

Antes de empezar

Si es este tu primer contacto con el mundo de la programación es probable que estés pensando que *todo esto está muy bien* pero que a ti lo que te interesa es *hacer cosas*. En una palabra, que quieres usar PHP como herramienta para el desarrollo de tus proyectos.

Esta página y las siguientes van a ser básicas para el éxito en ese lógico y razonable afán utilitarista.

Pensemos en nuestro día a día. En cada momento de nuestra vida *hacemos algo*. Cada cosa que hacemos suele requerir casi siempre esta *secuencia* de acciones: *comparar, evaluar, optar y hacer*.

Supongamos que queremos *cambiar de coche*. Lo más prudente será empezar por *comparar* las *características* (potencia, diseño, precio, etc.) de los diferentes modelos.

Salvo en casos excepcionales, no tomaríamos una decisión de compra a través de un solo parámetro sino que haríamos una *evaluación conjunta* de todos esos factores (*menos potencia pero mejor precio y diseño muy similar*, por citar un ejemplo) y sería a través de esa evaluación como *optaríamos* por una marca o modelo.

Una vez ejercida la opción –y no antes– sería el momento de *realizar* la compra del nuevo coche.

PHP, y en general todos los lenguajes de programación, disponen de herramientas que permiten emular cada uno de esos procesos de la conducta humana. Los iremos viendo en esta página y en las siguientes.

Operadores de comparación

PHP dispone de los siguientes operadores de comparación:

\$A == \$B

El operador **==** **compara** los valores de dos variables y devuelve **1** (CIERTO) en el caso de que sean iguales y el valor **NUL** –carácter ASCII 0– (FALSO) cuando son distintas.

Mediante este operador se pueden comparar variables de distinto tipo.

Manejando operadores de comparación

Desarrollamos aquí algunos ejemplos de manejo de los operadores de comparación.

El operador ==				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	==	\$A==\$B	1
\$A=123.0	\$B=123	==	\$A==\$B	1
\$A=123	\$B="123"	==	\$A==\$B	1
\$A=123	\$B="123ABC"	==	\$A==\$B	1
\$A=123	\$B=124	==	\$A==\$B	
\$A=123	\$B=124	==	ord(\$A==\$B)	0
\$A="abc"	\$B="ABC"	==	\$A==\$B	
\$A="abc"	\$B="abc"	==	\$A==\$B	1
\$A="abc"	\$B="abcd"	==	\$A==\$B	

Los valores de la columna señalada como *Resultados* se obtienen mediante la función **echo(\$A==\$B)**; Podemos ver que, en los casos que es cierta la igualdad, imprime un 1, mientras que cuando es falsa *no imprime nada*. Se justifica este extremo porque en caso de no coincidencia el valor devuelto es NUL (código ASCII 0) y ese carácter carece de símbolo gráfico. Hemos marcado en rojo una excepción. En ese caso, la instrucción es **echo ord(\$A==\$B)**; y –como recordarás– dado que **ord**, que devuelve el código ASCII del carácter contenido en el paréntesis que le acompaña, podemos comprobar a través del **cero** de este resultado, que, efectivamente, la no coincidencia devuelve NUL.

El operador ===				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	===	\$A===B	1
\$A=123.0	\$B=123	===	\$A===B	
\$A=123	\$B="123"	===	\$A===B	
\$A=123	\$B="123ABC"	===	\$A===B	
\$A=123	\$B=124	===	\$A===B	
\$A=123	\$B=124	===	ord(\$A===B)	0
\$A="abc"	\$B="ABC"	===	\$A===B	
\$A="abc"	\$B="abc"	===	\$A===B	1
\$A="abc"	\$B="abcd"	===	\$A===B	

Observa que los valores señalados en rojo –a diferencia de lo que ocurre con el operador anterior– devuelven **NUL** como resultado. En este caso no sólo compara valores sino que también compara tipos de variables. Al ser una de ellas tipo numérico y la otra cadena alfanumérica el resultado no puede ser otro que NUL.

El operador !=				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	!=	\$A!=B	
\$A=123.0	\$B=123	!=	\$A!=B	
\$A=123	\$B="123"	!=	\$A!=B	
\$A=123	\$B="123ABC"	!=	\$A!=B	
\$A=123	\$B=124	!=	\$A!=B	1
\$A=123	\$B=124	!=	ord(\$A!=B)	49
\$A="abc"	\$B="ABC"	!=	\$A!=B	1
\$A="abc"	\$B="abc"	!=	\$A!=B	
\$A="abc"	\$B="abcd"	!=	\$A!=B	1

Para comparar una *cadena* con un *número* se extrae el *valor entero de la cadena* (si lleva dígitos al comienzo los extrae y en caso contrario le asigna el valor cero) y utiliza ese valor para hacer la comparación.

Cuando se comparan cadenas *discrimina* entre *mayúsculas* y *minúsculas* ya que utiliza los códigos ASCII de cada uno de los caracteres para hacer la comparación.

La comparación se hace de izquierda a derecha y devuelve **1** (CIERTO) sólo en el caso que coincidan exactamente los contenidos de ambas cadenas.

\$A === \$B

El operador **===** es similar al anterior, pero realiza la comparación en sentido **estricto**.

Para que devuelva **1** es necesario que sean iguales *los valores de las variables y también su tipo*.

\$A != \$B

El operador **!=** devuelve **1** cuando los valores de las variables **son distintos** (en general **!** indica negación, en este caso podríamos leer «no igual») y devuelve **NUL** cuando son iguales.

Este operador **no compara** en sentido **estricto**, por lo que puede considerar iguales los valores de dos variables de distinto tipo.

\$A < \$B

El operador **<** devuelve **1** cuando los valores de **\$A** son **menores** que los de **\$B**.

Los criterios de comparación son los siguientes:

– Los *valores numéricos* siguen el criterio matemático.

– Cuando se trata de un *número* y una *cadena* extrae el valor numérico de ésta (es cero si no hay ningún dígito al principio de la misma) y hace una comparación matemática.

– En el supuesto de dos cadenas, compara **uno a uno** –de izquierda a derecha– los códigos ASCII de cada uno de los caracteres (primero con primero, segundo con segundo, etcétera).

Si al hacer esta comprobación encuentra –en la primera cadena– un carácter cuyo código ASCII es mayor que el correspondiente de la segunda cadena, o encuentra que todos son iguales en ambas cadenas devuelve **NUL**. Solo en el caso de no existir ninguno mayor y sí haber al menos uno menor devolverá **UNO**.

En los ejemplos señalados en **rojo** Puedes comprobar el carácter **no estricto** de este operador. Devuelve **NUL** porque considera que *no son distintos*, lo que equivale a interpretar que los considera *iguales* pese a que las variables sean de distinto tipo.

El operador <				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	<	\$A<\$B	
\$A=123	\$B="123"	<	\$A<\$B	
\$A=123.0	\$B="123"	<	\$A<\$B	
\$A=123	\$B="123ABC"	<	\$A<\$B	
\$A=123	\$B=124	<	\$A<\$B	1
\$A=123	\$B=124	<	ord(\$A<\$B)	49
\$A="abc"	\$B="ABC"	<	\$A<\$B	
\$A="abc"	\$B="abc"	<	\$A<\$B	
\$A="bcd"	\$B="abcd"	<	\$A<\$B	1
\$A="aacd"	\$B="abcd"	<	\$A<\$B	1
\$A="abc"	\$B="abcd"	<	\$A<\$B	1
\$A="abcd"	\$B="abc"	<	\$A<\$B	
\$A="A"	\$B="a"	<	\$A<\$B	1
\$A="a"	\$B="A"	<	\$A<\$B	
\$A="aBC"	\$B="A"	<	\$A<\$B	
\$A="123"	\$B=124	<	\$A<\$B	1
\$A=123	\$B="124"	<	\$A<\$B	1

Observa los ejemplos señalados en **rojo**. Cuando las cadenas tienen distinta longitud este operador considera que los caracteres que faltan en la cadena más corta son **NUL**. Esa es la razón por la que en el primer caso devuelve **CIERTO** (**NUL** es menor que **d**) y el segundo **FALSO** (**d** no es menor que **NUL**).

El operador <=				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	<=	\$A<=\$B	1
\$A=123.0	\$B=123	<=	\$A<=\$B	1
\$A=123	\$B="123"	<=	\$A<=\$B	1
\$A=123	\$B="123ABC"	<=	\$A<=\$B	1
\$A=123	\$B=124	<=	\$A<=\$B	1
\$A=123	\$B=124	<=	ord(\$A<=\$B)	49
\$A="abc"	\$B="ABC"	<=	\$A<=\$B	
\$A="abc"	\$B="abc"	<=	\$A<=\$B	1
\$A="abc"	\$B="abcd"	<=	\$A<=\$B	1
\$A="A"	\$B="a"	<=	\$A<=\$B	1
\$A="a"	\$B="A"	<=	\$A<=\$B	
\$A="aBC"	\$B="A"	<=	\$A<=\$B	
\$A="123"	\$B=124	<=	\$A<=\$B	1
\$A=123	\$B="124"	<=	\$A<=\$B	1

Hemos modificado la instrucciones marcadas en **rojo** para comprobar el código ASCII de carácter que devuelve, en el caso de ser cierto, el resultado de la comparación. El valor que aparece (49) como resultado es el código ASCII del carácter **1**.

El operador >				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	>	\$A>\$B	
\$A=123	\$B="123"	>	\$A>\$B	
\$A=123	\$B="123ABC"	>	\$A>\$B	
\$A=123	\$B=124	>	\$A>\$B	
\$A=123	\$B=124	>	ord(\$A>\$B)	0
\$A="abc"	\$B="ABC"	>	\$A>\$B	1
\$A="abc"	\$B="abc"	>	\$A>\$B	
\$A="abc"	\$B="abcd"	>	\$A>\$B	
\$A="A"	\$B="a"	>	\$A>\$B	
\$A="a"	\$B="A"	>	\$A>\$B	1
\$A="aBC"	\$B="A"	>	\$A>\$B	1

– Cuando las cadenas tengan distinta longitud, considerará (a efectos de la comparación) que los caracteres *que faltan* en la cadena más corta son NUL (ASCII 0).

\$A <= \$B

Se comporta de forma idéntica al anterior. La única diferencia es que ahora aceptará como *ciertos* los casos de *igualdad* tanto en el caso de números como en el de códigos ASCII.

\$A > \$B

Es idéntico –en el modo de funcionamiento– a **\$A < \$B**. Solo difiere de éste en el *criterio de comparación* que ahora requerirá que los valores de **\$A** sean *mayores* que los de la variable **\$B**.

\$A >= \$B

Añade al anterior la posibilidad de certeza en caso de igualdad.

\$A="123"	\$B=124	>	\$A>\$B	
\$A=123	\$B="124"	>	\$A>\$B	

El operador >=				
A	B	Operador	Sintaxis	Resultado
\$A=123	\$B=123	>=	\$A>= \$B	1
\$A=123	\$B="123"	>=	\$A>= \$B	1
\$A=123	\$B="123ABC"	>=	\$A>= \$B	1
\$A=123	\$B=124	>=	\$A>= \$B	
\$A=123	\$B=124	>=	ord(\$A>= \$B)	0
\$A="abc"	\$B="ABC"	>=	\$A>= \$B	1
\$A="abc"	\$B="abc"	>=	\$A>= \$B	1
\$A="abc"	\$B="abcd"	>=	\$A>= \$B	
\$A="A"	\$B="a"	>=	\$A>= \$B	
\$A="a"	\$B="A"	>=	\$A>= \$B	1
\$A="aBC"	\$B="A"	>=	\$A>= \$B	1
\$A="123"	\$B=124	>=	\$A>= \$B	
\$A=123	\$B="124"	>=	\$A>= \$B	

Ejercicio nº 17

Escribe un script en el que uses cada uno de los operadores de comparación que hemos descrito en esta página.

Eligiendo las variables adecuadas, debes escribir dos ejemplos –para cada uno de los operadores– de modo que uno de ellos devuelva CIERTO y el otro FALSO.

Guarda el documento como **ejercicio17.php** y comprueba que su funcionamiento es correcto y que los resultados de las comparaciones coinciden con lo que habías previsto.

Anterior

Índice

Siguiente