

Desarrollo Web en Entorno Cliente

UD 06. Formularios

Actualizado Octubre 2020

Licencia




Reconocimiento – NoComercial - CompartirIgual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 **Importante**

 **Atención**

 **Interesante**

ÍNDICE DE CONTENIDO

1. Introducción	3
2. Acceso a elementos de formularios	3
2.1. text	3
2.2. Radio button	3
2.3. checkbox	4
2.4. select	4
3. Validación de formularios: evento onsubmit	5
3.1. Validar un formulario	5
3.2. Deshabilitar enviar un formulario dos veces	5
3.3. Enviar un formulario desde código	5
4. Validación de formularios: Expresiones regulares	6
4.1. Funciones Javascript para el uso de expresiones regulares	6
5. Bibliotecas para validación formularios Javascript	6
6. Bibliografía	6
7. Autores (en orden alfabético)	6

UD06. FORMULARIOS

1. INTRODUCCIÓN

En esta unidad trataremos algunas de las operaciones más habituales en los formularios. En primer lugar veremos cómo acceder a elementos de formularios de distinto tipo. También veremos como validar envíos de formularios a mano, mediante el evento “onsubmit” y el uso sencillo de expresiones regulares.

! **Atención:** al programar una aplicación Web, el cliente debe validar la información introducida, pero ello no quita que el servidor también debe validarla, siendo esta validación incluso más importante que la validación en el cliente.

2. ACCESO A ELEMENTOS DE FORMULARIOS

2.1 text

Para acceder al valor de un input de tipo texto, simplemente debemos referenciar el atributo “value”.

Ejemplo:

```
<input type="text" id="miTexto">
```

Javascript asociado:

```
let elemento=document.getElementById("miTexto");  
alert(elemento.value);
```

2.2 Radio button

Los “Radio button” son elementos del formulario que ante varias entradas te permiten seleccionar sólo una de ellas. Se agrupan teniendo un “name” común.

Para acceder a ellos, se accede como un array, donde se tiene el atributo “value” y el atributo “checked” que es true si está seleccionado, false en caso contrario.

Ejemplo:

```
<input type="radio" id="preguntaSI" name="pregunta" value="si" />  
<label for="preguntaSI">SI</label>  
<input type="radio" id="preguntaNO" name="pregunta" value="no" />  
<label for="preguntaNO">NO</label>
```

Javascript asociado:

```
let elementos=document.getElementsByName("pregunta");  
for(let i=0;i<elementos.length;i++){  
    if(elementos[i].checked===true)  
        alert("Valor del elemento marcado "+elementos[i].value);  
}
```

2.3 checkbox

Similar a los “Radio button”, salvo que permite que haya más de un elemento marcado.

Ejemplo:

```
<input type="checkbox" id="preguntaAS" name="pregunta" value="asc" />
<label for="preguntaAS">Piso con ascensor</label>
<input type="checkbox" id="preguntaAM" name="pregunta" value="amb" />
<label for="preguntaAM">Piso amueblado</label>
```

Javascript asociado:

```
let elementos=document.getElementsByName("pregunta");
for(let i=0;i<elementos.length;i++){
    if(elementos[i].checked===true){
        alert("Valor del elemento marcado "+elementos[i].value);
    }
}
```

2.4 select

Elemento que muestra un desplegable y nos permite elegir una opción del mismo. Aquí destaca el atributo “options”, que es un atributo que contiene un array con las opciones disponibles y el atributo “selectedIndex” que contiene (y se puede modificar) la posición del array “options” seleccionada actualmente (o la primera si se permite multiselección) o -1 si no está seleccionada ninguna opción (o queremos des-seleccionarlas).

Dentro de cada “options”, “value” almacena el valor y “text” el texto mostrado.

Ejemplo:

```
<select id="aprobar" >
    <option value="10">Saco 10 en DWEC</option>
    <option value="9">Saco 9 en DWEC</option>
    <option value="8">Saco 8 en DWEC</option>
</select>
```

Javascript asociado:

```
let elemento=document.getElementById("aprobar");
for(let i=0;i<elemento.options.length;i++){
    alert("Valor de la opción "+elemento.options[i].value);
}
let sel=elemento.selectedIndex;
alert("El valor de la opción seleccionada es
```

```
" + elemento.options[sel].value + " y el texto asociado es  
" + elemento.options[sel].text);  
// Cambiamos el índice seleccionado  
elemento.selectedIndex=0;
```

3. VALIDACIÓN DE FORMULARIOS: EVENTO ONSUBMIT

3.1 Validar un formulario

Si un manejador de un evento devuelve true (o no devuelve nada), se realiza el evento asociado. Si el manejador devuelve false, se cancela el evento.

Existe un evento asociado a un formulario completo llamado “onsubmit”.

Aprovechando esto, podemos a nuestro antojo permitir el envío de información al servidor mediante el formulario devolviendo true o **cancelarlo devolviendo false**.

La estructura típica es la siguiente:

```
<form onsubmit="return validar();">
```

Si la función validar devuelve true, se realiza el envío. Si devuelve false, se cancela.

En la función validar podemos hacer las validaciones que estimemos convenientes.

3.2 Deshabilitar enviar un formulario dos veces

A veces un usuario pulsa enviar un formulario más de una vez por error.

Si queremos evitar esto, podemos usar “this.disabled=true”;

Ejemplo:

```
<form onsubmit="this.disabled=true;">
```

3.3 Enviar un formulario desde código

En algunas aplicaciones por motivos estéticos o de funcionalidad es deseable que el “enviar un formulario” no se haga desde un botón “submit”, sino desde cualquier otro evento que permita la ejecución de código. Esto se puede hacer recogiendo el elemento del formulario y aplicando el método submit().

Ejemplo:

```
<form id="formulario">
```

Javascript asociado:

```
let elemento=document.getElementById("formulario");  
elemento.submit();
```

4. VALIDACIÓN DE FORMULARIOS: EXPRESIONES REGULARES

4.1 Funciones Javascript para el uso de expresiones regulares

Javascript posee dos formas de crear expresiones regulares:

- Literal con expresión regular. Ejemplo `var re = /ab+c/;`
- El objeto `RegExp`. Ejemplo `var re = new RegExp("ab+c");`

Entre distintos métodos, uno de los más usados es "test". El método "test" recibe una cadena y devuelve true si la cadena cumple esa expresión regular, false en caso contrario.

Ejemplo:

```
let re = new RegExp("ab+c");
let cadena=prompt("Dime una cadena");
if(re.test(cadena)){
    alert("La cadena cumple el patrón de una a, entre 1 e infinitas b
y al final una c");
}
```

Sobre el uso de expresiones regulares tenéis más información en

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions

Podéis utilizarlas siempre que queráis para cualquier ejercicio.

5. BIBLIOTECAS PARA VALIDACIÓN FORMULARIOS JAVASCRIPT

Algunas de las bibliotecas más populares para la validación de formularios Javascript son:

- joi <https://joi.dev/>
- Validator.js <https://github.com/validatorjs/validator.js>
- Validate.js <http://rickharrison.github.io/validate.js/>
- jQuery Validation Plugin <https://jqueryvalidation.org/documentation/>

6. BIBLIOGRAFÍA

[1] Javascript Mozilla Developer <https://developer.mozilla.org/es/docs/Web/JavaScript>

[2] Javascript ES6 W3C https://www.w3schools.com/js/js_es6.asp

[3] Expresiones regulares

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions

7. AUTORES (EN ORDEN ALFABÉTICO)

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:

- García Barea, Sergi