



10. PHP 运算符

在 PHP 中，赋值运算符 = 用于给变量赋值。

在 PHP 中，算术运算符 + 用于把值加在一起。

10.1 PHP 算术运算符

PHP 算术运算符

运算符	名称	描述
<u>x + y.</u>	加	x 和 y 的和
<u>x - y.</u>	减	x 和 y 的差
<u>x * y.</u>	乘	x 和 y 的积
<u>x / y.</u>	除	x 和 y 的商
<u>x % y.</u>	模（除法的余数）	x 除以 y 的余数
<u>-x</u>	取反	x 取反
<u>a . b</u>	并置	连接两个字符串

以下实例演示了使用不同算术运算符得到的不同结果：

```
<?php
$x=10;
$y=6;
echo ($x + $y); // 输出16
```

```

echo '<br>'; // 换行

echo ($x - $y); // 输出4
echo '<br>'; // 换行

echo ($x * $y); // 输出60
echo '<br>'; // 换行

echo ($x / $y); // 输出1.6666666666667
echo '<br>'; // 换行

echo ($x % $y); // 输出4
echo '<br>'; // 换行

echo -$x;
?>

```

PHP7+ 版本新增整除运算符 intdiv(),使用实例:

```

<?php
var_dump(intdiv(10, 3));
?>

```

输出结果:

```
int(3)
```

10.2 PHP 赋值运算符

在 PHP 中,基本的赋值运算符是 "=". 它意味着左操作数被设置为右侧表达式的值。也就是说, "\$x = 5" 的值是 5。

PHP 赋值运算符

运算符	等同于	描述
<u>$x = y$</u>	$x = y$	左操作数被设置为右侧表达式的值
<u>$x += y$</u>	$x = x + y$	加
<u>$x -= y$</u>	$x = x - y$	减
<u>$x *= y$</u>	$x = x * y$	乘
<u>$x /= y$</u>	$x = x / y$	除
<u>$x \% = y$</u>	$x = x \% y$	模 (除法的余数)
<u>$a . = b$</u>	$a = a . b$	连接两个字符串

以下实例演示了使用不同赋值运算符得到的不同结果：

```
<?php
$x=10;
echo $x; // 输出10

$y=20;
$y += 100;
echo $y; // 输出120

$z=50;
$z -= 25;
echo $z; // 输出25

$i=5;
$i *= 6;
echo $i; // 输出30

$j=10;
$j /= 5;
echo $j; // 输出2

$k=15;
$k %= 4;
echo $k; // 输出3
?>
```

以下实例演示了使用不同字符串运算符得到的相同结果：

```
<?php
$a = "Hello";
$b = $a . " world!";
echo $b; // 输出Hello world!

$x="Hello";
$x .= " world!";
echo $x; // 输出Hello world!
?>
```

10.3 PHP 递增/递减运算符

PHP 递增/递减运算符

运算符	名称	描述
<u>++</u> X	预递增	x 加 1，然后返回 x
X <u>++</u>	后递增	返回 x，然后 x 加 1
<u>--</u> X	预递减	x 减 1，然后返回 x

运算符	名称	描述
<u><code>x--</code></u>	后递减	返回 x，然后 x 减 1

以下实例演示了使用递增/递减运算符得到的结果：

```
<?php
$x=10;
echo ++$x; // 输出11

$y=10;
echo $y++; // 输出10

$z=5;
echo --$z; // 输出4

$i=5;
echo $i--; // 输出5
?>
```

10.3 PHP 比较运算符

PHP 比较运算符

运算符	等同于	描述
<u><code>x == y</code></u>	等于	如果 x 等于 y，则返回 true
<u><code>x === y</code></u>	绝对等于	如果 x 等于 y，且它们类型相同，则返回 true
<u><code>x != y</code></u>	不等于	如果 x 不等于 y，则返回 true
<u><code>x <> y</code></u>	不等于	如果 x 不等于 y，则返回 true
<u><code>x !== y</code></u>	绝对不等于	如果 x 不等于 y，或它们类型不相同，则返回 true
<u><code>x > y</code></u>	大于	如果 x 大于 y，则返回 true
<u><code>x < y</code></u>	小于	如果 x 小于 y，则返回 true
<u><code>x >= y</code></u>	大于等于	如果 x 大于或者等于 y，则返回 true
<u><code>x <= y</code></u>	小于等于	如果 x 小于或者等于 y，则返回 true

以下实例演示了使用一些比较运算符得到的不同结果：

```
<?php
$x=100;
$y="100";

var_dump($x == $y);
echo "<br>";
var_dump($x === $y);
```

```

echo "<br>";
var_dump($x != $y);
echo "<br>";
var_dump($x !== $y);
echo "<br>";

$a=50;
$b=90;

var_dump($a > $b);
echo "<br>";
var_dump($a < $b);
?>

```

10.4 PHP 逻辑运算符

PHP 逻辑运算符

运算符	名称	描述
<u>x and y</u>	与	如果 x 和 y 都为 true，则返回 true
<u>x or y</u>	或	如果 x 和 y 至少有一个为 true，则返回 true
<u>x xor y</u>	异或	如果 x 和 y 有且仅有一个为 true，则返回 true
<u>x && y</u>	与	如果 x 和 y 都为 true，则返回 true
<u>x y</u>	或	如果 x 和 y 至少有一个为 true，则返回 true
<u>!x</u>	非	如果 x 不为 true，则返回 true

10.5 PHP 数组运算符

PHP 数组运算符

运算符	名称	描述
<u>x + y</u>	集合	x 和 y 的集合
<u>x == y</u>	相等	如果 x 和 y 具有相同的键/值对，则返回 true
<u>x === y</u>	恒等	如果 x 和 y 具有相同的键/值对，且顺序相同类型相同，则返回 true
<u>x != y</u>	不相等	如果 x 不等于 y，则返回 true
<u>x <> y</u>	不相等	如果 x 不等于 y，则返回 true
<u>x !== y</u>	不恒等	如果 x 不等于 y，则返回 true

以下实例演示了使用一些数组运算符得到的不同结果：

```
<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");
$z = $x + $y; // $x 和 $y 数组合并
var_dump($z);
var_dump($x == $y);
var_dump($x === $y);
var_dump($x != $y);
var_dump($x <> $y);
var_dump($x !== $y);
?>
```

10.6 三元运算符

10.6.1 语法格式

```
(expr1) ? (expr2) : (expr3)
```

对 expr1 求值为 TRUE 时的值为 expr2，在 expr1 求值为 FALSE 时的值为 expr3。

自 PHP 5.3 起，可以省略三元运算符中间那部分。表达式 expr1 ?: expr3 在 expr1 求值为 TRUE 时返回 expr1，否则返回 expr3。

10.6.2 实例

以下实例中通过判断 \$_GET 请求中含有 user 值，如果有返回 \$_GET['user']，否则返回 nobody：

```
<?php
$test = '菜鸟教程';
// 普通写法
$username = isset($test) ? $test : 'nobody';
echo $username, PHP_EOL;

// PHP 5.3+ 版本写法
$username = $test ?: 'nobody';
echo $us
```



注意：PHP_EOL 是一个换行符，兼容更大平台。

在 PHP7+ 版本多了一个 NULL 合并运算符 ??，实例如下：

```
<?php
// 如果 $_GET['user'] 不存在返回 'nobody', 否则返回 $_GET['user'] 的值
$username = $_GET['user'] ?? 'nobody';
// 类似的三元运算符
$username = isset($_GET['user']) ? $_GET['user'] : 'nobody';
?>
```

10.7 组合比较符 (PHP 7+)

PHP7+ 支持组合比较符 (combined comparison operator) 也称之为太空船操作符, 符号为 `<=>`。组合比较运算符可以轻松实现两个变量的比较, 当然不仅限于数值类数据的比较。

语法格式如下:

```
$c = $a <=> $b;
```

解析如下:

- 如果 **`$a > $b`**, 则 **`$c`** 的值为 **1**。
- 如果 **`$a == $b`**, 则 **`$c`** 的值为 **0**。
- 如果 **`$a < $b`**, 则 **`$c`** 的值为 **-1**。

实例如下:

```
<?php
// 整型
echo 1 <=> 1; // 0
echo 1 <=> 2; // -1
echo 2 <=> 1; // 1

// 浮点型
echo 1.5 <=> 1.5; // 0
echo 1.5 <=> 2.5; // -1
echo 2.5 <=> 1.5; // 1

// 字符串
echo "a" <=> "a"; // 0
echo "a" <=> "b"; // -1
echo "b" <=> "a"; // 1
?>
```

10.8 运算符优先级

下表按照优先级从高到低列出了运算符。同一行中的运算符具有相同优先级，此时它们的结合方向决定求值顺序。

说明：左 = 从左到右，右 = 从右到左。

Copy of PHP 数组运算符

运算符	名称	描述
无	clone new	clone 和 new
左	[array()
右	++ -- ~ (int) (float) (string) (array) (object) (bool) @	类型和递增／递减
无	instanceof	类型
右	!	逻辑运算符
左	* / %	算术运算符
左	,	多处用到
左	or	逻辑运算符
左	xor	逻辑运算符
左	and	逻辑运算符
右	= += -= *= /= .= %= &= = ^= <<= >>= ==>	赋值运算符
左	? :	三元运算符
左		逻辑运算符
左	&&	逻辑运算符
左		位运算符
左	^	位运算符
左	&	位运算符和引用
无	== != === !== <>	比较运算符
左	<< >>	位运算符
左	+ - .	算术运算符和字符串运算符

运算符优先级中，or 和 ||，&& 和 and 都是逻辑运算符，效果一样，但是其优先级却不一样。

```
<?php
// 优先级： && > = > and
// 优先级： || > = > or

$a = 3;
$b = false;
```



```
$c = $a or $b;  
var_dump($c);           // 这里的 $c 为 int 值3, 而不是 boolean 值 true  
$d = $a || $b;  
var_dump($d);           //这里的 $d 就是 boolean 值 true  
?>
```

以上实例输出结果为：

```
int(3)  
bool(true)
```

10.8.1 括号的使用

我们通过括号的配对来明确标明运算顺序，而非靠运算符优先级和结合性来决定，通常能够增加代码的可读性。

```
<?php  
// 括号优先运算  
  
$a = 1;  
$b = 2;  
$c = 3;  
$d = $a + $b * $c;  
echo $d;  
echo "\n";  
$e = ($a + $b) * $c; // 使用括号  
echo $e;  
echo "\n";  
?>
```

以上实例输出结果为：

```
7  
9
```