



aprenderaprogramar.com

Operadores lógicos y de comparación en programación. Not, and, or... Ejemplos. (CU00132A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 31 del Curso Bases de la programación Nivel I

24

OPERADORES LÓGICOS O DE COMPARACIÓN EN PROGRAMACIÓN. EJEMPLOS.

Otro tipo de operadores fundamentales en programación son los operadores lógicos o de comparación. Se usan para comparar dos variables o expresiones y obtener un valor verdadero o falso. Como operadores de comparación disponemos de:



Símbolo	Significado	Ejemplo
=	Igual que	$A = B$
>	Mayor que	$A > B$
<	Menor que	$A < B$
>=	Mayor o igual que	$A \geq B$
<=	Menor o igual que	$A \leq B$
<>	Distinto que	$A \neq B$

Nótese que las expresiones matemáticas de *mayor o igual que* (\geq) y *menor o igual que* (\leq) se escriben utilizando los símbolos de mayor o menor seguidos de un igual (\geq , \leq).

La evaluación de una comparación da lugar a un resultado de verdadero o falso. Este resultado puede servir de base para tomar una decisión, pero también puede almacenarse en una variable booleana. Por ejemplo:

$Cam = H < 3$ da lugar a que la variable *Cam* adquiera valor verdadero si *H* es menor que tres, o falso si *H* no es menor que tres.

Supongamos distintas expresiones de comparación entre dos variables A y B con valores $A = 2$ y $B = 7$.

Expresión	Evaluación
$A > B$	Falso
$A < B$	Verdadero
$A >= B$	Falso
$A <= B$	Verdadero
$B > A$	Verdadero
$A - B > 0$	Falso
$A + B > 0$	Verdadero
$Res = A > B$	Falso es almacenado en la variable Res
$(A / B) * B <> A$	*
$A <> B$	Verdadero
$A = B$	Falso

Hemos dejado una expresión sin evaluación aunque el ordenador le asignará un valor: verdadero o falso. Lo que no es tan inmediato es decir qué opción será la que aplique el ordenador. Depende de factores como la prioridad de las operaciones, el número de decimales, etc.

Pensemos que un ordenador, a efectos matemáticos, funciona como una calculadora. Y hagamos en nuestra casa la siguiente experiencia:

- Con una calculadora no científica, dividamos 2 entre 7. Una calculadora con 8 dígitos nos devuelve el resultado con 7 decimales como $0,2857142$. A continuación multiplicamos el resultado por 7 y obtenemos $1,9999994$.
- Con una calculadora más avanzada podemos obtener un resultado de $2 / 7$ con más decimales y un valor final igual a 2. Pero con esta misma calculadora podemos dividir 1 entre 3, obtener un valor $0,3$ periódico y luego multiplicar por 3 para obtener $0,999999999$.

$(A / B) * B$ tenemos claro que es igual a A . Por ejemplo $(2 / 7) * 7 = 2$. Pero un ordenador podría devolver un resultado $(2 / 7) * 7 = 1,999$. Con lo cual nos diría que $(A / B) * B$ es distinto de A .

En definitiva, un ordenador no trabaja con infinitos decimales y puede dar lugar a resultados inesperados. Estos casos habrán de ser previstos por el programador, que tomará las medidas que estime convenientes. La forma de operar de los ordenadores o las características del lenguaje que estemos utilizando en un momento dado pueden dar lugar a aparentes fallos o resultados inesperados. No son muy habituales pero conviene tener en cuenta que pueden aparecer. A estos “resultados inesperados” los llamamos *paradojas*. El caso de $(A / B) * B \neq A$ lo denominamos “paradoja de los decimales finitos”. Conocer las paradojas de un lenguaje y manejarlas a su antojo es un hobby para muchos programadores.

Un aviso para navegantes: cuando se produce un “resultado inesperado” se puede confiar con un 99,9% de probabilidad, que se debe a un fallo del programador. Por lo tanto lo que procede es revisar paso a paso el código hasta encontrar el punto que ha dado lugar al error, en vez de quejarse de que el ordenador no responde como debiera. Más bien el ordenador se quejaría de nosotros si pudiera.

En cuanto a variables alfanuméricas, de momento consideraremos válido únicamente el uso del operador “igual”. Si dos variables son iguales exactamente su comparación devuelve “verdadero”. Por ejemplo:

```

A = "Noche"
B = "Noches"
C = "Noche"

A = B Falso
A = C Verdadero
    
```

Por último, disponemos de operadores de negación (no se cumple que), conjugación (y) y disyunción (ó). En inglés *not*, *and* y *or*.

Símbolo	Significado	Ejemplo
No (Not)	No se cumple que	No $A = B$
y (And)	Y se cumple que	$A = B$ y $H > M$
o (Or)	O se cumple que	$A = B$ ó $H > M$

Nota: usaremos una o acentuada con el fin de evitar que se confunda con el número cero.

Respecto a estos operadores, sólo podrán ser usados para evaluar expresiones pero nunca para escribir instrucciones una detrás de otra, relacionadas con estos operadores. Por ejemplo, $A = 3$ y $B = 8$ no se pueden usar como instrucciones para que el ordenador asigne valores. Únicamente se pueden usar con el fin de evaluar el hecho de que se cumpla $A = 3$ y $B = 8$ dando lugar a un resultado de verdadero, o que no se cumpla, dando lugar a un resultado de falso.

Respecto al operador *No (Not)*, además de saber de su existencia y significado, prescindiremos de su uso ya que en general existirá una expresión equivalente en la que no es necesario y que proporciona mayor claridad y facilidad de lectura.

Expresiones equivalentes para el operador *No*:

Expresiones con Not	Equivalente
Not $A <> 0$	$A = 0$
Not $A > B$	$A <= B$
Not $A >= B$	$A < B$
Not $A = B$	$A <> B$

Próxima entrega: CU00133A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59