Assignment Operators

- Assignment by Reference
- Arithmetic Assignment Operators
- Bitwise Assignment Operators
- Other Assignment Operators

The basic assignment operator is "=". Your first inclination might be to think of this as "equal to". Don't. It really means that the left operand gets set to the value of the expression on the right (that is, "gets set to").

The value of an assignment expression is the value assigned. That is, the value of "`$a = 3`" is 3. This allows you to do some tricky things:

Nested Assignments

```
<?php
$a = ($b = 4) + 5; // $a is equal to 9 now, and $b has been set to 4.
var_dump($a);
?>
```

In addition to the basic assignment operator, there are "combined operators" for all of the binary arithmetic, array union and string operators that allow you to use a value in an expression and then set its value to the result of that expression. For example:

Combined Assignments

```
<?php
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b . "There!";
```

```
var_dump($a, $b);
?>
```

Note that the assignment copies the original variable to the new one (assignment by value), so changes to one will not affect the other. This may also have relevance if you need to copy something like a large array inside a tight loop.

An exception to the usual assignment by value behaviour within PHP occurs with `object`s, which are assigned by reference. Objects may be explicitly copied via the clone keyword.

Assignment by Reference

Assignment by reference is also supported, using the "`$var = ENTITY$othervar;`" syntax. Assignment by reference means that both variables end up pointing at the same data, and nothing is copied anywhere.

Assigning by reference

```
<?php
$a = 3;
$b = &$a; // $b is a reference to $a

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

$a = 4; // change $a

print "$a\n"; // prints 4
print "$b\n"; // prints 4 as well, since $b is a reference to $a, which has
              // been changed
?>
```

The new operator returns a reference automatically, as such assigning the result of new by reference is an error.

new Operator By-Reference

```
<?php
class C {}

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

ENTITY

```
Parse error: syntax error, unexpected token ";", expecting "("
```

More information on references and their potential uses can be found in the References Explained section of the manual.

Arithmetic Assignment Operators

Example	Equivalent	Operation
$a += $b	$a = $a + $b	Addition
$a -= $b	$a = $a - $b	Subtraction
$a *= $b	$a = $a * $b	Multiplication
$a /= $b	$a = $a / $b	Division
$a %= $b	$a = $a % $b	Modulus
$a **= $b	$a = $a ** $b	Exponentiation

Bitwise Assignment Operators

Example	Equivalent	Operation
$a ENTITY= $b	$a = $a ENTITY $b	Bitwise And
$a |= $b	$a = $a | $b	Bitwise Or
$a ^= $b	$a = $a ^ $b	Bitwise Xor
$a ENTITYENTITY= $b	$a = $a ENTITYENTITY $b	Left Shift

Example	Equivalent	Operation
$a ENTITYENTITY= $b	$a = $a ENTITYENTITY $b	Right Shift

Other Assignment Operators

Example	Equivalent	Operation
$a .= $b	$a = $a . $b	String Concatenation
$a ??= $b	$a = $a ?? $b	Null Coalesce

ENTITY

arithmetic operators, bitwise operators, null coalescing operator

---

□□□ #2
□ □□□ □□□ 2026-10-31 17:41:05 UTC
□ □□□ □□□ 2026-10-31 17:41:06 UTC