



Validación de formularios con HTML5 y JavaScript

Validación básica de formularios con HTML5

- Podemos realizar validaciones básicas empleando HTML5.
- Es Importante verificar que el elemento empleado funciona en todos los navegadores. Por ejemplo, si queremos emplear el objeto `<input type="week">` hay que tener en cuenta que no está soportado en Firefox ni IE12 o versiones anteriores.

(https://www.w3schools.com/jsref/dom_obj_week.asp)

- Hay atributos generales que podemos utilizar_
 - **Establecer un campo como obligatorio**
`<input type="text" id="nombre" required>`
 - **Limitar el tamaño de caracteres permitido en un elemento** `<input type="text" id="nombre" maxlength="15">`

Validación básica de formularios con HTML5

- **Establecer valor máximo y mínimo.** En campos de tipo Number, no es posible emplear patrones, pero si podemos hacer uso de los atributos min y max.

```
<input type="number" id="edad" min=18 max=65>
```

- **Establecer un patrón** (expresión regular) que el valor del campo debe cumplir.

Validación de un campo texto empleando el atributo pattern:

```
<input type="text" id="nombre" maxlength="15"
pattern="[A-Za-z]{2,15}" title="Introduce entre 2
y 15 letras">
```

Validación de campo teléfono empleando el atributo pattern:

```
<input type="text" id="telefono" maxlength="9"
pattern="[0-9]{9}" title="Número de 9 cifras">
```

Validación básica con JavaScript

- El primer paso es recuperar el formulario

```
Var formulario = document.getElementById("Miformulario")
```

(o empleando la opción de recuperación que más nos interese)

- Seleccionar los elementos (campos) del formulario. Tenemos varias opciones `getElementById`, `getElementsByTagName`, etc, `getElementsByName` (p.ej con type Radio), etc..

Validación básica con JavaScript

- Añadir código .js en fichero independiente
- Establecer manejador de evento “load” para asegurarnos de que todas los elementos html han sido cargados.

```
Window.addEventListener("load", iniciar);
```

- Iniciar añade un manejador de eventos para el botón del formulario “Enviar”, encargado de enviar los datos del mismo al servidor

```
function iniciar() {  
    document.getElementById("Enviar").addEventListener("click", validar, false);  
    //el elemento enviar es el botón que permite  
    enviar el formulario al servidor  
}
```

- Validar, será una función que permite comprobar que los campos son válidos
- Crearemos una función por cada campo a validar. A continuación se muestran tres funciones que validan respectivamente un campo nombre, un campo teléfono y un campo fecha

Validación básica con JavaScript

```
function validarNombre() {  
    var elem=document.getElementById("nombre"); limpiarError(elem);  
    if(elem.value==""){alert("El campo no puede estar vacío");  
    error(elem); return false;}  
    return true;  
}  
  
function validarTlfno(){  
Var elem=document.getElementById("tlfno"); limpiarError(elem);  
    If (isNaN(elem.value) == ""){alert("El campo tiene que ser  
    numéricos");error(elem);return false;}  
    return true;  
}  
  
function validarFecha(){  
Var dia=document.getElementById("dia");  
Var mes=document.getElementById("mes");  
Var ano=document.getElementById("ano");  
Var fecha = new Date(ano, mes, dia); limpiarError(dia); //mes y año  
If (isNaN(fecha)) { alert("dia mes o año incorrectos");  
error(elem); return false;  
}  
return true;  
}
```

La funciones **error()** y **limpiarError()** se explican más adelante.

Validación básica con Javascript

- Código de la función validar

```
function validar(e) {  
    If (validaNombre() && validarTlfno() && validarFecha() &&  
        confirm ("Pulsa aceptar para enviar el formulario")){  
        return true,  
    } else { e.preventDefault();}  
    //evitamos que el formulario sea enviado a la pagina indicada  
    en action //No se ejecuta la acción por defecto  
}
```

- Seria conveniente crear una clase .css para resaltar y mostrar al usuario de una forma clara que campos son erroneos.

```
Function error(elemento) {  
    Elemento.className="error";  
    Elemento.focus();  
}  
Function limpiarError(elemento) {  
    Elemento.className = "";  
}
```

Validación avanzada con JavaScript

JavaScript API para validaciones(https://www.w3schools.com/js/js_validation_api.asp)

Método	Descripción
checkValidity()	Devuelve verdadero si un elemento <code><input></code> contiene datos básicos.
setCustomValidity()	Establece la propiedad mensaje de validación para un elemento <code><input></code>

Propiedad	Descripción
validity	Contiene un valor booleano relativo a la validación de un <code><input></code>
validationMessage	Contiene el mensaje que el navegador visualizará si la validación es falsa.
willValidate	Indica si un elemento <code><input></code> será validado

La propiedad **validity** de un elemento `input` contiene una serie de propiedades (valores booleanos) relacionadas con la validación de datos. He aquí algunas:

Propiedad	Description
customError	verdadero, si un mensaje de validación está establecido
patternMismatch	Verdadero si el valor de un elemento no coincide con el patrón
rangeOverflow	verdadero, si el valor de un elemento es mayor que el indicado en el atributo <code>max</code>
rangeUnderflow	verdadero, si el valor de un elemento es menor que el indicado en su atributo <code>min</code>
tooLong	verdadero, si el valor de un elemento excede <code>maxLength</code> .
valueMissing	verdadero, si un elemento que es obligatorio (<code>required</code>) no tiene valor
valid	verdadero, si el valor de un elemento es valido

Validación avanzada con JavaScript

- Elemento.checkValidity() comprueba que la validación del elemento HTML es correcta

```
function validarNombre() {  
    var elem = document.getElementById("nombre");  
    if (!elem.checkValidity()) { //si la validación es incorrecta  
        error(elem); return false;  
    }  
    return true;  
}  
  
function validarEdad() {  
    var elem = document.getElementById("edad");  
    if (!elem.checkValidity()) { //si la validación es incorrecta  
        error(elem); return false;  
    }  
    return true;  
}  
  
function validarTlfno() {  
    var elem = document.getElementById("telefono");  
    if (!elem.checkValidity()) { //si la validación es incorrecta  
        error(elem); return false;  
    }  
    return true;  
}
```

Validación avanzada con JavaScript

A continuación añadimos las funciones validar, error y limpiar error:

```
function validar(e){
limpiarError();
if validarNombre() && validarEdad() && validarTlfno() &&
confirm("Deseas enviar el formulario?"){
    return true;
} else {
    e.preventDefault(); return false;
}
}

function error(elemento){
    document.getElementById("mensajeError".innerHTML =
    elemento.validationMessage;
    elemento.className = "error";
    elemento.focus();
}

function limpiarError(){
//quita todas las clases error de los <input>
var formulario=document.forms[0];
for (var i=0; i<.formulario.elements.lenght;i++){
    formulario.elements[i].className="";
}
}
```

Validación avanzada con JavaScript

```
function validarNombre() {  
    var elem = document.getElementById("nombre");  
    if (!elem.checkValidity()) { //validación incorrecta  
        If (elem.validity.valueMissing) {  
            error2(elem, "Debe introducir un nombre");  
        }  
        If (elem.validity.patternMismatch) {  
            error2(elem, "El nombre debe tener entre 2 y 15  
caracteres");  
        }  
        //error(elem);  
    }  
    return false;  
}  
return true;  
} //validarNombre
```

Validación avanzada con JavaScript

```
function validarEdad(){
var elem = document.getElementById("edad");
if (!elem.checkValidity()){ //validación incorrecta
    if (elem.validity.valueMissing){
        error2(elem,"Debe introducir un valor");
    }
    if (elem.validity.rangeOverflow){
        error2(elem,"El valor debe ser menor que
65");
    }
    if (elem.validity.rangeUnderflow){
        error2(elem,"El valor debe ser mayor que
18");
    }
    //error(elem);
    return false;
}
return true;
}
```

Validación avanzada con Javascript

```
function validarTlfno(){
var elem = document.getElementById("telefono");
    if (!elem.checkValidity()){ //validación incorrecta
        if (elem.validity.valueMissing){
            error2(elem,"Debe introducir un numero de tlfno");
        }
        if (elem.validity.patternMismatch){
            error2(elem,"El telefono debe tener 9 digitos");
        }
        //error(elem);
        return false;
    }
    return true;
}

function error2(elem,mensaje){
//visualiza en un parrafo de la pagina el mensaje de error
    document.getElementById("mensajeError").innerHTML =
mensaje;
    elem.className = "error";
    elem.focus();
}
```