

Índice: Tema 3.6

3.6 Formularios en JavaScript

3.6.1 Acceso y propiedades

3.6.2 Métodos y utilidades

3.6.3 Validación de formularios

Índice: Temas 3.6 y 3.7

3.6 Formularios en JavaScript

3.6.1 Acceso y propiedades

3.6.2 Métodos y utilidades

3.6.3 Validación de formularios



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ Formularios en JavaScript

- ✓ La programación de aplicaciones que contienen **formularios** web siempre ha sido una de las tareas fundamentales de JavaScript:
 - Una de las principales razones por las que se inventó el lenguaje de programación JavaScript fue la necesidad de validar los datos de los formularios directamente en el navegador del usuario. De esta forma, se evitaba recargar la página cuando el usuario cometía errores al rellenar los mismos.
- ✓ No obstante, la aparición de las aplicaciones AJAX ha relevado al tratamiento de formularios como la principal actividad de JavaScript:
 - Ahora, el principal uso de JavaScript es el de las comunicaciones asíncronas con los servidores y el de la manipulación dinámica de las aplicaciones.
- ✓ De todas formas, el manejo de los formularios sigue siendo un requerimiento imprescindible para cualquier programador de JavaScript.



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ Acceso a los formularios y sus elementos

usuario

2 -----

contraseña
enviar

- ✓ Cuando se carga una página web, el navegador crea automáticamente un array llamado **forms** (accesible desde el objeto **document**) que contiene la referencia a todos los formularios de la página.

3

```
var primerFormulario = document.forms[0];
```

- ✓ Además, para cada formulario se crea otro array **elements** con la referencia a todos los elementos (cuadros de texto, botones, listas, etc...) del mismo:

```
var primerElementoDelPrimerFormulario = document.forms[0].elements[0];
```

document.forms=[form1,form2,form3]

- ✓ Otra opción (mejor que la anterior) es acceder mediante nombre o id con las funciones del DOM. Por ejemplo, supóngase que al primer formulario e la página tiene una **id="formulario"**, entonces, para acceder al mismo, se puede hacer:

```
var primerFormulario = document.getElementById("formulario");
```

```
var primerFormulario = document.formulario; // si name="formulario"
```



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ Propiedades de los elementos

importante `document.formulario.password.value`

para acceder a que ha escrito el usuario en el cuadro de texto

- ✓ Cada elemento dispone de las siguientes propiedades útiles para el desarrollo de las aplicaciones:

password es un elemento campo de un formulario

..... `<input type="text" name="password" value="3" id="password" >`

- **type**: Indica el tipo de elemento (control) del formulario. Para los elementos de tipo `<input>` coincide con el valor de su atributo `type`. Para los elementos `<select>` su valor es *select-one* o *select-multiple*. Para los elementos de tipo `<textarea>`, su valor es *textarea*.
- **form**: Referencia directa al formulario al que pertenece el elemento. Ejemplo de uso:
`document.getElementById("id_del_elemento").form`
- **name**: Valor del atributo *name* del HTML. Solamente se puede leer su valor.
formulario al que pertenece el elemento password: `document.formulario.password.form`
- **value**: Permite leer y modificar el valor del atributo *value* del HTML. Para los campos de texto (`<input type="text">` y `<textarea>`) obtiene el texto que ha escrito el usuario. Para los botones obtiene el texto que se muestra en el botón. Para los elementos *checkbox* y *radiobutton* no es muy útil (como se verá más adelante).



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



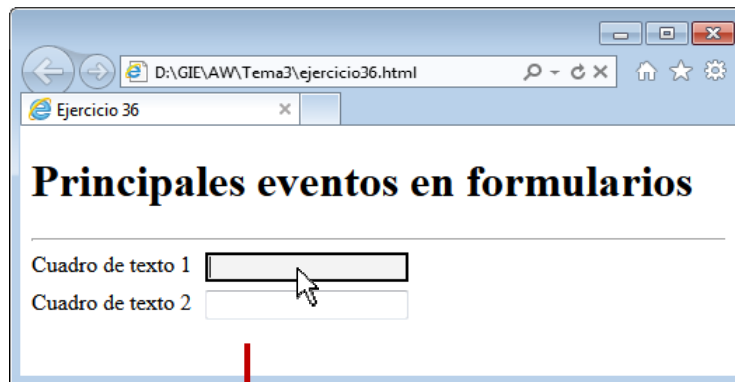
➤ Principales eventos (1)

imprtate

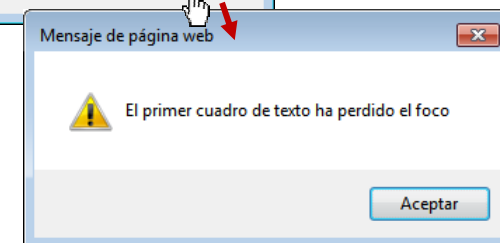
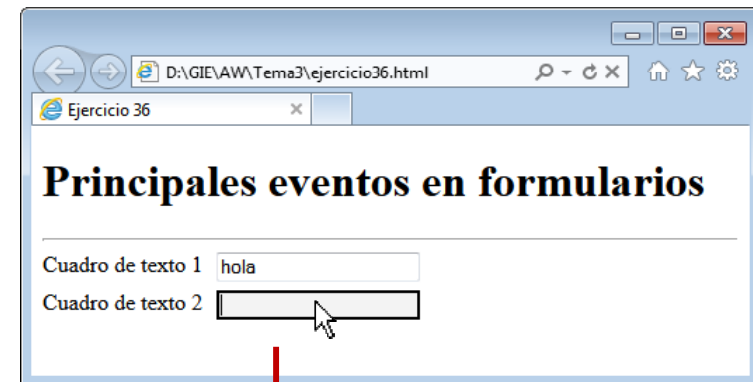
- ✓ Los eventos más utilizados en el manejo de formularios son:
 - **onclick**: Evento que se produce cuando se pincha con el ratón sobre un elemento. Normalmente se utiliza con cualquiera de los tipos de botones que permite definir HTML (<input type="button">, <input type="submit">, <input type="image">).
 - **onfocus**: Evento que se produce cuando el usuario selecciona un elemento del formulario (gana el foco).
`onfocus=" myfunc()"`
 - **onblur**: Evento complementario de onfocus, ya que se produce cuando el usuario ha deseleccionado un elemento por haber seleccionado otro (pierde el foco).
`onblur="my func()"`
 - **onchange**: Evento que se produce cuando el usuario cambia el valor de un elemento de texto (<input type="text"> o <textarea>). También se produce cuando el usuario selecciona una opción en una lista desplegable (<select>). El evento sólo se produce si después de realizar el cambio, el usuario pasa al siguiente campo del formulario.

➤ Principales eventos (2)

- ✓ Probar los eventos **onfocus**, **onblur** y **onchange** sobre el primer cuadro de texto del siguiente formulario mostrando mensajes que informen de su estado.



on focus=alert("text")





3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ Acceso a valores (1) imprtate

✓ A continuación se muestra cómo obtener el valor de los campos de formulario más utilizados:

- **Cuadros de texto y textarea:** El valor del texto mostrado por estos elementos se obtiene y se establece directamente mediante la propiedad **value**.
- **Casillas de verificación y botones de radio:** Además de **value**, la propiedad **checked** devuelve **true** si el elemento está seleccionado y **false** en caso contrario.

imprtate. **Las listas de selección:** En general lo que se quiere obtener es el valor de la propiedad **value** para los elementos seleccionados. Para ello, las listas disponen del array **options** con todas las opciones de la lista y la propiedad **selectedIndex** con el índice de la opción seleccionada. A continuación, las propiedades **text** y **value** de la opción correspondiente proporcionan el texto y valor de la misma respectivamente.

```
var objlista = document.getElementById("id_del_select");  
var valor = objlista.options[objlista.selectedIndex].value;  
var texto = objlista.options[objlista.selectedIndex].text;
```

<select i
<option



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES

EJERCICIO 37



➤ Acceso a valores (2)

- ✓ Acceder a los campos del siguiente formulario y mostrar el resultado en la caja de la derecha con el formato especificado.

El navegador muestra la URL `D:\GIE\AW\Tema3\ejercicio37.html` y la pestaña `Ejercicio 37`.

Valores de los campos de formulario

Formulario

Cuadro de texto:

Área de texto:

Lista de selección:

☒ Casilla de verificación 1

☒ Casilla de verificación 2

☐ Botón de radio 1

☒ Botón de radio 2

El valor del cuadro de texto es TEXTO

El valor del área de texto es AREA

Opción seleccionada en la lista 'Cuarto valor' con valor '4'

El valor de la casilla 1 es 'casilla 1' con checked 'true'

El valor de la casilla 2 es 'casilla 2' con checked 'true'

El valor del botón de radio 1 es 'on' con checked 'false'

El valor del botón de radio 2 es 'on' con checked 'true'



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ El elemento innerHTML

- ✓ Una posibilidad para agregar los párrafos con el resultado de cada una de las lectura de los datos del formulario de la izquierda en la caja de la derecha, es utilizar los métodos del DOM. Para ello:
 - Se crea el párrafo con el primer resultado y se añade a la caja mediante el método **appendChild** (transparencia 20 del Tema 3.3).
- ✓ Otra posibilidad, es utilizar la propiedad **innerHTML** de la caja. Dicha propiedad la tienen todas las etiquetas HTML (objetos DOM) y permite modificar el contenido de las mismas.
 - Se crea una cadena (string) con los párrafos resultado, y se establece como contenido de la caja (supóngase un id de la misma id="resultado"):

```
cadena = "<p>El valor del cuadro de texto es " + document.formulario.texto.value +  
        "</p><br> <p>El valor del area de texto es " + document.form...  
  
document.getElementById("resultado").innerHTML = cadena;
```



3.6 FORMULARIOS EN JAVASCRIPT

3.6.1 ACCESO Y PROPIEDADES



➤ Funciones de parseo a datos numéricos

- ✓ Los valores leídos de los diferentes campos de un formulario son cadenas (string), por lo que puede ser necesario convertirlos a un tipo diferente si se desea operar con ellos.
- ✓ Supóngase el caso en el que los valores tomados de dos campos de texto son los operandos de una suma: el resultado de $1 + 2$ daría la cadena 12.
- ✓ Java Script (al igual que otros lenguajes) posee una serie de funciones para llevar a cabo la conversión entre tipos adecuada:
 - **parseInt(cadena, base)**: Devuelve el número entero al que convierte la cadena entrada según la base especificada. Una base de 10 indica una conversión a número decimal, 2 binario, 8 octal, 16 hexadecimal, etc...
 - **parseFloat(cadena)**: Devuelve el número en punto flotante al que convierte la cadena entrada.

Índice: Temas 3.6 y 3.7

3.6 Formularios en JavaScript

3.6.1 Acceso y propiedades

3.6.2 Métodos y utilidades

3.6.3 Validación de formularios



3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES



➤ Métodos

- ✓ Además de poder acceder a los valores de cada uno de los elementos de un formulario, éstos disponen de ciertos métodos de utilidad:

- **blur()**: Quita el foco al elemento.
- **focus()**: Pone el foco en el elemento.
- **select()**: Selecciona el texto de un cuadro o área de texto.
- **add(opt,pos)**: Añade una nueva opción a la lista en la posición especificada.
- **click()**: Simula la pulsación de un botón o una casilla.
- **reset()**: Simula la pulsación del botón de reset.
- **submit()**: Simula la pulsación del botón de envío.



3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES



➤ Establecer el foco

- ✓ En páginas de búsqueda o páginas con un formulario para registrarse, se considera una buena práctica de usabilidad el asignar automáticamente el foco al primer elemento del formulario cuando se carga la página.

```
window.onload = function() {  
  
    document.getElementById("id_del_elemento").focus();  
  
}
```

➤ Evitar duplicados

- ✓ Uno de los problemas habituales con el uso de formularios web es la posibilidad de que el usuario pulse dos veces seguidas sobre el botón "Enviar".

```
<input type="button" value="Enviar" onclick= "this.disabled=true;  
                                                this.value='Enviando... ';  
                                                this.form.submit();" />
```



3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES



➤ Añadir opciones a una lista de selección

- ✓ Como se ha comentado antes, las listas de selección crean un array **options** con todas las opciones de la lista.
- ✓ Cada una de estas opciones es un objeto del tipo **Option**.
- ✓ Así, para crear nuevas opciones dinámicamente en la lista, es suficiente con:
 1. Crear un objeto de tipo **Option** para la nueva opción.
 2. Asignarle texto y valor mediante sus propiedades **text** y **value**.
 3. Añadirlo a la lista mediante la función **add(opcion, posicion)** de la misma.
- ✓ Siendo el código correspondiente:
 1. `var objopcion = new Option();`
 2. `objopcion.text = texto;`
`objopcion.value = valor;`
 3. `document.formulario.lista.add(objopcion, objopcion.length);`



3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES



➤ Limitar la entrada en un área de texto (1)

- ✓ La carencia más importante del elemento **textarea** es la imposibilidad de limitar el máximo número de caracteres que se pueden introducir, de forma similar al atributo **maxlength** de los cuadros de texto normales.
- ✓ Sin embargo, esta funcionalidad puede simularse teniendo en cuenta que el comportamiento normal de algunos eventos se puede evitar si se devuelve el valor **false** a su correspondiente manejador:
 - Si por ejemplo se devuelve el valor **false** en el evento **onkeypress** de un área de texto, la tecla pulsada por el usuario no se tiene en cuenta.

```
<textarea onkeypress="return false;"></textarea>
```

- Si se devuelve **false** en el evento **onclick** de un elemento como un enlace, el navegador no carga la página indicada por el enlace.

```
<a href="http://www.pagina.es" onclick="return false;">Enlace</a>
```




3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES



➤ Limitar la entrada en un área de texto (2)

- ✓ Esta característica puede utilizarse para limitar y restringir la entrada de texto en algunos campos de un formulario.
- ✓ Por ejemplo, supongamos que queremos limitar a 10 el número de caracteres a introducir en un área de texto:

```
<form name="formulario">  
  <label>Área de texto</label>  
  <textarea name="area" onkeypress="return limitar(10)"></textarea>  
</form>  
  
function limitar(MAXCHAR) {  
  return document.formulario.area.value.length<MAXCHAR;  
}
```

- ✓ Siguiendo esta filosofía, se puede restringir la entrada a un determinado conjunto de caracteres como sólo números, sólo letras, o prohibir las teclas especiales, etc...



➤ Utilidades con formularios (1)

- ✓ Utilizar los métodos anteriores para diseñar una web en la que:
 - El foco se posicione en el primer cuadro de texto al abrir la página.
 - Dicho cuadro de texto pierda el foco al pasar el ratón por encima.
 - El texto introducido en el segundo área de texto se seleccione con un solo click.
 - La lista tenga tres opciones con valores 1, 2 y 3 y textos “Primer valor”, “Segundo valor” y “Tercer valor” respectivamente.
 - Se puedan introducir nuevas opciones con el texto y valor introducidos desde los cuadros de texto situados bajo la misma.
 - La casilla de verificación se seleccione o deseleccione con tan sólo pasar el ratón por encima.
 - El área de texto no permita la introducción de números y 10 caracteres como máximo.
 - El botón enviar evite el envío duplicado al enviar el formulario por email.



3.6 FORMULARIOS EN JAVASCRIPT

3.6.2 MÉTODOS Y UTILIDADES

EJERCICIO 38



➤ Utilidades con formularios (2)

Utilidades con formularios

Cuadro de texto 1:

Cuadro de texto 2:

Lista de selección: Primer valor ▼

Añadir una opción: texto valor Añadir

Casilla de verificación: ☐

Área de texto:

Enviar



Utilidades con formularios

Cuadro de texto 1:

Cuadro de texto 2:

Lista de selección: Cuarto valor ▼

Añadir una opción: Cuarto valor 4 Añadir

Casilla de verificación: ☒

Área de texto:

Enviando...

Índice: Temas 3.6 y 3.7

3.6 Formularios en JavaScript

3.6.1 Introducción

3.6.2 Utilidades

3.6.3 Validación de formularios



3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Introducción (1)

- ✓ La principal utilidad de JavaScript en el manejo de los formularios es la validación de los datos introducidos por los usuarios.
- ✓ Antes de enviar un formulario al servidor, se recomienda validar mediante JavaScript los datos insertados por el usuario.
- ✓ De esta forma, si el usuario ha cometido algún error al rellenar el formulario, se le puede notificar de forma instantánea, sin necesidad de esperar la respuesta del servidor.
- ✓ Normalmente, la validación de un formulario consiste en llamar a una función de validación cuando el usuario pulsa sobre el botón de envío del formulario.
- ✓ En esta función, se comprueban si los valores que ha introducido el usuario cumplen las restricciones impuestas por la aplicación.



3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Introducción (2)

- ✓ A continuación se muestra el código JavaScript básico necesario para incorporar la validación a un formulario:

```
<form action="" method="" ... onsubmit="return funValidar(this)">  
</form>
```

- ✓ Siendo el esquema de la función **funValidar()** el siguiente:

```
function funValidar(objformulario) {  
    if (no se cumplen las restricciones del primer campo del formulario) {  
        alert("[ERROR] El campo debe tener un valor de...");  
        return false;  
    }  
    else if (no se cumplen las restricciones del segundo campo del formulario) {  
        alert("[ERROR] El campo debe tener un valor de...");  
        return false;  
    }  
    ...  
    else if (no se cumplen las restricciones del último campo del formulario) {  
        alert("[ERROR] El campo debe tener un valor de...");  
        return false;  
    }  
    return true;  
}
```

➤ Introducción (3)

- ✓ Validar una contraseña antes de enviarla por correo:

Validación de una contraseña

Contraseña

Repita la contraseña

Enviar

Validación de una contraseña

Contraseña

Enviar

Mensaje de página web

Las contraseñas no coinciden

Aceptar



3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Campo de texto obligatorio

- ✓ Se trata de forzar al usuario a introducir un valor en un cuadro o área de texto o en los que sea obligatorio:

```
valor = document.getElementById("id_el_campo_de_texto").value;  
if( valor == null || valor.length == 0 || /^\\s+$/.test(valor) ) {  
    alert("[ERROR] El campo de texto es obligatorio");  
    return false;  
}
```

➤ Campo de texto con valores numéricos

- ✓ Se trata de obligar al usuario a introducir un valor numérico en un cuadro de texto:

```
valor = document.getElementById("id_el_campo_de_texto").value;  
if( isNaN(valor) ) {  
    alert("[ERROR] El campo de texto sólo debe contener números");  
    return false;  
}
```




3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Dirección de email con un formato adecuado

- ✓ Se trata de obligar al usuario a introducir una dirección de email con un formato válido. Por tanto, lo que se comprueba es que la dirección *parezca válida, ya que no se comprueba si se trata de una cuenta de correo electrónico real y operativa.*

```
valor = document.getElementById("id_el_campo_de_texto").value;  
if( !(/^\w+([.-]?\w+)*@\w+([.-]?\w+)*(\.\w{2,3})+$/).test(valor)) {  
    alert("[ERROR] El campo de texto debe contener un email");  
    return false;  
}
```

➤ Selección de una opción en la lista

- ✓ Se trata de obligar al usuario a seleccionar un elemento de una lista desplegable:

```
indice = document.getElementById("id_de_la_lista").selectedIndex;  
if( indice == null || indice == 0 ) {  
    alert("[ERROR] Debe seleccionar una opción de la lista");  
    return false;  
}
```



3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Validar una fecha

- ✓ Un evento es una señal que se dispara al producirse algún cambio. Las fechas suelen ser los campos de formulario más complicados de validar por la multitud de formas diferentes en las que se pueden introducir.
- ✓ El siguiente código asume que de alguna forma se ha obtenido el año, el mes y el día introducidos por el usuario:

```
ano = obtener el año;  
mes = obtener el mes;  
dia = obtener el día;  
  
fecha= new Date(ano, mes-1, dia);  
if( !fecha || fecha.getFullYear()!=ano || fecha.getMonth()!=(mes-1) ||  
    fecha.getDate()!=dia) {  
    alert("[ERROR] Introduzca una fecha válida");  
    return false;  
}
```



3.6 FORMULARIOS EN JAVASCRIPT

3.6.3 VALIDACIÓN DE FORMULARIOS



➤ Selección de una casilla de verificación

- ✓ Si una casilla de verificación se debe seleccionar de forma obligatoria, JavaScript permite comprobarlo de forma muy sencilla:

```
elemento = document.getElementById("id_de_la_casilla");  
if( !elemento.checked ) {  
    alert("[ERROR] La casilla de verificación es obligatoria");  
    return false;  
}
```

➤ Selección de un botón de radio

- ✓ En este caso se ha de comprobar que alguno de los botones de radio que forman un determinado grupo esté seleccionado:

```
elementos = document.getElementsByName("name_del_grupo");  
var seleccionado = false;  
for(var i=0; i<elementos.length; i++)  
    if(elementos[i].checked) return true;  
return seleccionado;
```

➤ Validación de formularios

- ✓ Probar las validaciones anteriores mediante el siguiente formulario:

The screenshot shows a web browser window with the address bar displaying 'D:\GIE\AW\Tema3\ejercicio40.html'. The browser window has a tab titled 'Ejercicio 40'. The page content is titled 'Validación de formularios' and contains the following form elements:

- Campo obligatorio:
- Campo numérico:
- Correo electrónico:
- Lista de selección:
- Fecha (día-mes-año):
- Botón de radio 1: ☐
- Botón de radio 2: ☒
- Acepto los términos: ☒
- Enviar:

A red arrow points from the 'Enviar' button to a JavaScript alert message box titled 'Mensaje de página web'. The message box contains a yellow warning icon and the text '[ERROR 5] Introduzca una fecha válida'. There is an 'Aceptar' button at the bottom of the message box.