



APRENDERAPROGRAMAR.COM

FUNCIÓN EVAL  
JAVASCRIPT.  
GETELEMENTSBY  
CLASSNAME . EJEMPLO  
CÓDIGO CALCULADORA  
SIMPLE (CU01133E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº33 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

## FUNCIÓN EVAL. GETELEMENTSBYCLASSNAME.

La función eval tiene diferentes aplicaciones en JavaScript. Veremos la aplicación más básica y comentaremos algunos aspectos de esta función cuyo uso es poco recomendado. Veremos también una posibilidad adicional para seleccionar nodos según el valor de su atributo class. La sintaxis a emplear es getElementByClassName. Se utiliza ClassName en lugar de Class porque class es una palabra reservada JavaScript.



## FUNCIÓN EVAL.

La función eval tiene diferentes aplicaciones en JavaScript. Algunos expertos en programación desaconsejan directamente su uso indicando que debe evitarse su utilización (es famosa la frase de Douglas Crockford “eval is evil”, eval es el demonio). Otros expertos indican que es una herramienta que bien usada puede ser muy útil. Nosotros no vamos a decantarnos por una u otra postura. Simplemente expondremos algunas cuestiones básicas sobre esta función.

La sintaxis de la función es la siguiente:

```
eval ('cadenaDeTextoPasadaALaFuncion');
```

El valor de cadenaDeTextoPasadaALaFuncion puede ser:

a) Una expresión JavaScript. Por ejemplo '3+2'. En este caso eval('3+2') evalúa la expresión y devuelve el valor 5, es decir, el resultado de evaluar la expresión. Podríamos escribir valorOperacion = eval('3+2'); y valorOperacion tomaría el valor 5.

b) Una sentencia o fragmento de código JavaScript. Por ejemplo:

eval('moverImagen(\'adelante\')'); dará lugar a que se ejecute la llamada a la función moverImagen('adelante') y que se devuelva, si existe, un valor de retorno. Fíjate que para poder incluir la comilla simple dentro de la cadena de texto hemos usado el escape del carácter con el símbolo \. También sería válido usar esta sintaxis:

```
eval("moverImagen('adelante')");
```

Como decimos eval ('cadenaDeTextoPasadaALaFuncion'); tiene dos implicaciones: una evaluación o ejecución, y la obtención de un valor de retorno (si existe). De este modo la función eval puede usarse como argumento para otras funciones, por ejemplo:

```
alert(Hemos obtenido un resultado: ' + eval("moverImagen('adelante')"));
```

El resultado será el contenido de la sentencia `return` de la función si existía. En caso de no existir un valor de retorno este será un valor no válido como `undefined`.

Veamos un ejemplo un poco más elaborado donde se utiliza `eval` para hacer una llamada a una función:

```
function componer(sentido) {
  var accion = 'mover';
  var sobreElemento = 'Imagen';
  var haciaDonde = sentido;
  var llamada = accion + sobreElemento+'('+haciaDonde+')';
  alert ('La unión del texto genera ' + llamada);
  eval(llamada);
}
```

En este ejemplo hemos compuesto un texto a partir de las partes “mover”, “Imagen”, y un parámetro “sentido”. Si `sentido` es igual a `'adelante'` por ejemplo, el resultado es que se ejecutará `eval(moverImagen('adelante'))`; y por tanto se ejecuta la función `moverImagen('adelante')`

Esto nos permite invocar funciones compuestas a partir de fragmentos, donde cada fragmento puede estar determinado por condicionales o por otras funciones.

Se señalan distintos inconvenientes para la función `eval`:

- Genera código difícil de leer, donde se mezcla texto con variables.
- Resulta de ejecución pesada y lenta para el intérprete, pudiendo dar lugar a que nuestra aplicación web responda mal o lentamente.
- Puede poner en riesgo la seguridad de las aplicaciones, ya que si por ejemplo introducimos en `eval` una cadena proporcionada por un usuario malicioso, éste puede intentar que se ejecute código que comprometa la seguridad.

Iremos conociendo alternativas al uso de `eval`, pero de momento señalaremos que el siguiente código genera el mismo resultado que el anterior evitando el uso de `eval`:

```
function componer(sentido) {
  var accion = 'mover';
  var sobreElemento = 'Imagen';
  var haciaDonde = sentido;
  var llamada = accion + sobreElemento+'('+haciaDonde+')';
  alert ('La unión del texto genera ' + llamada);
  var tmpFunc = new Function(llamada);
  tmpFunc();
}
```

## GETELEMENTSBYCLASSNAME

El acceso a un nodo concreto del DOM usando getElementById es muy frecuente, pero hay otras maneras de acceder a los nodos del DOM. Ya conocemos algunas alternativas. Otra de ellas es usar getElementByClassName('valorClassBuscado'). Esta función nos devuelve un array conteniendo todos los nodos DOM cuyo atributo class coincide con valorClassBuscado.

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 elementosDestacado = document.getElementsByClassName('destacado');
```

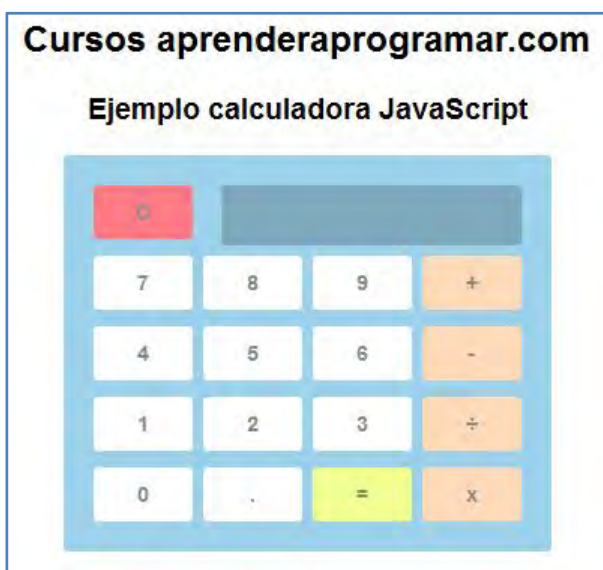
Nos devolverá **un array con todos los nodos** de tipo element que tengan como atributo html class "destacado", empezando con índice cero: elementosDestacado[0], elementosDestacado[1], elementosDestacado[2], elementosDestacado[3] ... hasta el índice que sea necesario para abarcar tantos elementos como haya en el código con el atributo class="destacado". Podemos obtener los nodos de cualquier tipo de etiqueta que lleve el atributo class (div, span, input, etc.).

A diferencia del atributo id que ha de ser único para un elemento dentro de una página web, el atributo class puede aparecer repetido en diversos elementos del código HTML de la página web. Normalmente se usa para identificar partes de la página web a la que deben aplicársele las mismas reglas CSS.

Tener en cuenta que escribimos document.getElementsByClassName('destacado') 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.

## CREAR UNA CALCULADORA SIMPLE CON JAVASCRIPT

A continuación vamos a estudiar un ejemplo de código donde creamos una calculadora simple usando JavaScript con la sintaxis getElementByClassName y la función eval para realizar los cálculos.



Hemos incluido código CSS para crear una visualización agradable. Así, CSS se encarga principalmente del aspecto y JavaScript se encarga principalmente de la parte dinámica (respuesta ante las acciones del usuario). 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">
body {font-family: sans-serif; text-align:center;}
#calculadora { font: bold 14px Arial,sans-serif; background-color: #9DD2EA; border-radius: 3px; height: auto; margin: 0 auto;
padding: 20px 20px 9px; width: 285px;}
.parteSuperior .pantalla { background-color: rgba(0, 0, 0, 0.2); border-radius: 3px; color: #FFFFFF; float: right; font-size: 17px;
height: 40px; letter-spacing: 1px; line-height: 40px; padding: 0 10px; text-align: right; width: 180px;}
.teclas, .parteSuperior {overflow: hidden; }
.teclas span, .parteSuperior span.limpiar { background-color: #FFFFFF; border-radius: 3px; color: #888888; cursor: pointer;
float: left; height: 36px; line-height: 36px; margin: 0 7px 11px 0; text-align: center; transition: all 0.4s ease 0s; width: 66px;}
.parteSuperior span.limpiar { background-color:#FF7C87;}
.teclas span.operador { background-color: #FFDAB9; margin-right: 0; }
.teclas span.igual { background-color: #F1FF92; color: #888E5F;}
.parteSuperior span.limpiar, { background-color: #FF9FA8; color: #FFFFFF; }
.teclas span:hover, .teclas span.igual:hover, .parteSuperior span.limpiar:hover {
background-color: #9C89F6; color: #FFFFFF;
}
</style>
<script type="text/javascript">
function pulsada (tecla) {
var listaNodosPantalla = document.getElementsByClassName('pantalla');
var nodoTextoPantalla = listaNodosPantalla[0].firstChild;
switch(tecla) {
case 'C':
nodoTextoPantalla.nodeValue = ' ';
break;
case '=':
var resultado = eval(nodoTextoPantalla.nodeValue);
nodoTextoPantalla.nodeValue = resultado;
break;
default:
nodoTextoPantalla.nodeValue = nodoTextoPantalla.nodeValue + tecla;
break;
}
}
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplo calculadora JavaScript</h3></div>
<!-- Calculadora -->
<div id="calculadora">
<!-- pantalla y tecla limpiar -->
<div class="parteSuperior">
<span class="limpiar" onclick="pulsada('C')">C</span>
<div class="pantalla"> </div> <!--Ojo tenemos un espacio para que exista nodeValue-->
</div>
<div class="teclas">
<!-- operadores y otras teclas -->
<span onclick="pulsada('7')">7</span>
<span onclick="pulsada('8')">8</span>

```

```
<span onclick="pulsada('9')">9</span>
<span class="operador" onclick="pulsada('+')">+</span>
<span onclick="pulsada('4')">4</span>
<span onclick="pulsada('5')">5</span>
<span onclick="pulsada('6')">6</span>
<span class="operador" onclick="pulsada('-')">-</span>
<span onclick="pulsada('1')">1</span>
<span onclick="pulsada('2')">2</span>
<span onclick="pulsada('3')">3</span>
<span class="operador" onclick="pulsada('/')">÷</span>
<span onclick="pulsada('0')">0</span>
<span onclick="pulsada('.')">.</span>
<span class="igual" onclick="pulsada('=')">=</span>
<span class="operador" onclick="pulsada('*')">x</span>
</div>
</div>
</body>
</html>
```

De este código podemos comentar lo siguiente. Debes ser capaz de comprender cómo el código CSS da lugar a la visualización de la calculadora. Debes ser capaz de comprender por qué la invocación listaNodosPantalla[0].firstChild nos permite acceder al nodo de texto con el contenido de la pantalla de la calculadora. Debes ser capaz de comprender cómo manipulamos el contenido de la pantalla accediendo al nodo texto y cómo modificamos su contenido y mostramos resultados usando la función eval. Si tienes dudas consulta en los foros aprenderaprogramar.com

## EJERCICIOS

- 1) Modifica el código del ejemplo de la calculadora javascript para que en lugar del condicional switch uses el condicional if (con if else ó if else if cuando sea necesario).
- 2) Utilizando el ejemplo de código que usa la función eval en la función <<function componer(sentido)>> que hemos visto dentro del código de los ejemplos, crea una página web donde existan dos botones que pongan “Adelante” y “Atrás”, de modo que cuando se pulsen den lugar a que se llame la función componer(sentido). En esta función, a través de la función eval se ejecutará moverImagen('adelante') ó moverImagen('atras') según el parámetro recibido. La función moverImagen(sentido) puede contener un simple mensaje indicando el valor del parámetro <<sentido>> recibido.

Para comprobar si tus respuestas son correctas puedes consultar en los foros aprenderaprogramar.com.

**Próxima entrega:** CU01134E

**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](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)