





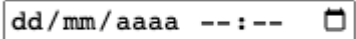


Formularios

Los formularios HTML son el mecanismo propuesto por este estándar para que el usuario pueda ingresar información y enviarla al destino que especifiquemos. En otras palabras, representa una sección del documento que actúa como contenedora de elementos interactivos que permiten el envío de información.

Se representa mediante la etiqueta `<form>`, que representa al formulario propiamente dicho, el cual a su vez se va construyendo mediante diversos controles o elementos representados mediante las etiquetas `<input>`, `<label>`, `<select>` o `<textarea>` (entre las más comunes), las cuales estudiaremos a continuación.

Etiqueta `<input>`

Esta etiqueta genera un control de formulario donde el usuario podrá ingresar o indicar un dato, siendo la más comúnmente usada y una de las más complejas, por la cantidad de variantes y atributos que puede contener. Dependiendo de su atributo `type` se generará uno u otro tipo de control de formulario.

Tabla de tipos de controles de formulario		
Valor de atributo <code>type</code>	Descripción	Vista gráfica
button	Botón sin comportamiento predefinido. Su etiqueta es el valor de su atributo <code>value</code>	
checkbox	Casilla de verificación que permite que ninguno, uno o más valores individuales se seleccionen o deseleccionen	
color	Muestra un selector de color	
date	Muestra un selector de fecha con día, mes y año	
datetime-local	Muestra un selector de fecha y hora	
email	Caja para ingreso de correo electrónico	
file	Permite el ingreso de un archivo; el atributo <code>accept</code> permite	

	restringir los tipos de archivo permitidos	
hidden	Control oculto pero cuyo valor es enviado al servidor	
month	Control para ingreso de mes y año	<input type="text" value="----- de -----"/>
number	Caja para ingreso de valor numérico	<input type="text"/>
password	Caja para ingreso de contraseña, no visible a simple vista	<input type="password"/>
radio	Botón de opción, que permite que un único valor sea seleccionado entre un grupo de opciones	<input type="radio"/>
range	Ingreso de un valor numérico, dentro de un tope de valores establecidos por los atributos min y max	<input type="range"/>
reset	Botón de reinicio del formulario; vuelve los controles a su valor predeterminado	<input type="button" value="Restablecer"/>
search	Caja de texto para el ingreso de cadenas de búsqueda	<input type="text"/>
submit	Botón de envío del formulario	<input type="button" value="Enviar"/>
tel	Caja para el ingreso de un número telefónico	<input type="text"/>
text	Caja de texto genérica	<input type="text"/>
time	Permite el ingreso de un valor horario	<input type="text" value="-- : --"/>
url	Ingreso de una dirección URL	<input type="text"/>
week	Ingreso de un valor de número de semana y año	<input type="text" value="Semana --, -----"/>

A su vez, cada tipo de control posee un conjunto de atributos que le son específicos y que controlan su funcionamiento, pero en general, los atributos (genéricos) que siempre debemos utilizar son:

Tabla de atributos para controles de formulario	
Atributo	Descripción
type	Define el tipo de elemento (explicados en la tabla más arriba)
name	Define el nombre del control, que permite identificarlo dentro del formulario
value	Define el dato que el control enviará al servidor, al momento de enviarse el formulario
id	Atributo general que define un identificador único para el elemento dentro del documento HTML. En el caso de los formularios, se usa para vincular un elemento del mismo a una etiqueta <label>

Etiqueta <textarea>

El elemento <textarea> representa un control donde el usuario puede ingresar texto plano en múltiples líneas. Es útil cuando se pretende que el usuario ingrese mayor cantidad de texto que lo permitido en, por ejemplo, un control <input> de tipo text.

Lo común es utilizar los atributos name, cols y rows en este elemento, siendo los dos últimos necesarios para mantener una visualización consistente en todos los navegadores.

```
Unset
<textarea name="review" cols="20" rows="20">Ingrese su contenido
aquí...</textarea>
```

Además, se diferencia de los demás elementos en que debe representarse con las correspondientes etiquetas de apertura y de cierre, siendo su contenido el valor que se enviará al servidor cuando el formulario sea enviado.

Etiqueta <select>

El elemento <select> representa un elemento que permite al usuario elegir una o varias opciones en un menú. Estas opciones se indican con el elemento <option>, el cual a su vez posee un atributo value que especifica el dato que el control enviará al servidor, si dicha opción

es la seleccionada. El atributo `selected` indica que dicha opción será la seleccionada por defecto.

Unset

```
<select name="dinosaurio">
  <option selected value="tiranosaurio">Tiranosaurio</option>
  <option value="triceratops">Triceratops</option>
  <option value="estegosaurio">Estegosaurio</option>
  <option value="brontosaurio">Brontosaurio</option>
  <option value="velociraptor">Velociraptor</option>
</select>
```

También posible agrupar las opciones mediante el uso de la etiqueta `<optgroup>`

Unset

```
<select name="dinosaurio">
  <optgroup label="Carnívoros">
    <option selected value="tiranosaurio">Tiranosaurio</option>
    <option value="velociraptor">Velociraptor</option>
  </optgroup>
  <optgroup label="Herbívoros">
    <option value="triceratops">Triceratops</option>
    <option value="estegosaurio">Estegosaurio</option>
    <option value="brontosaurio">Brontosaurio</option>
  </optgroup>
</select>
```

Por último el uso del atributo `multiple` en el elemento `<select>` permite que se seleccione más de una opción simultáneamente:

Unset

```
<select multiple name="dinosaurio">
  <option selected value="tiranosaurio">Tiranosaurio</option>
  <option value="triceratops">Triceratops</option>
  <option value="estegosaurio">Estegosaurio</option>
  <option value="brontosaurio">Brontosaurio</option>
  <option value="velociraptor">Velociraptor</option>
</select>
```

Formularios y Javascript

En los textos anteriores hemos trabajado el Modelo de Objetos del Documento, asociado también con un árbol que el navegador construye a partir de los elementos HTML que componen las páginas Web. Cada nodo de este árbol es entonces un objeto representativo de un elemento HTML, un atributo perteneciente a un elemento HTML o el texto contenido en un elemento HTML. Así, a cada nodo le corresponde determinado tipo de dato y responde a determinada interfaz. Por ejemplo, los nodos correspondientes a elementos HTML tendrán como interfaz a `HTMLElement` y los que correspondan a atributos tendrán la interfaz `Attr`.

Cada nodo que corresponda a un elemento HTML tendrá además una interfaz particular. Por ejemplo, los encabezados de cualquier nivel (`<h1>`, `<h2>`, etc) se corresponderán con la interfaz `HTMLHeadingElement`.

Por su parte, los formularios poseen la interfaz `HTMLFormElement`, la cual, obviamente, proporciona propiedades y métodos específicos para gestionar y trabajar con formularios desde un programa escrito, en nuestro caso, en lenguaje Javascript.

El navegador normalmente expone un objeto accesible mediante `document.forms`, el cual puede tratarse como arreglo o diccionario, que contiene todos los formularios presentes en el documento HTML.

Por ejemplo, dado el código HTML a continuación (donde a propósito se han omitido partes de la página):

Unset

```
<form method="GET" action="https://www.example.com/form.php"
name="form_datos_personales">
  <label for="nombre">Ingrese nombre: </label>
  <input type="text" name="nombre" id="nombre" autocomplete="off">
  <input type="submit" value="Enviar">
</form>
```

Podemos trabajar con el contenido del objeto antes dicho:

JavaScript

```
console.log(document.forms);
// HTMLCollection [form, form_datos_personales: form]

console.log(document.forms[0]);
// <form method="GET" action="https://www.example.com/form.php"
name="form_datos_personales">
```

```
console.log(document.forms['form_datos_personales']);  
// <form method="GET" action="https://www.example.com/form.php"  
name="form_datos_personales">  
  
console.log(document.forms.form_datos_personales);  
// <form method="GET" action="https://www.example.com/form.php"  
name="form_datos_personales">
```

Como puede observarse, el resultado para los tres últimos ejemplos es exactamente el mismo.

A su vez, cada formulario contiene un arreglo conteniendo los elementos de cada formulario. Este arreglo se denomina `elements`, y se accede así:

```
JavaScript  
console.log(document.forms['form_datos_personales'].elements);  
// HTMLFormControlsCollection(2) [input#nombre, input, nombre: input#nombre]  
  
console.log(document.forms['form_datos_personales'].elements[0]);  
// <input type="text" name="nombre" id="nombre" autocomplete="off">  
  
console.log(document.forms['form_datos_personales'].elements['nombre']);  
// <input type="text" name="nombre" id="nombre" autocomplete="off">
```

Mostrando diferentes maneras de hacer referencia a los elementos integrantes de un formulario.

La interfaz `HTMLFormElement` expone además un conjunto de métodos, que son:

- `reportValidity()`
Permite conocer si los datos que el usuario ha cargado en el formulario satisfacen las restricciones propuestas por sus controles (devuelve verdadero si el formulario es válido).
- `requestSubmit(submitter)`
Permite el envío del formulario utilizando un botón de envío específico
- `submit()`
Envía el formulario
- `reset()`
Reinicia el formulario: vuelve los controles a su valor predeterminado

Como dato adicional diremos que los elementos de un formulario (`<input>`) son capaces de disparar un evento `invalid` luego de enviado el formulario, o bien haberse invocado el método

`reportValidity()`. Este evento es útil para conocer las causas que provocaron el fallo del envío.