



APRENDERAPROGRAMAR.COM

FORMULARIOS
JAVASCRIPT: EJEMPLOS
DE DOCUMENT.FORMS Y
ELEMENTS. ACCESO
DIRECTO A FORMULARIOS
Y CAMPOS CON ID Y
NAME (CU01175E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº75 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

TRABAJO CON FORMULARIOS JAVASCRIPT

En la jerarquía de objetos JavaScript que hemos visto podríamos distinguir entre tres tipos de objetos: objetos del navegador (window, document, etc.), objetos del lenguaje (Math, RegExp, etc.) y por su importancia tradicional en la programación JavaScript, los objetos campos de formularios. Vamos a hablar sobre el trabajo con formularios en JavaScript.



Cuando comenzó el desarrollo de internet y de la programación web existía el problema de que el trabajo con formularios era muy pesado. Supón que un usuario deja un campo de un formulario que es obligatorio vacío: si la información tiene que ser enviada al servidor, en el servidor procesarse la información, comprobar que hay un campo vacío, y enviar una respuesta al usuario informándole de que hay un campo vacío, se consume tiempo, puede haber interrupciones, el usuario se desespera... Por ello JavaScript se convirtió en una potente herramienta para solucionar este problema: realizando la validación del formulario del lado del cliente (en el propio pc del usuario) la detección de un error es instantánea y no requiere comunicación con el servidor. Sólo cuando el formulario está correcto se permite su envío. Por esto el uso en formularios ha sido una de las aplicaciones principales de JavaScript históricamente.

Si recordamos la jerarquía de objetos JavaScript podemos ver gráficamente la importancia de los formularios y campos de formularios.



Hoy en día el trabajo con formularios no tiene tanta importancia como la tuvo en el pasado, pero sigue siendo un aspecto importante de la programación web y de JavaScript.

TODOS LOS FORMULARIOS EN FORMS

La propiedad forms del objeto document devuelve una colección de objetos tipo array conteniendo todos los formularios que existan en el documento HTML, con índice 0 para el primer formulario por orden de aparición, índice 1 para el siguiente y así sucesivamente.

En este ejemplo vemos cómo podemos acceder a document.forms y obtener el número de formularios existentes en el documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css"> label{display:block;margin:5px;}</style>
<script type="text/javascript">
window.onload = function () {          var ejemplo = document.getElementById('ejemplo');
                                      ejemplo.addEventListener("click", ejecutarEjemplo);      }
function ejecutarEjemplo () {
var formularios = document.forms; alert('El número de formularios en el documento es: '+formularios.length);      }
</script></head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue; margin:20px;" id="ejemplo">Pulsa aquí</div>
<form name="formularioContacto" method="get" action="accion1.html">
<h2>Formulario de contacto</h2>
<label>Nombre:<input id="nombreFormContacto" type="text" name="nombre" /></label>
<label>Apellidos:<input id="apellidosFormContacto" type="text" name="apellidos" /></label>
<label><input id="botonEnvio1" type="submit" value="Enviar"></label>
</form>
<form name="formularioReclamacion" method="get" action="accion2.html">
<h2>Formulario de reclamación</h2>
<label>Motivo reclamación:<input id="motivoFormReclama" type="text" name="motivo" /></label>
<label>Fecha del hecho:<input id="fechaFormReclama" type="text" name="fecha" /></label>
<label><input id="botonEnvio2" type="submit" value="Enviar"></label>
</form>
</body></html>
```

TODOS LOS CAMPOS DE FORMULARIOS EN ELEMENTS

Si prestamos atención a los campos de un formulario, cada uno de estos campos puede tener hasta dos identificadores importantes. Por ejemplo:

```
<input id="nombreFormContacto" type="text" name="nombre" />
```

Estos identificadores son el id, que identifica de forma única a un elemento dentro de un documento html (es decir, el id es único en toda la página web), y el atributo name, que identifica de forma única a un elemento dentro de un formulario (es decir, no habrá dos elementos dentro de un formulario con el mismo atributo name, pero sí puede haber elementos en distintos formularios que tengan el mismo atributo name).

Es frecuente que id y name tengan un mismo valor, por ejemplo `<input id="nombre" type="text" name="nombre" />`, pero no siempre ocurre así.

El atributo id, si existe, nos permitiría acceder a este elemento del formulario usando el método `getElementById('valorDeld')`.

La propiedad name posiblemente no nos sea útil para acceder con `getElementsByName` debido a que pueden existir varios elementos con el mismo name, sin embargo sí nos va a ser útil con métodos para trabajar con formularios de los que nos provee JavaScript de los que vamos a ir hablando.

La propiedad `elements` de cada objeto form obtenido mediante `document.forms` devuelve una colección de objetos tipo array conteniendo todos los campos de formulario que existan en el formulario concreto al que hagamos alusión, con índice 0 para el primer elemento por orden de aparición, índice 1 para el siguiente y así sucesivamente.

Cada campo de formulario es para JavaScript un "HTML Object" y como tal tiene propiedades a las que podemos acceder. Por ejemplo name es una propiedad, value otra propiedad, id otra propiedad, etc.

Analiza este código y comprueba sus resultados en tu navegador para ver cómo `elements` nos permite acceder a los campos de formulario.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css"> label{display:block;margin:5px;}</style>
<script type="text/javascript">
window.onload = function () {
    var ejemplo = document.getElementById('ejemplo');
    ejemplo.addEventListener("click", ejecutarEjemplo);
}

function ejecutarEjemplo () {
var msg = '';
var formularios = document.forms;
for (var i=0; i<formularios.length;i++){
    for (var j=0; j<formularios[i].elements.length; j++){
        msg = msg + '\n\nElemento '+j+ ' del formulario '+ (i+1)+ ' tiene id: ' + formularios[i].elements[j].id;
        msg = msg + ' y name: ' + formularios[i].elements[j].name;
    }
}
alert (msg);
}

</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue; margin:20px;" id="ejemplo">Pulsa aquí</div>

<form name="formularioContacto" method="get" action="accion1.html">
<h2>Formulario de contacto</h2>
<label>Nombre:<input id="nombreFormContacto" type="text" name="nombre" /></label>
<label>Apellidos:<input id="apellidosFormContacto" type="text" name="apellidos" /></label>
<label><input id="botonEnvio1" type="submit" value="Enviar"></label>
</form>
```

```
<form name = "formularioReclamacion" method="get" action="accion2.html">
<h2>Formulario de reclamación</h2>
<label>Motivo reclamación:<input id="motivoFormReclama" type="text" name="motivo" /></label>
<label>Fecha del hecho:<input id="fechaFormReclama" type="text" name="fecha" /></label>
<label><input id="botonEnvio2" type="submit" value="Enviar"></label>
</form>
</body></html>
```

El resultado esperado es que cuando pulsemos en el texto “Pulsa aquí” se muestre por pantalla lo siguiente:

```
Elemento 0 del formulario 1 tiene id: nombreFormContacto y name: nombre
Elemento 1 del formulario 1 tiene id: apellidosFormContacto y name: apellidos
Elemento 2 del formulario 1 tiene id: botonEnvio1 y name:
Elemento 0 del formulario 2 tiene id: motivoFormReclama y name: motivo
Elemento 1 del formulario 2 tiene id: fechaFormReclama y name: fecha
Elemento 2 del formulario 2 tiene id: botonEnvio2 y name:
```

Vemos cómo usando elements hemos accedido a cada uno de los elementos HTML dentro del formulario y mostrado sus propiedades. Vemos cómo los campos name correspondientes a los botones de envío quedan vacíos debido a que estos elementos input no tienen establecido atributo name.

ACCESO DIRECTO A FORMULARIOS MEDIANTE EL ATRIBUTO NAME

Aunque podamos acceder a cualquier elemento mediante el uso de `document.forms[indice1].elements[indice2]`, el acceso mediante índices numéricos posiblemente no resulta cómodo. Además, si introducimos nuevos elementos en los formularios o nuevos formularios, los índices numéricos cambiarán y esto afectaría al diseño de nuestro código.

JavaScript provee una forma cómoda de acceder a los formularios a la que denominamos “acceso directo al formulario a través de su atributo name”. Para ello nos basamos en que cada formulario se mantiene como una propiedad de document a la que se puede acceder escribiendo simplemente:

```
document.valorAtributoNameDelFormulario
```

De esta forma podemos obtener una referencia a un formulario simplemente con una expresión del tipo: `var formulario1 = document.formularioContacto;` donde `formularioContacto` es el atributo name de un formulario dentro del documento HTML.

ACCESO DIRECTO A ELEMENTOS DE FORMULARIOS MEDIANTE EL ATRIBUTO NAME

Del mismo modo que podemos acceder a un formulario usando el valor de su atributo name, podemos acceder a elementos de un formulario usando el atributo name de estos elementos con una sintaxis de este tipo:

```
document.valorAtributoNameDelFormulario.valorAtributoNameDelElementoHTML
```

De esta forma podemos obtener una referencia a un elemento simplemente con una expresión del tipo: `var elemento1 = document.formularioContacto.apellidos;` donde `formularioContacto` es el atributo name de un formulario dentro del documento HTML y `apellidos` es el atributo name de un elemento input dentro del formulario. Aunque en una primera lectura puede parecer un trabalenguas, leyéndolo un par de veces y viendo un ejemplo se comprenderá con facilidad.

Veamos el ejemplo. Escribe este código, guárdalo con un nombre como `ejemplo.html` y visualiza los resultados que se obtienen.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css"> label{display:block;margin:5px;}</style>
<script type="text/javascript">
window.onload = function () {      var ejemplo = document.getElementById('ejemplo');
    ejemplo.addEventListener("click", ejecutarEjemplo); }
function ejecutarEjemplo () {
var formulario1 = document.formularioContacto; dimeDatos(formulario1);
var formulario2 = document.formularioReclamacion; dimeDatos(formulario2);
}
function dimeDatos(formulario){
var msg=""; msg = msg + '\n\nElemento input del formulario '+formulario.name+ ' tiene id: '+ formulario.nombre.id;
msg = msg + '\n\nOtro elemento input del formulario '+formulario.name+ ' tiene id: '+ formulario.apellidos.id;
alert (msg+'\n\n');
}
</script></head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue; margin:20px;" id="ejemplo">Pulsa aquí</div>
<form name="formularioContacto" method="get" action="accion1.html">
<h2>Formulario de contacto</h2>
<label>Nombre:<input id="nombreFormContacto" type="text" name="nombre" /></label>
<label>Apellidos:<input id="apellidosFormContacto" type="text" name="apellidos" /></label>
<label><input id="botonEnvio1" type="submit" value="Enviar"></label>
</form>
<form name="formularioReclamacion" method="get" action="accion2.html">
<h2>Formulario de reclamación</h2>
<label>Nombre:<input id="nombreFormReclamacion" type="text" name="nombre" /></label>
<label>Apellidos:<input id="apellidosFormReclamacion" type="text" name="apellidos" /></label>
<label><input id="botonEnvio2" type="submit" value="Enviar"></label>
</form>
</body></html>
```

En este ejemplo `var formulario2 = document.formularioReclamacion`; nos devuelve el objeto HTML que es el form cuyo atributo `name` es `formularioReclamacion`. A su vez `document.formularioReclamacion.apellidos` nos devuelve el elemento HTML que es el input cuyo atributo `name` es `apellidos`.

Estos mismos resultados podrían conseguirse combinando los distintos métodos de acceso a nodos del DOM que ofrece JavaScript (`document.getElementById`, `getElementsByTagName`, `childNodes`, etc.) o los arrays `document.forms` y `document.elements`, pero el acceso basado en nombres resultará cómodo si hemos creado una buena definición de nombres en nuestro documento HTML.

ACCESO CON SINTAXIS MIXTA FORMS – ELEMENTS – NAME – ID

Hay más forma de acceder a los elementos de un formulario, e incluso al formulario en sí.

Podemos acceder a formularios o a los elementos de un formulario con una sintaxis como:

```
var formulario1 = document.forms['valorAtributoIdDelFormulario'];  
var elemento1 = formulario1.elements['valorAtributoIdDelElemento']
```

Por ejemplo `var formulario1 = document.forms['form1']`; nos permite acceder a un formulario cuyo atributo `id` es `form1`.

Igualmente podemos acceder a formularios o a los elementos de un formulario con una sintaxis como:

```
var formulario1 = document.forms['valorAtributoNameDelFormulario'];  
var elemento1 = formulario1.elements['valorAtributoNameDelElemento'];
```

Por ejemplo `var formulario1 = document.forms['formularioContacto']`; nos permite acceder a un formulario cuyo atributo `name` es `formularioContacto`.

Finalmente, un elemento de un formulario tiene una propiedad que es una referencia al formulario en el cual está inserto. La sintaxis sería del tipo:

```
var formulario1 = document.getElementById('valorIdDelCampo').form;
```

Suponiendo que un formulario tiene un campo como: `<label>Apellidos:<input id="apellidosFormContacto" type="text" name="apellidos" /></label>`, podemos acceder al formulario usando `document.getElementById('apellidosFormContacto').form`

EVENTOS RELACIONADOS CON FORMULARIOS

Ya hemos mencionado los eventos relacionados con formularios en apartados anteriores del curso, cuando tratamos los eventos. Recordar que los principales eventos son onfocus (un elemento toma el foco), onblur (pérdida del foco), onchange (cambio), onselect (selección de texto de un elemento input o textarea), onsubmit (pulsar botón de envío, antes del envío), y onreset (pulsar botón de cancelación).

EJERCICIO

Un uso habitual de JavaScript con formularios es usar JavaScript para validar que el contenido introducido por el usuarios sea válido. Crea un formulario que conste de cinco campos: nombre, apellidos, email, ciudad y país. Usando el evento onsubmit, realiza la validación para:

- a) Comprobar que en el momento del envío ninguno de los campos tiene menos de dos caracteres (es decir, si está vacío, contiene una letra o dos letras se considerará no válido) accediendo a los campos mediante document.forms y elements.
- b) Igual que el apartado a) pero accediendo a los campos directamente usando el atributo name (por ejemplo formularioContacto.apellidos haría alusión a un elemento input cuyo atributo name es apellidos en un formulario cuyo atributo name es formularioContacto)

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01176E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206