

## UD 06. FORMULARIOS

Desarrollo web en entorno cliente  
CFGS DAW

Sergio García Barea  
[sergio.garcia@ceedcv.es](mailto:sergio.garcia@ceedcv.es)

2019/2020

Versión:191007.1720


## 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

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Acceder a elementos de formularios.....</b>	<b>3</b>
2.1 text.....	3
2.2 Radio button.....	3
2.3 checkbox.....	4
2.4 select.....	4
<b>3. Validación de formularios: evento onsubmit.....</b>	<b>5</b>
3.1 Validar un formulario.....	5
3.2 Deshabilitar enviar un formulario dos veces.....	5
3.3 Enviar un formulario desde código.....	5
<b>4. Validación de formularios: Expresiones regulares.....</b>	<b>6</b>
4.1 Funciones Javascript para el uso de expresiones regulares.....	6
<b>5. Bibliotecas para validación formularios Javascript.....</b>	<b>6</b>
<b>6. Material adicional.....</b>	<b>6</b>
<b>7. Bibliografía.....</b>	<b>7</b>

## UD06. FORMULARIOS

### 1. INTRODUCCIÓN

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

 Al hacer una aplicación Web, el cliente debe validar la información introducida, pero ello no quita el servidor también deba validarla.

### 2. ACCEDER 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

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

#### 2.2 Radio button

Radio button son elementos del formulario, que ante varias entradas, te dejan seleccionar 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" />SI  
<input type="radio" id="preguntaNO" name="pregunta" value="no" />NO
```

##### Javascript asociado

```
var elementos=document.getElementsByName("pregunta");  
for(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 puede haber más de uno marcado.

### Ejemplo:

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

### Javascript asociado

```
var elementos=document.getElementsByName("pregunta");  
for(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

```
var elemento=document.getElementById("aprobar");  
for(i=0;i<elemento.options.length;i++){  
    alert("Valor de la opción "+elemento.options[i].value);  
}  
var 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 mas 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 aplicándole el método submit();

Un ejemplo de esto en Javascript.

**Ejemplo:**

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

#### Javascript asociado

```
var 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`. `test` recibe una cadena y devuelve `true` si la cadena cumple esa expresión regular, `false` en caso contrario.

#### Ejemplo:

```
var re = new RegExp("ab+c");
var cadena=prompt("Dime una cadena");
if(re.test(cadena)){
    alert("La cadena cumple el patron 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](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:

- `Validate.js` <http://rickharrison.github.io/validate.js/>
- `jQuery Validation Plugin` <https://jqueryvalidation.org/documentation/>

## 6. MATERIAL ADICIONAL

[1] Curso de Javascript en Udacity validate form jquery

<https://www.udacity.com/course/javascript-basics--ud804>

[2] Enlaces Libros Web validación

[http://librosweb.es/libro/javascript/capitulo\\_7/validacion.html](http://librosweb.es/libro/javascript/capitulo_7/validacion.html)

## 7. BIBLIOGRAFÍA

[1] Referencia Javascript

<http://www.w3schools.com/jsref/>

[2] Expresiones regulares

[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Regular_Expressions)