

■PHP演算子の詳細と優先順位

作成: 2015.7.11 yoshi

PHP演算子は式の値の組み合わせ、比較、または修正の方法を指定する文字です。

代数演算子		優先順位	説明	サンプル
+	addition	8	数値式を加算します。	a+b → aにbを加える
-	subtraction	8	減算に使用します。	a-b → aからbを引く
-	minus	8	マイナス(負にする)	(y = -x → y = -12)
*	multiplication	7	2つの数値や式を乗算します。	a*b → aとbを掛ける
/	division	7	2つの数値や式を除算します。	a/b → aをbで割る
%	modulus	7	2つの数値や式を除算ときの剰余を計算します。	a%b → aをbで割った余り
**	power	3	累乗(PHP 5.6で導入)	a**b → aのb乗
++	increment	4	加算子:式に1を加算します。	(y = ++x → y=13,x=13:代入前に加算) (y = x++ → y=12,x=13:代入後に加算)
--	decrement	4	減算子:オペランドから1を減算します。	(y = --x → y=11,x=11:代入前に減算) (y = x-- → y=12,x=11:代入後に減算)
代入演算子				
=	assignment	18	右側のオペランドの値を変数、配列エレメント、またはプロパティに代入(割り当て)する。	a=5 → aに5を代入する yy = xx = 5; → yyとxxに5が代入される
+=	addition assignment	18	指定された値を加算する	a+=5 → aに5を加える(a=a+5と同じ)
-=	subtraction assignment	18	指定された値を減算する	a-=5 → aから5を引く(a=a-5と同じ)
=	addition assignment	18	指定された値を乗算する	a=5 → aに5を掛ける(a=a*5と同じ)
**=	power assignment	18	指定された値を累乗する	
/=	division assignment	18	指定された値を除算する	a/=5 → aを5で割る(a=a/5と同じ)
%=	modulos assignment	18	指定された値で除算し、余りを求める	a%=5 → aを5で割った余り(a=a%5と同じ)
&=	Bitwise And	18	expression1にexpression1 & expression2の値を割り当てます。	[expression1] &= [expression2];
=	Bitwise Or	18	expression1にexpression1 expression2の値を割り当てます。	[expression1] = [expression2];
^=	Bitwise Xor	18	expression1にexpression1 ^ expression2の値を割り当てます。	[expression1] ^= [expression2];
<<=	Left shift	18	ビット単位の左シフト(<<=)演算を行い、その内容を結果としてexpression1に格納します。	
>>=	Right shift	18	ビット単位の右シフト演算を行い、結果をexpressionに格納します。	
ビット演算子				
&	bitwise AND	12	expression1とexpression2を整数パラメータをビット単位で論理積(AND)演算します。	xx = 0x87654321 & 0xffff0000; // xxには0x87650000が代入される
<<	bitwise left shift	9	expression1とshiftCountをshiftCountの変換により生成された整数で指定される桁数だけexpression1内のすべてのビットを左にシフトします。	xx = 0x12345678 << 8; // xxには0x34567800が代入される
	bitwise OR	14	expression1とexpression2をexpression1とexpression2の対応ビットの少なくとも一方が1である各ビット位置に1を設定します。	xx = 0x87654321 0xffff0000; // xxには0xffff4321が代入される
>>	bitwise right shift	9	expressionとshiftCountをshiftCountの変換により生成された整数で指定される桁数だけexpression内のすべてのビットを右にシフトします。	xx = 0x12345678 >> 8; // xxには0x00123456が代入される
~	bitwise NOT	4	否定	
^	bitwise XOR	13	expression1とexpression2をexpression1とexpression2の対応ビットのいずれか一方のみが1である各ビット位置に1を設定します。	xx = 0xffff0000 ^ 0xff00ff00; // xxには0x00ffff00が代入される
比較演算子				
==	equality	11	2つの式の等価性をテストします。	a==5 → aが5のときtrue、それ以外false
>	greater than	10	2つの式を比較し、expression1がexpression2より大きいかどうかを判定します。大きい場合、結果はTRUEになります。	a>5 → aが5より大きいときtrue、それ以外false
>=	greater than or equal to	10	2つの式を比較し、expression1がexpression2より大きいか等しい(TRUE)、またはexpression1がexpression2より小さい(FALSE)かどうかを判定します。	a>=5 → aが5以上のときtrue、それ以外false
!=	inequality	11	等価(==)演算子の正反対がTRUEであるかどうかをテストします。	a!=5 → aが5以外のときtrue、5のときfalse
<	less than	10	2つの式を比較し、expression1がexpression2より小さいかどうかを判定します。小さい場合、結果はTRUEになります。	a<5 → aが5未満のときtrue、それ以外false
<=	less than or equal to	10	2つの式を比較し、expression1がexpression2より小さいか等しいかどうかを判定します。小さいか等しい場合、結果はTRUEになります。	a<=5 → aが5以下のときtrue、それ以外false
===	strict equality	11	自動データ変換を行わずに、2つの式が等しいかどうかをテストします。	a===5 → aが5と等価のときtrue、それ以外false
!==	strict inequality	11	厳密な等価(===)演算子の正反対がTRUEであるかどうかをテストします。	a!==5 → aが5と不等価のときtrue、それ以外false
<>	not equality	11		
論理演算子				
!	logical NOT 論理否定	6	expression1がTRUEではない時にTRUEを返します。	![expression1]
&& and	logical AND 論理積	15 19	expression1がTRUE、かつ、expression2がTRUEであればTRUEを、さもなければFALSEを返します。	[expression1] && [expression2]
 or	logical OR 論理和	16 21	expression1がTRUE、または、expression2がTRUEであればTRUEを返します。	[expression1] [expression2]
xor	Exclusive OR 排他的論理和	20	expression1またはexpression2のどちらかがTRUEでかつ両方ともTRUEでない場合にTRUE	[expression1] xor [expression2]
三項演算子				
?:	conditional 条件	17	expression1を評価し、expression1の値がTRUEである場合、結果はexpression2の値となり、そうでない場合はexpression3の値となります。	[expression1]? [expression2]: [expression3]
文字列演算子				
.	concatenation (dot)	8	ストリングを連結(結合)します。	xx = "123".abc; → xxには"123abc"が代入される
.=	concatenation assignment	18	expression1にexpression1 + expression2の値を割り当てます。	
特殊演算子				
new	new	1	クラスインスタンスをインスタンス化します。	
clone	clone	1		
instanceof	instanceof	5	式のプロトタイプチェーンにfunctionのプロトタイプオブジェクトが含まれるかどうかを評価します。	
その他				
[]	array access	2	指定されたエレメント(a0など)で新しい配列または多次元配列を初期化するか、配列内のエレメントにアクセスします。	
@		4	エラー制御演算子:PHPの式の前に付けた場合、その式により生成されたエラーメッセージは無視されます。	
&	reference	12	リファレンス	
..		22	さまざまな利用法	
`	back quote		実行演算子:バッククォートの中身をシェルコマンドとして実行しようとする。出力が返されます	\$output = `ls -al`;
(int)		4	型演算子:整数	
(float)		4	型演算子:浮動小数点	
(string)		4	型演算子:文字列	
(array)		4	型演算子:配列	
(object)		4	型演算子:オブジェクト	
(bool)		4	型演算子:論理値	