



APRENDERAPROGRAMAR.COM

OPERADORES LÓGICOS
JAVASCRIPT. EJEMPLOS.
RELACIONALES MAYOR,
MENOR, IGUAL, DISTINTO.
AND, OR, NOT. CORTO-
CIRCUITO (CU01117E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº17 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

OPERADORES LÓGICOS PRINCIPALES EN JAVASCRIPT

En JavaScript disponemos de los operadores lógicos habituales en lenguajes de programación como son “es igual”, “es distinto”, menor, menor o igual, mayor, mayor o igual, and (y), or (o) y not (no). La sintaxis se basa en símbolos como veremos a continuación y cabe destacar que hay que prestar atención a no confundir `==` con `=` porque implican distintas cosas.



OPERADORES LÓGICOS Y RELACIONALES	DESCRIPCIÓN	EJEMPLO
<code>==</code>	Es igual	<code>a == b</code>
<code>===</code>	Es estrictamente igual	<code>a === b</code>
<code>!=</code>	Es distinto	<code>a != b</code>
<code>!==</code>	Es estrictamente distinto	<code>a !== b</code>
<code><, <=, >, >=</code>	Menor, menor o igual, mayor, mayor o igual	<code>a <= b</code>
<code>&&</code>	Operador and (y)	<code>a && b</code>
<code> </code>	Operador or (o)	<code>a b</code>
<code>!</code>	Operador not (no)	<code>!a</code>

Operadores lógicos y relacionales principales en JavaScript

La sintaxis coincide con la empleada en otros lenguajes de programación.

Además de los operadores habituales existen los operadores `===` que se interpreta como “es estrictamente igual” y `!==` que se interpreta como “no es estrictamente igual”. Estos operadores resultan un poco más complejos de comprender por lo que volveremos a hablar de ellos más adelante. De momento tener en cuenta que si una variable contiene `texto1 = "1"` y hacemos la comparación `texto1 === 1`, obtendremos `false`, es decir, que no es igual (porque un texto no es igual a un número). Sin embargo una comparación como `texto == 1` devolverá `true` ya que esta comparación no es estricta y

trata de realizar automáticamente conversiones para comprobar si se puede establecer una equivalencia entre los dos valores. En este caso se busca el equivalente numérico del texto y luego se hace la comparación, motivo por el cual se obtiene true.

Las expresiones donde se utilizan operadores lógicos y relacionales devuelven un valor booleano, es decir, verdadero (true) o falso (false). Por ejemplo si $a = 7$ y $b = 5$ la expresión $a < b$ devuelve false (es falsa). Si $a = \text{true}$ y $b = \text{false}$ la expresión $a \&\& b$ devuelve false (es falsa porque no se cumple que a y b sean verdaderas). Si $a = \text{true}$ y $b = \text{false}$ la expresión $a \|\| b$ devuelve true porque uno de los dos operandos es verdadero. Si $a = \text{true}$ la expresión $!a$ devuelve false (el opuesto o contrario).

El operador $\|\|$ se obtiene en la mayoría de los teclados pulsando ALT GR + 1, es decir, la tecla ALT GR y el número 1 simultáneamente.

Los operadores $\&\&$ y $\|\|$ se llaman **operadores en cortocircuito** porque si no se cumple la condición de un término no se evalúa el resto de la operación. Por ejemplo: $(a == b \&\& c != d \&\& h >= k)$ tiene tres evaluaciones: la primera comprueba si la variable a es igual a b . Si no se cumple esta condición, el resultado de la expresión es falso y no se evalúan las otras dos condiciones posteriores.

En un caso como $(a < b \|\| c != d \|\| h <= k)$ se evalúa si a es menor que b . Si se cumple esta condición el resultado de la expresión es verdadero y no se evalúan las otras dos condiciones posteriores.

El operador $!$ recomendamos no usarlo hasta que se tenga una cierta destreza en programación. Una expresión como $(!esVisible)$ devuelve false si $(esVisible == \text{true})$, o true si $(esVisible == \text{false})$. En general existen expresiones equivalentes que permiten evitar el uso de este operador cuando se desea.

OPERADOR DE NEGACIÓN APLICADO SOBRE NÚMEROS O TEXTO

Si $a = \text{true}$ su negación $!a$ devuelve false. ¿Pero qué ocurre si a es un número o un texto? Si a es un número se considera que equivale a false si su valor numérico es 0, o que equivale a true si su valor numérico es distinto de cero. Seguidamente se aplica la negación. Por tanto si $a = 7$, a se considera equivalente a true y $!a$ es false. Si $a = 0$, a se considera equivalente a false y $!a$ es true.

Para cadenas de texto, la cadena vacía se considera equivale a false y cualquier otra cadena se considera que equivale a true. Si $\text{texto1} = ""$ (cadena vacía) entonces $!\text{texto1}$ vale true.

COMPARACIÓN DE CADENAS DE TEXTO CON OPERADORES RELACIONALES

Dos cadenas de texto se pueden comparar resultando que se comparan letra a letra por el valor del equivalente numérico de cada letra. Cada letra tiene un número asociado: por ejemplo la a es el número 97, la b el 98, etc.

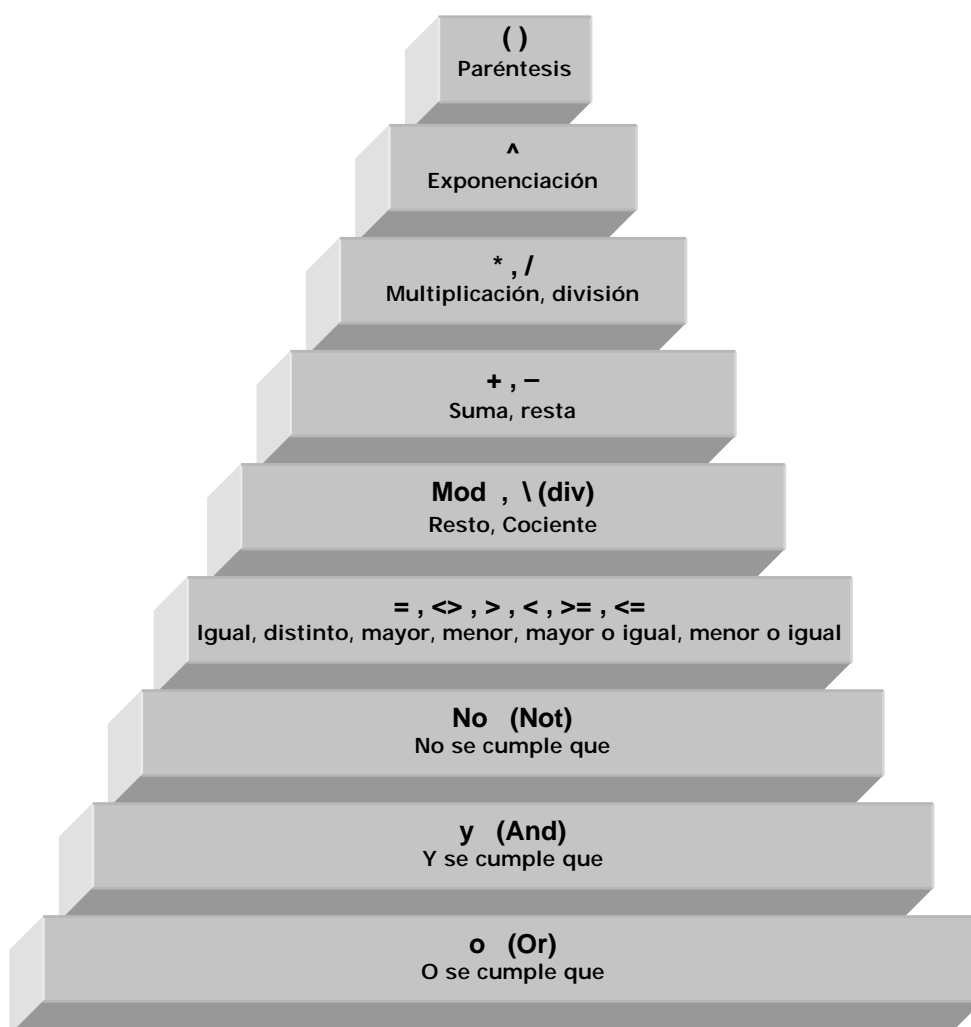
Si comparamos $'avellana' < 'sandia'$ obtenemos true.

Sin embargo, los códigos numéricos pueden generar resultados no previstos. Por ejemplo, ¿qué código numérico es menor, el de la a ó el de la A? Aún más, resulta que todos los códigos numéricos de mayúsculas son menores que los de minúsculas, con lo cual podemos obtener que 'Zulú' < 'avellano' devuelve true (cosa que a priori nos resultará ciertamente extraña).

Para comparar cadenas en base a un orden alfabético necesitaremos usar entonces otras técnicas que comentaremos más adelante.

ORDEN DE PRIORIDAD, PRELACIÓN O PRECEDENCIA

Los operadores lógicos y matemáticos tienen un orden de prioridad o precedencia. Este es un esquema general que indica el orden en que deben evaluarse en la mayoría de los lenguajes de programación:



Una expresión como $A+B == 8 \ \&\& \ A-B == 1$ siendo $A = 3$ y $B = 5$ supondrá que se evalúa primero $A+B$ que vale 8, luego se evalúa $A-B$ que vale -2. Luego se evalúa si se cumple que la primera operación es cierta y luego si la segunda también es cierta, resultando que no, por lo que la expresión es falsa.

EJEMPLO

Veamos un ejemplo. Escribe este código y guárdalo en un archivo de extensión html (puedes cambiar la ruta de la imagen si quieres):

```
<html>
<head>
<title>Curso JavaScript aprenderaprogramar.com</title> <meta charset="utf-8">
<script type="text/javascript">
function mostrarMensaje1() {
var a, b, c, d, eTexto; a = 3; b = 5; c = true; d = false; eTexto = '1';
alert ('Valor de verdad para a+b == 8 && a-b ==1 es: ' + (a+b == 8 && a-b ==1));
alert ('Valor de verdad para a+b == 8 && a-b ==-2 es: ' + (a+b == 8 && a-b ==-2));
alert ('Valor de verdad para c == d es: ' + (c==d));
alert ('Valor de verdad para c&&d es: ' + (c&&d));
alert ('Valor de verdad para c | d es: ' + (c | d));
alert ('Valor de verdad para !a es: ' + (!a));
alert ('Valor de verdad para eTexto === 1: ' + (eTexto === 1));
alert ('Valor de verdad para eTexto == 1: ' + (eTexto == 1));
alert ('Valor de verdad para Zapato < avellano es: ' + ('Zapato'<'avellano'));
}
</script>
</head>
<body>
<div>
<p>Aquí un párrafo de texto situado antes de la imagen, dentro de un div contenedor</p>

<p onclick ="alert('Alerta JavaScript')" style="background-color:yellow;">Aquí otro párrafo de texto.
JavaScript es un lenguaje utilizado para dotar de efectos dinámicos a las páginas web.
</p>
</div>
</body>
</html>
```

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

El resultado esperado es que se muestre lo siguiente (compruébalo):

Valor de verdad para a+b == 8 && a-b ==1 es: false (Aceptar)

Valor de verdad para a+b == 8 && a-b ==-2 es: true (Aceptar)

Valor de verdad para c == d es: false (Aceptar)

Valor de verdad para c&&d es: false (Aceptar)

Valor de verdad para c | d es: true (Aceptar)

Valor de verdad para !a es: false (Aceptar)

Valor de verdad para eTexto === 1: false (Aceptar)

Valor de verdad para eTexto == 1: true (Aceptar)

Valor de verdad para Zapato < avellano es: true (Aceptar)

Importante: recordar que cuando se quieran realizar comparaciones de igualdad hay que usar el operador `==` y no el operador `=`.

EJERCICIO 1

Dadas las variables de tipo entero con valores $A = 5$, $B = 3$, $C = -12$ indicar si la evaluación de estas expresiones daría como resultado verdadero o falso:

- | | |
|-------------------|--|
| a) $A > 3$ | i) $C / B < A$ |
| b) $A > C$ | j) $C / B == -10$ |
| c) $A < C$ | k) $C / B == -4$ |
| d) $B < C$ | l) $A + B + C == 5$ |
| e) $B != C$ | m) $(A+B == 8) \ \&\& \ (A-B == 2)$ |
| f) $A == 3$ | n) $(A+B == 8) \ \ (A-B == 6)$ |
| g) $A * B == 15$ | o) $A > 3 \ \&\& \ B > 3 \ \&\& \ C < 3$ |
| h) $A * B == -30$ | p) $A > 3 \ \&\& \ B >= 3 \ \&\& \ C < -3$ |

Crea un script donde declares estas variables, les asignes valores y muestres por pantalla el valor de verdad que tienen cada una de las expresiones antes indicadas. ¿Coincide lo que se muestra con pantalla con lo que tú esperarías que se mostrara? Puedes comprobar si tus resultados son correctos consultando en los foros aprenderaprogramar.com.

EJERCICIO 2

Crea un script donde declares dos variables a y b y fuerces a que ambas contengan valor NaN. Ahora realiza las comparaciones $a == b$, $a === b$, $a !== a$ ¿Qué resultados obtienes? ¿A qué crees que se debe este resultado?

Próxima entrega: CU01118E

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