

Comentario previo

Incluimos la sintaxis de este tipo de operadores a título meramente informativo. Rara vez será necesario utilizarlos en nuestras aplicaciones PHP.

Su utilidad suele limitarse a la gestión de periféricos y algunas operaciones de cálculo de carácter muy reiterativo en la que se puede conseguir un rendimiento muy superior a los operadores tradicionales.

En el ámbito propio del PHP pueden tener algún interés a la hora de elaborar rutinas para *encriptar* el código fuente de algunos scripts que por su importancia pueden requerir ese tipo de protección.

Los que sí han de resultarnos de gran interés serán el resto de los operadores. Los iremos viendo en páginas sucesivas.

Operadores bit a bit

\$A & \$B

El operador **&** compara los valores binarios de cada uno de los bits de las cadenas \$A y \$B y devuelve **1** en el caso que ambos sean **1**, y **0** en cualquier otro caso.

Cuando las variables **\$A** y **\$B** son **cadenas** compara los valores binarios de los códigos ASCII de sus caracteres y devuelve los caracteres ASCII correspondientes al resultado de esa comparación.

\$A | \$B

Funciona de forma idéntica al anterior y devuelve **1** cuando **al menos** el valor de uno de los bits comparados es **1**, y devolverá **0** cuando **ambos** sean **0**.

\$A ^ \$B

Devuelve **1** cuando los bits comparados son **distintos**, y **0** cuando son **iguales**.

\$A << \$B

Realiza la operación **\$A * 2^{\$B}**.

Hace el cálculo añadiendo **\$B CEROS** (binarios) a la derecha de la cadena binaria \$A.

\$A >> \$B

Divide el valor **\$A** entre **2^{\$B}**. Hace la operación en la cadena **binaria** quitando **\$B CEROS** (por la derecha) de la cadena \$A.

~ \$A

Invierte los valores de los bits de la cadena **\$A** convirtiendo los CEROS en UNO y los UNO en CERO.

Manejando operadores bit a bit

Desarrollamos aquí algunos ejemplos de manejo de los operadores bit a bit.

El operador &					
Números		Números como cadenas		Cadenas alfanuméricas	
Variables	Valores binarios	Variables	Valores binarios	Variables	Valores binarios
\$a=12	1100	\$A="12"	110001110010	\$A1="Rs"	10100101110011
\$b=23	10111	\$B="23"	110010110011	\$B1="aZ"	11000011011010
\$a&\$b=4	100	\$A&\$B=02	110000110010	\$A1&\$B1=@R	10000001010010

En los casos de cadenas hemos diferenciado en rojo el valor binario correspondiente al primer carácter. Esos valores binarios corresponden a la forma binaria del código ASCII de cada uno de los caracteres

Puedes observar que el tratamiento es distinto cuando los mismos valores numéricos se asignan como entero y como cadena.

Al asignarlos como cadena *opera* los valores binarios de los códigos ASCII de los caracteres, mientras que cuando se trata de números compara los valores de las expresiones binarias de los valores de cada uno de ellos

El operador					
Números		Números como cadenas		Cadenas alfanuméricas	
Variables	Valores binarios	Variables	Valores binarios	Variables	Valores binarios
\$a=12	1100	\$A="12"	110001110010	\$A1="Rs"	10100101110011
\$b=23	10111	\$B="23"	110010110011	\$B1="aZ"	11000011011010
\$a b=31	11111	\$A B=33	110011110011	\$A1 B1=s{	11100111111011

Se comporta de forma idéntica al anterior en lo relativo a números y cadenas.

El operador ^					
Números		Números como cadenas		Cadenas alfanuméricas	
Variables	Valores binarios	Variables	Valores binarios	Variables	Valores binarios
\$a=12	1100	\$A="12"	110001110010	\$A1="Rs"	10100101110011
\$b=23	10111	\$B="23"	110010110011	\$B1="aZ"	11000011011010
\$a^b=27	11011	\$A^B=	000011000001	\$A1^B1=3)	01100110101001

Los criterios de tratamiento de números y cadenas coinciden con los operadores anteriores.

El operador <<					
Números		Números como cadenas		Cadenas alfanuméricas	
Variables	Valores binarios	Variables	Valores binarios	Variables	Valores binarios
\$a=12	1100	\$A="12"	110001110010	\$A1="Rs"	10100101110011
\$b=2	10	\$B=2	10	\$B1=2	10
\$a<<b=48	110000	\$A<<B=48	110000	\$A1<<B1=0	

El operador **<<** **multiplica** el valor de la primera cadena por 2 *elevado al valor de la segunda*.

Al ser un operador *matemático* solo tiene sentido cuando ambas variables son números naturales. En las **cadenas alfanuméricas** **extrae los números que pudiera haber al comienzo** y, en caso de no haberlos, toma valor cero.

El operador >>					
Números		Números como cadenas		Cadenas alfanuméricas	
Variables	Valores binarios	Variables	Valores binarios	Variables	Valores binarios
\$a=12	1100	\$A="12"	110001110010	\$A1="Rs"	10100101110011
\$b=2	10	\$B=2	10	\$B1=2	10
\$a>>b=3	11	\$A>>B=3	11	\$A1>>B1=0	

Para este operador (**>>**) son aplicables los mismos comentarios hechos en el párrafo anterior.

El operador ~	
\$a=12	1100
~\$a=-13	11111111111111111111111111111111