Search

Keyboard Shortcuts

?
: This help

j
: Next menu item

k
: Previous menu item

g p
: Previous man page

g n
: Next man page

G
: Scroll to bottom

g g
: Scroll to top

g h
: Goto homepage

g s
: Goto search
(current page)

/
: Focus search box

Побитовые операторы »
« Инкремент и декремент

- Руководство по PHP
- Справочник языка
- Операторы

Change language: Russian

Операторы присваивания

Базовый оператор присваивания обозначается символом «=». Кажется, что это оператор «равно». Это не так. Оператор присваивания означает, что левый операнд получает значение правого выражения (то есть левому операнду «будет присвоено значение»).

Результатом выполнения оператора присваивания будет само присвоенное значение. То есть, результат выполнения выражения «`$a = 3`» будет равен 3. Это разрешает делать трюки наподобие:

```
<?php

$a = ($b = 4) + 5; // Значение переменной $a равно 9, а переменной $b присвоено значение 4.

?>
```

В дополнение к базовому оператору присваивания существуют «комбинированные операторы» для всех бинарных арифметических операций, операций объединения массивов и строковых операций, которые дают присвоить значение в выражении, а затем установить его значение в результат этого выражения. Например:

```
<?php

$a = 3;
$a += 5; // устанавливает для переменной $a значение 8, как если бы было написано: $a = $a + 5;
$b = "Привет";
$b .= "-привет!"; // устанавливает переменной $b значение «Привет-привет!», как и $b = $b . "-привет!";
```

Обратите внимание, что присвоение копирует оригинальную переменную в новую (присвоение по значению), поэтому следующие изменения одной из переменных никак не отразятся на другой. Это также может быть уместным при копировании чего-то вроде большого массива в длинном цикле.

Исключение из обычного для PHP способа присваивания по значению — объекты (object), которые присваиваются по ссылке. Принудительно скопировать объекты по значению можно через ключевое слово clone.

Присваивание по ссылке

Присваивание по ссылке тоже поддерживается, для этого можно использовать синтаксис $var = &$othervar;. Присваивание по ссылке означает, что обе переменные указывают на одни и те же данные и ничего никуда не копируется.

Пример #1 Assigning by reference

```
<?php

$a = 3;
$b = &$a; // $b — это ссылка на переменную $a

print "$a\n"; // печатает 3
print "$b\n"; // печатает 3

$a = 4; // меняем переменную $a

print "$a\n"; // печатает 4
print "$b\n"; // также печатает 4, так как переменная $b — это ссылка на переменную $a,
// а значение переменной $a успело измениться

?>
```

Оператор new автоматически возвращает ссылку, поэтому присваивание результата операции new по ссылке вызывает ошибку.

```
<?php
class C {}
```

```php
$o = &new C;
?>
```

Результат выполнения приведённого примера:

```
Parse error: syntax error, unexpected 'new' (T_NEW) in …
```

Подробно о ссылках рассказано в разделе «Объяснение ссылок».

Операторы арифметического присваивания

Пример	Эквивалент	Операция
$a += $b	$a = $a + $b	Сложение
$a -= $b	$a = $a - $b	Вычитание
$a *= $b	$a = $a * $b	Умножение
$a /= $b	$a = $a / $b	Деление
$a %= $b	$a = $a % $b	Модуль
$a **= $b	$a = $a ** $b	Возведение в степень

Операторы побитового присваивания

Пример	Эквивалент	Операция
$a &= $b	$a = $a & $b	Побитовое И
$a |= $b	$a = $a | $b	Побитовое ИЛИ
$a ^= $b	$a = $a ^ $b	Побитовое исключающее ИЛИ (Xor)
$a <<= $b	$a = $a << $b	Побитовый сдвиг влево
$a >>= $b	$a = $a >> $b	Побитовый сдвиг вправо

Другие операторы присваивания

Пример	Эквивалент	Операция
$a .= $b	$a = $a . $b	Конкатенация строк
$a ??= $b	$a = $a ?? $b	Объединение с Null

Смотрите также

- Арифметические операторы
- Побитовые операторы
- Операторы объединения с null

User Contributed Notes 4 notes

129
Peter, Moscow ¶
13 years ago
Using $text .= "additional text"; instead of $text = $text ."additional text"; can seriously enhance performance due to memory allocation efficiency.

I reduced execution time from 5 sec to .5 sec (10 times) by simply switching to the first pattern for a loop with 900 iterations over a string $text that reaches 800K by the end.

59
Robert Schneider ¶
8 years ago
Be aware of assignments with conditionals. The assignment operator is stronger as 'and', 'or' and 'xor'.

```
<?php
$x = true and false; //$x will be true
$y = (true and false); //$y will be false
?>
```

up
down
34

Hayley Watson ¶
16 years ago

bradlis7 at bradlis7 dot com's description is a bit confusing. Here it is rephrased.

```
<?php
$a = 'a';
$b = 'b';

$a .= $b .= "foo";

echo $a,"\n",$b;?>
```

outputs

```
abfoo
bfoo
```

Because the assignment operators are right-associative and evaluate to the result of the assignment

```
<?php
$a .= $b .= "foo";
?>
```

is equivalent to

```
<?php
$a .= ($b .= "foo");
?>
```

and therefore

```
<?php
$b .= "foo";
$a .= $b;
?>
```

up
down
12

asc at putc dot de ¶
8 years ago

PHP uses a temporary variable for combined assign-operators (unlike JavaScript), therefore the left-hand-side (target) gets evaluated last.

```
Input:
$a += $b + $c;

Meaning:
$a = ($b + $c) + $a;

Not:
$a = $a + ($b + $c);
```

This can be important if the target gets modified inside the expression.

```
$a = 0;
$a += (++$a) + (++$a); // yields 5 (instead of 4)
```

＋add a note

- Операторы
  - Приоритет
  - Арифметика