



APRENDERAPROGRAMAR.COM

OPERADORES LÓGICOS Y
DE COMPARACIÓN EN PHP.
PRIORIDADES. EJEMPLOS.
EJERCICIOS RESUELTOS.
(CU00818B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº18 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez

OPERADORES LÓGICOS Y DE COMPARACIÓN EN PHP

Los operadores de comparación permiten comparar dos valores, tal y como su nombre indica. En general, esto nos servirá para tomar decisiones. Por ejemplo, quizás tenemos que decidir si autorizar una operación según exista saldo suficiente en su cuenta bancaria o no. Para ello en PHP disponemos de los operadores que se indican en la siguiente tabla.



Nombre	Ejemplo	Resultado
Mayor que	$\$a > \b	<code>true</code> si $\$a$ es mayor que $\$b$ <code>false</code> en caso contrario
Menor que	$\$a < \b	<code>true</code> si $\$a$ es menor que $\$b$ <code>false</code> en caso contrario
Mayor o igual que	$\$a >= \b	<code>true</code> si $\$a$ es mayor o igual que $\$b$ <code>false</code> en caso contrario
Menor o igual que	$\$a <= \b	<code>true</code> si $\$a$ es menor o igual que $\$b$ <code>false</code> en caso contrario
Diferente	$\$a <> \b ó $\$a != \b	<code>true</code> si $\$a$ es diferente a $\$b$ <code>false</code> en caso contrario
Idéntico o estrictamente igual	$\$a === \b	<code>true</code> si $\$a$ es igual a $\$b$ y son del mismo tipo <code>false</code> en caso contrario
No idéntico ó estrictamente distinto	$\$a !== \b	<code>true</code> si $\$a$ no es igual a $\$b$ o no son del mismo tipo <code>false</code> en caso contrario
Igual	$\$a == \b	<code>true</code> si $\$a$ es igual a $\$b$ <code>false</code> en caso contrario

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.

Es decir, si se compara un número con una cadena de caracteres o la comparación implica cadenas de caracteres numéricas, cada cadena de caracteres es convertida en un número y la comparación se realizará numéricamente excepto cuando el operador utilizado es === o !== porque, en estos casos, también comparamos el tipo.

OPERADORES LÓGICOS EN PHP

Los operadores lógicos, nos permiten crear condiciones para las distintas estructuras en PHP, tanto en estructuras condicionales como en estructuras repetitivas. Los operadores lógicos más importantes son and (en español: y) y or (en español: o)

Nombre	Ejemplo	Resultado
and ó &&	\$a and \$b	true si \$a es true y \$b es true false en caso contrario
or ó	\$a or \$b	true si \$a o \$b es true, o ambos false en caso contrario
xor	\$a xor \$b	true si \$a o \$b es true, pero no ambos false en caso contrario
not ó !	!\$a	true si \$a no es true false en caso contrario

Una condición puede ser tan larga como se quiera o necesite y usar tantos operadores lógicos como se quiera o necesiten. El uso de los paréntesis otorgará la prioridad de la ejecución de unas operaciones y otras (como en matemáticas).

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 = true y \$b = false la expresión \$a && \$b devuelve false (es falsa porque no se cumple que \$a y \$b sean verdaderas). Si \$a = true y \$b = false la expresión \$a || \$b devuelve true porque uno de los dos operandos es verdadero. Si \$a = 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 == true), o true si (\$esVisible == false). En general existen expresiones equivalentes que permiten evitar el uso de este operador cuando se desea.

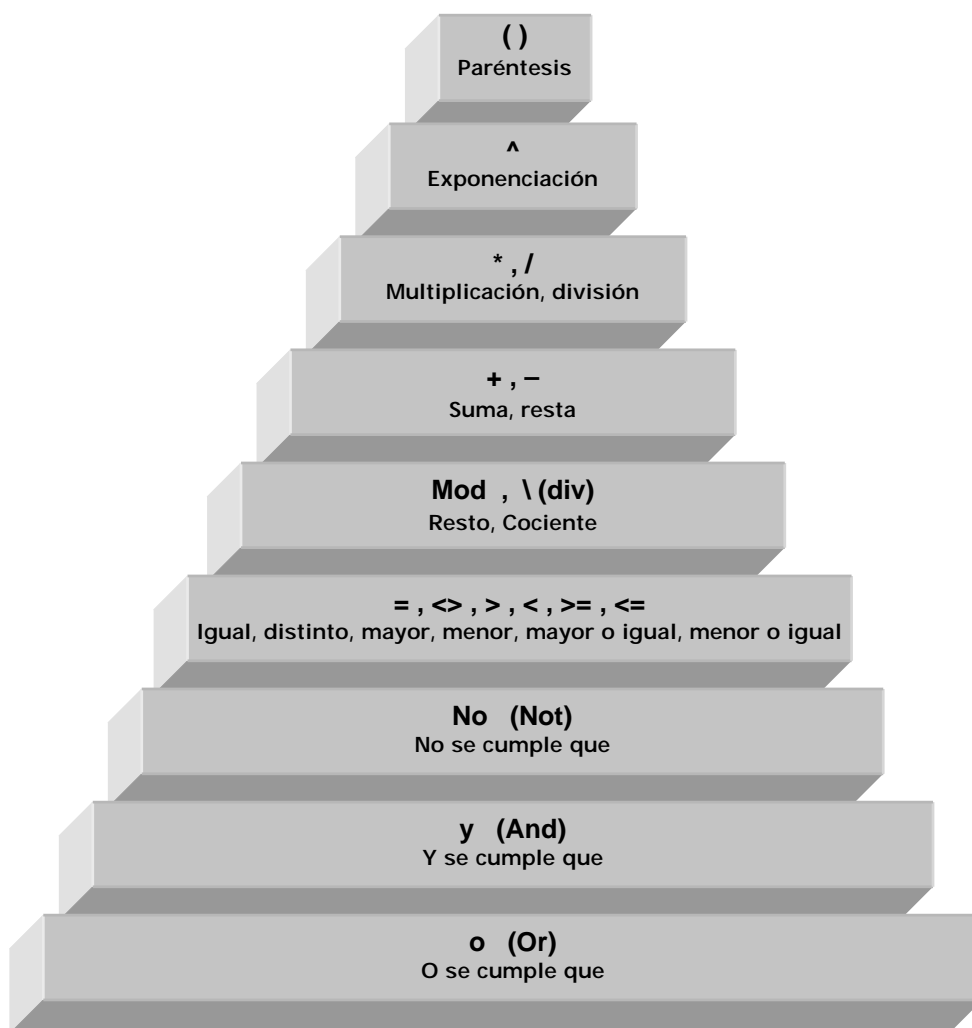
OPERADOR DE NEGACIÓN APLICADO SOBRE NÚMEROS O TEXTO

Si \$a = 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 \$texto1 = "" (cadena vacía) entonces !\$texto1 vale true.

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.

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.

EJEMPLO

Escribe este código en un editor como Notepad++ y guárdalo con un nombre de archivo como `ejemplo1.php`. A continuación, sube el fichero al servidor y visualiza el resultado.

```
<?php
$a = 3;
$b = 9;
$result = ($a <= 3 and $b <> 9);
if ( $result == true) {
    echo "Se cumple la condición";
}else {
    echo "No se cumple la condición";
}
?>
```



EJERCICIO

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 el código PHP 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 tu código y tus respuestas son correctos consultando en los foros aprenderaprogramar.com.

Próxima entrega: CU00819B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193