validación devuelven `true`, es importante el orden de comprobación para que los posibles mensajes de error aparezcan en el orden deseado. Adicionalmente, se implementa un cuadro de diálogo `confirm` para confirmar que el usuario está seguro de enviar el formulario.

```
function validar(e) {  
    if (validaNombre() && validaTelefono() && validaFecha() && validaCheck()  
    && confirm("Pulsa aceptar si deseas enviar el formulario")) {  
        return true;  
    }  
}
```

VALIDACIÓN DEL FORMULARIO

validar()



En caso de que alguna función de validación devuelva `false` o bien el usuario pulse *cancelar* en el `confirm`, debemos evitar que se envíe el formulario.

Para ello, incluimos como parámetro (`e`), que hace referencia al objeto Evento creado automáticamente cuando se asocia una función a un evento. Y a este objeto le aplicamos el método `preventDefault()`, el cual cancela el evento por defecto asociado a la acción.

En este caso, la acción asociada al botón `submit`, es ir a la página `procesar.php`.

```
else {  
    e.preventDefault();  
    return false;  
}
```

VALIDACIÓN DEL FORMULARIO

Código JS



FINALMENTE EL FICHERO JS QUEDARÍA:

```
window.onload = iniciar; //Sin paréntesis
```

```
function iniciar() {  
    document.getElementById("enviar").addEventListener("click", validar, false);  
}
```

```
function validaNombre() {  
    let elemento = document.getElementById("nombre");  
    if (elemento.value == "") {  
        alert("El campo está vacío. Por favor, introduzca un nombre.");  
        error(elemento);  
        return false;  
    }  
    return true;  
}
```

VALIDACIÓN DEL FORMULARIO

Código JS



```
function validaTelefono() {  
  let elemento = document.getElementById("telefono");  
  if (isNaN(elemento.value)) {  
    alert("El campo teléfono tiene que ser numérico");  
    return false;  
  }  
  return true;  
}
```

```
function validaCheck() {  
  let campoCheck = document.getElementById("mayor");  
  if (!campoCheck.checked) {  
    alert("Debes ser mayor de edad");  
    return false;  
  }  
  return true;}  
}
```


VALIDACIÓN DEL FORMULARIO

Código JS



```
function validaFecha() {  
    let dia = document.getElementById("dia").value;  
    let mes = document.getElementById("mes").value;  
    let ano = document.getElementById("ano").value;  
  
    let fecha = new Date(ano, mes, dia);  
    if (isNaN(fecha)) {  
        alert("Los campos de la fecha son incorrectos");  
        return false;  
    }  
    return true;  
}
```

VALIDACIÓN DEL FORMULARIO

Código JS



```
function validar(e) {  
    if (  
        validaNombre() &&  
        validaTelefono() &&  
        validaFecha() &&  
        validaCheck() &&  
        confirm("Pulsa aceptar si deseas enviar el formulario")  
    ) {  
        return true;  
    } else {  
        e.preventDefault();  
        return false;  
    }  
}
```

VALIDACIÓN DEL FORMULARIO

Mejora



Opcionalmente, es posible implementar mediante CSS, la identificación del campo erróneo cuando estamos realizando la validación.

Para ello, definimos la clase CSS `.error`:

```
.error {  
    border: solid 2px red;  
}
```

VALIDACIÓN DEL FORMULARIO

Mejora



Y mediante la función `error()`, le aplicamos dicha clase al elemento pasado como argumento y además que se focalice.

```
function error(elemento) {  
    elemento.className = "error";  
    elemento.focus();  
}
```

Adicionalmente, implementamos una función `limpiarError()`, para que una vez el usuario corrija los posibles errores, desasocie la clase al elemento pasado como argumento.

```
function limpiarError(elemento) {  
    elemento.className = "";  
}
```

VALIDACIÓN DEL FORMULARIO

Mejora



A continuación, invocamos `error(elemento)` en la función `validaNombre()`, cuando no se cumpla la condición de validación.

```
if (elemento.value == "") {  
    alert("El campo está vacío. Por favor, introduzca un nombre.");  
    error(elemento);  
    return false;  
}
```

Si hubiéramos asignado a la variable `elemento`, directamente el valor del elemento en cuestión agregando `.value` para su comprobación en el `if`, ahora no podríamos pasar como argumento de la función `error()` esta variable.

VALIDACIÓN DEL FORMULARIO

Mejora



Por último, antes de hacer la validación del campo, nos aseguramos que eliminamos la clase `.error` de campo erróneo. De esta manera, cuando el usuario vuelve a comprobar la validez del campo, si este es correcto no aparecerá este *recordatorio*. Resultando la función `validaNombre()`:

```
function validaNombre() {  
    let elemento = document.getElementById("nombre");  
    limpiarError(elemento);  
    if (elemento.value == "") {  
        alert("El campo está vacío. Por favor, introduzca un nombre.");  
        error(elemento);  
        return false;  
    }  
    return true;  
}
```

VALIDACIÓN DEL FORMULARIO

Visualización del ejemplo



Puedes apreciar el comportamiento de este código en el siguiente link o en la imagen adjunta:

<https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/WbeYGeQ>

The screenshot shows a web interface for a form. At the top, there's a dark header with the text 'e formularios' and a pencil icon, a heart icon, and a 'Save' button. Below the header, the title 'Formulario' is displayed. The form is titled 'Datos personales' and contains the following fields:

- Nombre:
- Teléfono:
- Fecha de nacimiento: / /
- Sexo : ☒ Hombre ☐ Mujer
- Mayor de 18: ☒

At the bottom of the form, there are two buttons: 'Enviar' and 'Borrar'.