



APRENDERAPROGRAMAR.COM

EJEMPLOS JAVASCRIPT
OPERADORES
INCREMENTO Y
DECREMENTO.
ASIGNACIÓN Y
ASIGNACIÓN
COMPUESTA. (CU01118E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº18 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

OPERADORES EXTRA EN JAVASCRIPT

En JavaScript disponemos además de la asignación tradicional basada en = de algunos operadores menos habituales como incremento unitario (++), decremento unitario (--), asignación compuesta de suma (+=) , asignación compuesta de resta (-=) y asignación compuesta de multiplicación .



OPERADORES	DESCRIPCIÓN	EJEMPLO	EQUIVALE A
=	Asignación	a = 5	
+=	Suma lo indicado	a += b	a = a + b
-=	Resta lo indicado	a -= b	a = a - b
*=	Multiplica por lo indicado	a *= b	a = a * b
%=	Calcula el módulo por lo indicado	a %= b	a = a % b
++ (anterior)	Incremento unitario antes de operar	++a * 2	a = a + 1 a * 2
++ (posterior)	Incremento unitario después de operar	a++ * 2	a * 2 a = a + 1
-- (anterior)	Decremento unitario antes de operar	--a * 2	a = a - 1 a * 2
-- (posterior)	Decremento unitario después de operar	a-- * 2	a * 2 a = a - 1

Operadores en JavaScript

El operador `=` es el operador de asignación y hay que tener bien claro que no sirve para realizar comparaciones. Para realizar comparaciones ha de usarse `==` (es igual a) ó `===` (es estrictamente igual a). La asignación `a = b` se lee: "asigna a a el contenido de b". Si b es una operación o expresión lógica, a almacenará el valor numérico resultado de la operación o el valor booleano resultado de evaluar la expresión lógica. Por ejemplo `a = 3 > 5` implicará que a vale `false` porque `3 > 5` es falso.

`++` y `--` son sólo válidos para variables numéricas y sirven para incrementar una unidad el valor de la variable. Dependiendo de dónde se coloquen (antes o después de la variable) el resultado del cálculo puede diferir debido al momento en que se ejecuta la adición de la unidad.

Los operadores `+=`, `-=` y `*=` son formas abreviadas de escribir operaciones habituales.

Tener en cuenta que `++`, `--`, `+=`, `-=` y `*=` son expresiones que siempre se aplican sobre variables. Por ejemplo no es válido escribir `2++` porque 2 no es una variable. Todas estas operaciones pueden sustituirse por otra equivalente más evidente. Muchos programadores prefieren no usar estos operadores porque hacen menos legible el código. A otros programadores les gusta usarlos porque les ahorra escribir. Nosotros preferimos no usarlos, pero es cierto que los puedes encontrar cuando tengas que revisar el código escrito por otra persona.

EJEMPLO

Veamos un ejemplo. Vamos a usar dos pequeñas novedades.

Una vez pedido un número al usuario usaremos `nombreVariable = Number (nombreVariable);` para indicarle a JavaScript que considere la variable como de tipo numérico y no de tipo texto. De esta forma al usar el operador `+` se ejecutará la operación de suma en lugar de la operación de concatenación de cadenas de texto.

Además usaremos la sentencia `document.write('cadena de texto a introducir como HTML en el documento');` para introducir HTML en el documento. Hablaremos de esta sentencia más adelante, de momento sólo nos interesa saber que sirve para introducir código HTML en el documento.

Fíjate cómo en este ejemplo se usan tres scripts cuya ejecución no está en función de la ocurrencia de un evento como un click de un usuario, sino que se ejecutan automáticamente cuando carga la página web, en un determinado orden.

Escribe el siguiente código y guárdalo en un archivo de extensión html:

```

<html> <head> <title>Curso JavaScript aprenderaprogramar.com</title> <meta charset='utf-8'>
<style type='text/css'>
body {background-color: #FAEBD7; font-family: sans-serif; line-height: 1.3;}
div {border-style: solid; margin: 15px; padding: 10px; float: left;}
</style>
<script type='text/javascript'>
var a = prompt('Introduzca un número entero'); var b = prompt('Introduzca otro número entero');
a = Number(a) //Si no hacemos esto a es un texto y a+b concatena el texto
b = Number(b)
</script>
</head>
<body>
<div> <h2>Operadores de incremento</h2>
<script type='text/javascript'>
var valorInicial_a = a; document.write('Valores iniciales: a = ' + a + ', b = ' + b + '<br/> <br/>');
aumentar = ++a * b; document.write ('Operador ++ (anterior): ++a * b == ' + aumentar + ' <br/>');
document.write ('(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = valorInicial_a; aumentar = a++ * b;
document.write ('Operador ++ (posterior): a++ * b == ' + aumentar + ' <br/>');
document.write ('(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = valorInicial_a; //Restablecemos el valor que inicialmente tenía a
disminuir = --a * b; document.write ('Operador -- (anterior): --a * b == ' + disminuir + ' <br/>');
document.write ('(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = valorInicial_a;
disminuir = a-- * b;
document.write ('Operador -- (posterior): a-- * b == ' + disminuir + ' <br/>');
document.write ('(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = valorInicial_a;
</script>
</div>
<div>
<h2>Operadores de asignación compuestos</h2>
<script type='text/javascript'>
document.write('Valores iniciales: a = ' + a + ', b = ' + b + '<br/> <br/>');
inicio = a;
a += b;
document.write ( 'Asignación compuesta de suma: a += b equivale a = a + b<br/>');
document.write ( '(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = inicio; a -= b; document.write ( 'Asignación compuesta de resta: a -= b equivale a = a - b<br/>');
document.write ( '(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = inicio; a *= b;
document.write ( 'Asignación compuesta de multiplicación: a *= b equivale a = a * b<br/>');
document.write ( '(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = inicio; a /= b;
document.write ( 'Asignación compuesta de división: a /= b equivale a = a / b<br/>');
document.write ( '(Ahora el valor de a es: ' + a + ')<br/><br/>');
a = inicio; a %= b;
document.write ( 'Asignación compuesta de módulo: a %= b equivale a = a % b<br/>');
document.write ( '(Ahora el valor de a es: ' + a + ')<br/><br/>');
</script> </div> </body> </html>

```

Visualiza el resultado y comprueba que la página web se muestra con normalidad y que JavaScript se ejecuta con normalidad.

El resultado esperado es que se muestre lo siguiente (compruébalo introduciendo 4 y 2 como números):

Operadores de incremento

Valores iniciales: $a = 4$, $b = 2$

Operador ++ (anterior): $++a * b == 10$

(Ahora el valor de a es: 5)

Operador ++ (posterior): $a++ * b == 8$

(Ahora el valor de a es: 5)

Operador -- (anterior): $--a * b == 6$

(Ahora el valor de a es: 3)

Operador -- (posterior): $a-- * b == 8$

(Ahora el valor de a es: 3)

Operadores de asignación compuestos

Valores iniciales: $a = 4$, $b = 2$

Asignación compuesta de suma: $a += b$ equivale $a = a + b$

(Ahora el valor de a es: 6)

Asignación compuesta de resta: $a -= b$ equivale $a = a - b$

(Ahora el valor de a es: 2)

Asignación compuesta de multiplicación: $a *= b$ equivale $a = a * b$

(Ahora el valor de a es: 8)

Asignación compuesta de división: $a /= b$ equivale $a = a / b$

(Ahora el valor de a es: 2)

Asignación compuesta de módulo: $a %= b$ equivale $a = a \% b$

(Ahora el valor de a es: 0)

EJERCICIO 1

a) Describe paso a paso lo que hace el código anterior, indicando en qué orden se carga cada cosa (instrucciones HTML, instrucciones CSS, instrucciones JavaScript). Por ejemplo, ¿se carga el html `<h2>Operadores de incremento</h2>` en el instante en que solicitamos al navegador que muestre la página web? ¿En qué orden se ejecutan los scripts?

b) Introduce errores en los scripts de modo que estos no se ejecuten ¿Qué se visualiza en la página web?

c) Introduce 0 y 0 como números en el ejemplo anterior. ¿Cuántas veces el resultado de las operaciones es NaN? ¿Por qué?

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

Próxima entrega: CU01119E

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