

Arithmetic Operators

This lesson discusses arithmetic operators and how you can use them to perform basic arithmetic in PHP.

We'll cover the following

- Binary Operators
- Unary Operators
- Quick Quiz on Arithmetic Operators

Arithmetic operators, as the name suggests, are used to perform basic arithmetic. They are further divided into types:

- Binary Operators
- Unary Operators

Binary Operators

Binary operators are the ones that take two values and perform an arithmetic operation on them.



The following table discusses the various arithmetic operators and their functions.

Operator	Function	Example
+	Addition	<code>\\$a + \\$b</code>
-	Subtraction	<code>\\$a - \\$b</code>

*	Multiplication	<code>\\$a * \\$b</code>
/	Division	<code>\\$a / \\$b</code>
%	Modulus	<code>\\$a % \\$b</code>

The modulus operator(%) returns the remainder when `$a` is divided by `$b`.

Run the code below to see how these arithmetic operators work in PHP.

```
<?php
$a = 18;
$b = 4;
echo "$a + $b = " . ($a + $b); //addition
echo "\n";
echo "$a - $b = " . ($a - $b); //subtraction
echo "\n";
echo "($a * $b) = " . ($a * $b); //multiplication
echo "\n";
echo "$a / $b = " . ($a / $b); //division
echo "\n";
echo "$a % $b = " . ($a % $b); //modulus
echo "\n";
?>
```



Unary Operators

Unary operators are the type of arithmetic operators that perform arithmetic on only one value.

`$myvar`  variable
`++`
 operator

There are two unary operators:

- Incrementing operator: `++`

- Decrementing operator: `--`

Variables can be incremented by **1** using the `++` operator and can be decremented by **1** using the operator `--`. They can either precede or succeed variables, resulting in different executions of the code. An example of this is shown below:

```
<?php
$i = 1;
echo $i . "\n"; // Prints 1
// Pre-increment operator increments $i by one, then returns $i
echo ++$i . "\n"; // Prints 2
// Pre-decrement operator decrements $i by one, then returns $i
echo --$i . "\n"; // Prints 1
// Post-increment operator returns $i, then increments $i by one
echo $i++ . "\n"; // Prints 1 (but $i value is now 2)
// Post-decrement operator returns $i, then decrements $i by one
echo $i-- . "\n"; // Prints 2 (but $i value is now 1)

?>
```



Quick Quiz on Arithmetic Operators



What is the output of the following script?

```
<?php
$myvar = 5;
echo $myvar++;
?>
```

[Retake Quiz](#)

Now that we know about arithmetic operators, let's learn how to deal with multiple operators in the same expression.