



APRENDERAPROGRAMAR.COM

GETELEMENTSBYNAME
JAVASCRIPT. EVENTO
ONSUBMIT AL ENVIAR UN
FORMULARIO FORM.
CHECKED. EJEMPLO
(CU01130E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº30 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

GETELEMENTSBYNAME JAVASCRIPT

El acceso a un nodo concreto del DOM usando `getElementById` es muy frecuente, pero hay otras maneras de acceder a los nodos del DOM. Una de ellas es usando `getElementsByName('valorNameBuscado')`. Esta función nos devuelve un array conteniendo todos los nodos DOM cuyo atributo `name` coincide con `valorNameBuscado`.



El orden en que aparecerán los elementos en el array (comenzando con índice cero) es el mismo en el que aparezcan en el código de la página web. Ejemplo:

```
var elementosAnimal = document.getElementsByName('animal');
```

Nos devolverá **un array con todos los nodos** de tipo `element` que tengan como atributo `html name` "animal", empezando con índice cero: `elementosAnimal[0]`, `elementosAnimal[1]`, `elementosAnimal[2]`, `elementosAnimal [3]` ... hasta el índice que sea necesario para abarcar tantos elementos como haya en el código con el atributo `name="animal"`. Podemos obtener los nodos de cualquier tipo de etiqueta que lleve el atributo `name` (`input`, `textarea`, `select`, `button`, `img`, etc.).

No todos los elementos HTML admiten que se les incluya un atributo `name`. Por ejemplo `<div name="animal"> ... </div>` no es correcto porque el atributo `name` no es aplicable a los elementos `div`.

Los elementos HTML que admiten el atributo `name` son: `button`, `form`, `fieldset`, `iframe`, `input`, `keygen`, `object`, `output`, `select`, `textarea`, `map`, `meta`, `param`.

A diferencia del atributo `id` que ha de ser único para un elemento dentro de una página web, el atributo `name` puede aparecer repetido en diversos elementos del código HTML de la página web. Un caso típico sería disponer varios `radiobuttons` cuyo atributo `name` es el mismo.

Siempre que sea posible elegir entre usar `getElementById` ó `getElementsByName` recomendamos usar `getElementById` pues aporta mayor seguridad.

Tener en cuenta que escribimos `document.getElementsByName('animal')` porque queremos empezar la búsqueda desde el nodo raíz del DOM, es decir, el nodo `document` (así exploramos todos los nodos). Podríamos empezar la búsqueda por otro nodo si resultara de interés.

EVENTO ONSUBMMIT

Ya hemos trabajado con algunos eventos como `onclick`, `onload`... añadimos ahora a nuestro repertorio un evento denominado `onsubmit`, que es el evento que se produce cuando el usuario pulsa el botón de envío de un formulario. Ejemplo:

