



APRENDERAPROGRAMAR.COM

SETATTRIBUTE Y
GETATTRIBUTE
JAVASCRIPT. DIFERENCIAS
DE SINTAXIS
PROPIEDADES CON HTML.
EJEMPLO MAXLENGTH.
(CU01176E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº76 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

FORMULARIOS. SETATTRIBUTE Y GETATTRIBUTE

JavaScript facilita el trabajo con formularios proveiéndonos de formas de acceso simplificadas a los formularios y sus elementos internos. Estas formas de acceso directo son ligeramente distintas a lo que es el acceso al resto de nodos del DOM, aunque al fin y al cabo lo que logramos es lo mismo: acceder a un nodo.



En su momento describimos el DOM, y ahora creemos oportuno recordar algunas cosas que ya habíamos estudiado para refrescarlas.

Los principales tipos de nodos en el DOM son:

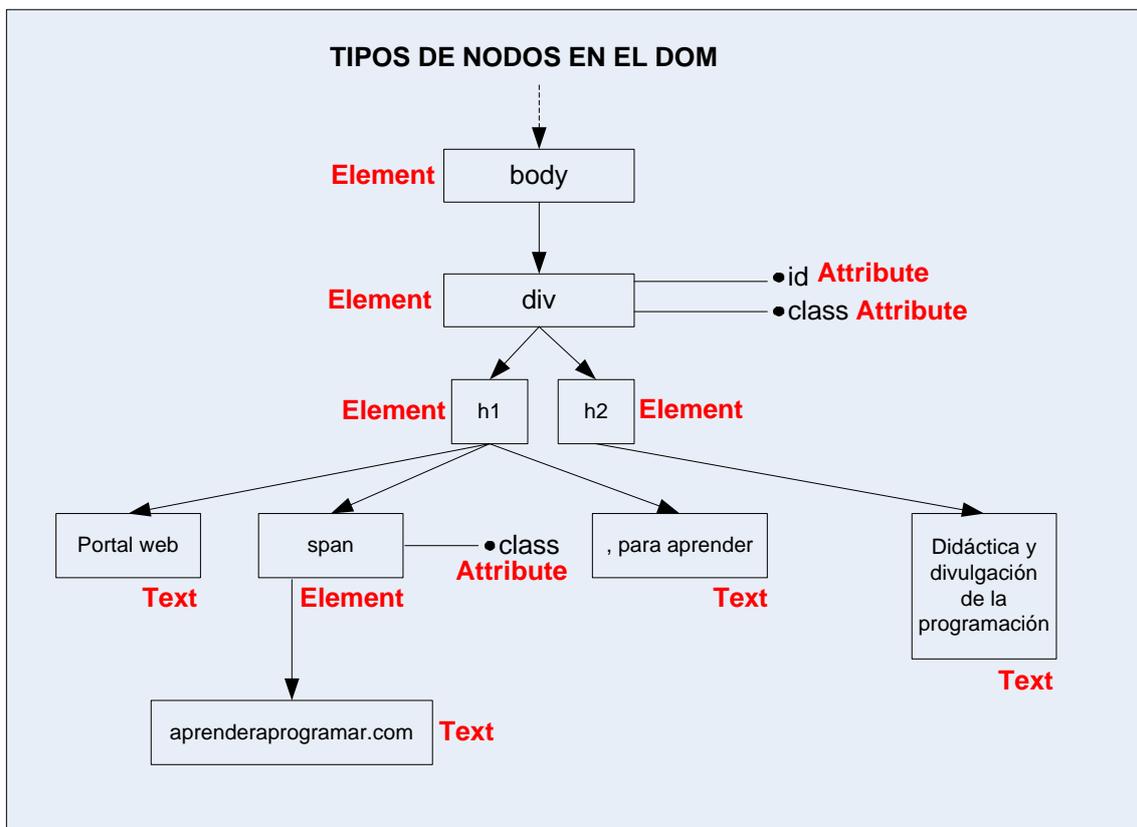
- **Document:** el nodo document es el nodo raíz, a partir del cual derivan el resto de nodos.
- **Element:** son los nodos definidos por etiquetas html. Por ejemplo una etiqueta div genera un nodo. Si dentro de ese div tenemos tres etiquetas p, dichas etiquetas definen nodos hijos de la etiqueta div. Un formulario y los elementos de un formulario son nodos element.
- **Text:** el texto dentro de un nodo element se considera un nuevo nodo hijo de tipo text (texto). Los navegadores también crean nodos tipo texto sin contenido para representar elementos como saltos de línea o espacios vacíos.
- **Attribute:** los atributos de las etiquetas definen nodos, aunque trabajando con JavaScript no los veremos como nodos, sino que lo consideramos información asociada al nodo de tipo element. Para trabajar con formularios nos va a interesar conocer los atributos de los formularios y de los elementos de los formularios. De esta forma podremos recuperarlos para realizar verificaciones o ejecutar condicionales, o modificarlos.
- **Comentarios y otros:** los comentarios y otros elementos como las declaraciones doctype en cabecera de los documentos HTML generan nodos.

Existen más tipos de nodos en el DOM, pero de uso más infrecuente.

Un fragmento de código como este:

```
<body>
<div id="cabecera" class="brillante">
<h1>Portal web <span class="destacado">aprenderaprogramar.com</span>, para aprender</h1>
<h2>Didáctica y divulgación de la programación</h2>
</div>
</body>
```

Se puede ver como un árbol de nodos del DOM:



Una etiqueta como la etiqueta input de un formulario genera un nodo. Sus atributos id, name, size, etc. diremos que son propiedades del nodo. Por ejemplo supongamos un elemento input de tipo texto. Si escribimos algo como `var nodo = document.getElementById('nombreFormularioContacto')` nos devuelve un nodo, y escribiendo `nodo.maxlength` podemos recuperar o establecer un valor para esta propiedad del nodo.

No todos los nodos de tipo element tienen las mismas propiedades disponibles. Por ejemplo un nodo de tipo input de un formulario que es un campo de texto dispondrá de un atributo `maxlength`, pero un nodo `span` no dispondrá de este atributo. Las propiedades de los nodos están por tanto ligadas a la definición del estándar HTML, que para cada elemento HTML define una serie de atributos permitidos.

Por tanto, para trabajar con estas propiedades en JavaScript, tendremos que conocer o remitirnos al estándar HTML para saber a qué propiedades podemos acceder.

En general las propiedades son accesibles y modificables mediante JavaScript.

HTML FRENTE A DOM Y `maxlength` FRENTE A `maxLength`

Vamos ahora a pensar en los elementos input de los formularios. Estos elementos si son de tipo text tienen como atributo opcional una longitud máxima en número de caracteres. Por ejemplo `<input id="nombreFormContacto" type="text" name="nombre" maxlength="4"/>` indica que para este input el número máximo de caracteres que podrá introducir el usuario por teclado es 4.

Supongamos que usamos `nodo = document.getElementById('nombreFormContacto')` para recuperar el nodo del DOM correspondiente a este elemento. Este nodo tiene atributos como `type` que podemos recuperar como `nodo.type`. Igualmente podríamos pensar que `nodo.maxLength` nos debería devolver el valor del atributo `maxLength` del elemento. Sin embargo no ocurre así, y tenemos que escribir `nodo.maxLength` con la L mayúscula intermedia para poder recuperar el valor del atributo con JavaScript. ¿Por qué?

Pues básicamente por el mismo motivo por el que no podemos acceder a las propiedades CSS de los elementos con exactamente los mismos nombres en JavaScript y en CSS. Por ejemplo, podemos recordar que en CSS escribimos `background-color: red`; mientras que en JavaScript usamos una sintaxis como `nodo.style.backgroundColor = 'red'`;

JavaScript nos da acceso al DOM y a la manipulación de elementos del DOM, pero no se ciñe siempre a la sintaxis especificada por estándares como CSS ó HTML.

Muchas propiedades se escriben igual en HTML que con JavaScript. Por ejemplo la propiedad `type` de HTML encuentra su equivalente en una sintaxis de tipo `nodo.type`. Pero no siempre ocurre así.

En el caso de `maxLength`, para acceder a esta propiedad a través de JavaScript hemos de escribir `maxLength`. En general podemos decir que se sigue la sintaxis "camelCase", pero siempre debemos hacer comprobaciones porque puede haber casos especiales.

Escribe este código y comprueba sus resultados:

```
<!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++){
        if (formularios[i].elements[j].type=='text' ) {
            msg = msg + '\n\nValor inicial maxLength: '+formularios[i].elements[j].maxLength;
            msg = msg + '\n\nFijada longitud máxima 5 para elemento con id: '+formularios[i].elements[j].id;
            formularios[i].elements[j].maxLength=5;
        }
    }
}
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" maxlength="4"/></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 que hemos obtenido con este código ha sido que después de pulsar sobre “Pulsa aquí” las casillas de texto quedan con un máximo de 5 caracteres admitidos para entrada y que por pantalla se muestra los siguiente:

Navegadores 1 y 2	Navegador 3
Valor inicial maxlength: 4	Valor inicial maxlength: 4
maxlength ahora 5 para: nombreFormContacto	maxlength ahora 5 para: nombreFormContacto
Valor inicial maxlength: -1	Valor inicial maxlength: 2147483647
maxlength ahora 5 para: apellidosFormContacto	maxlength ahora 5 para: apellidosFormContacto
Valor inicial maxlength: -1	Valor inicial maxlength: 2147483647
maxlength ahora 5 para: motivoFormReclama	maxlength ahora 5 para: motivoFormReclama
Valor inicial maxlength: -1	Valor inicial maxlength: 2147483647
maxlength ahora 5 para: fechaFormReclama	maxlength ahora 5 para: fechaFormReclama

Lo que nos llama la atención de este resultado es: cómo distintos navegadores devuelven un resultado distinto cuando maxlength no está establecida (en unos casos obtenemos -1, que interpretamos como una señal de valor no establecido, y en otros 2147483647 que interpretamos como el valor más grande para un entero, que interpretamos como que no existe límite en el número de caracteres que se pueden introducir en el input.

Prueba ahora a sustituir en el código anterior maxLength por maxlength. Abre la consola de depuración de tu navegador y comprueba cómo no aparece ningún error. Sin embargo, después de hacer click en “Pulsa aquí” los input siguen sin cambiar el límite de caracteres admitido. ¿Por qué? En este caso, lo que estaríamos haciendo al invocar formularios[i].elements[j].maxlength sería invocar una propiedad que no existe sobre un objeto, y esto nos devuelve undefined, que es precisamente lo que se muestra por pantalla. Al escribir formularios[i].elements[j].maxlength=5; no estamos modificando el atributo que limita el máximo número de caracteres admisible, sino que estamos creando un nuevo atributo al que hemos denominado maxlength con minúsculas, y que hemos establecido con valor 5. Pero esto es lo mismo que si hubiéramos escrito formularios[i].elements[j].lechugas = 5; ya que podemos escribir una nueva propiedad de un objeto y asignarle un valor en el momento en que queramos.

ENCONTRAR LAS EQUIVALENCIAS ENTRE HTML Y JAVASCRIPT

El HTML y el propio JavaScript son realidades relativamente cambiantes: a medida que surgen nuevos estándares, nuevas versiones de navegadores, van apareciendo nuevas formas sintácticas y propiedades, por lo que es casi imposible encontrar un lugar donde tengamos todas las propiedades HTML y sus equivalentes para JavaScript.

Nos encontramos ante un pequeño problema: ¿cómo encontrar las equivalencias entre la sintaxis HTML y la sintaxis JavaScript?

De forma orientativa podemos seguir las siguientes reglas:

- 1) Escribe la propiedad en JavaScript de la misma forma que lo harías en HTML. Esto es válido para la gran mayoría de propiedades como value, src, type, etc.
- 2) Si no funciona lo anterior usa sintaxis camelcase, por ejemplo maxLength para JavaScript frente a maxlength para HTML.
- 3) Si tienes dudas sobre la sintaxis o existencia de una propiedad en HTML, busca la referencia HTML correspondiente. Por ejemplo, si quieres utilizar un elemento input y conocer sus propiedades válidas en HTML puedes recurrir a un libro o a una búsqueda en internet donde buscaríamos <<input HTML>> ó <<input HTML mozilla developer>>. En este caso hemos optado por incluir <<mozilla developer>> en la búsqueda para acceder a la documentación que facilita la fundación Mozilla al respecto, que suele ser de buena calidad en líneas generales. Por ejemplo, encontraríamos que nos aparece maxLength y nos indica "If the value of the type attribute is text, email, search, password, tel, or url, this attribute specifies the maximum number of characters (in Unicode code points) that the user can enter"
- 4) Para conocer el equivalente para JavaScript nos hemos de remitir a la especificación del DOM. Por tanto haríamos una búsqueda en internet del texto <<html input element dom>> ó <<html input element dom mozilla developer>>. En los resultados de búsqueda comprobaremos que existe la propiedad maxLength, de la que se indica "Reflects the maxlength HTML attribute, containing the maximum length of text..."

SETATTRIBUTE Y GETATTRIBUTE

JavaScript provee más formas de acceder y modificar los atributos HTML relacionados con un nodo. Vamos a ver una de ellas. Podemos usar esta sintaxis para establecer el valor de un atributo:

```
nodo.setAttribute('nombreAtributoHTML', 'valorAtributoHTML');
```

O esta otra sintaxis para recuperar el valor del atributo:

```
nodo.getAttribute('nombreAtributoHTML');
```

En general se recomienda el acceso directo a las propiedades, por ejemplo `nodo.type = 'text'`; en lugar de usar `nodo.setAttribute('type', 'text')`; pero la alternativa está ahí y debe conocerse.

En este caso, donde se indica nombreAtributoHTML tendremos que usar la forma de nombrar el atributo con HTML. Por ejemplo si nos referimos a maxlength, habremos de escribirlo todo en minúsculas.

EJERCICIO

Dado el siguiente código 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>
</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" maxlength="4"/></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>
```

Crea un script que recorra todos los elementos input dentro de cada uno de los formularios presentes y si son de tipo text, modifique su atributo asociado maxlength usando el método setAttribute de los objetos tipo Element para limitar a 8 el número máximo de caracteres que pueda introducir el usuario. Una vez modificado el atributo, muestra por pantalla el valor que tiene dicho atributo para todos los elementos de tipo input usando el método getAttribute.

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

Próxima entrega: CU01177E

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