Operators

Operators

Operators carry out operations on variables and values.

Arithmetic Operators

Arithmetic operators work with numeric values to perform common arithmetical operations.

Operator	Name	Example
+	Addition	$x + $y
-	Subtraction	$x - $y
*	Multiplication	$x * $y
/	Division	$x / $y
%	Modulus	$x % $y

Example:

```
<?php
 $num1 = 8;
 $num2 = 6;

 //Addition
 echo $num1 + $num2; //14

 //Subtraction
 echo $num1 - $num2; //2

 //Multiplication
 echo $num1 * $num2; //48

 //Division
  echo $num1 / $num2; //1.33333333333
?>
```

Try It Yourself

Tap **Try It Yourself** to play around with the code!

Modulus

The **modulus** operator, represented by the % sign, returns the remainder of the division of the first operand by the second operand:

```
<?php
 $x = 14;
 $y = 3;
 echo $x % $y; // 2
?>
```

Try It Yourself

If you use floating point numbers with the modulus operator, they will be converted to **integers** before the operation.

Increment & Decrement

The **increment** operators are used to increment a variable's value.
The **decrement** operators are used to decrement a variable's value.

```
$x++; // equivalent to $x = $x+1;
$x--; // equivalent to $x = $x-1;
```

Increment and decrement operators either precede or follow a variable.

```
$x++; // post-increment
$x--; // post-decrement
++$x; // pre-increment
--$x; // pre-decrement
```

The difference is that the post-increment returns the original value **before** it changes the variable, while the pre-increment changes the variable first and then returns the value.
Example:

```
$a  = 2; $b = $a++; // $a=3,  $b=2
$a  = 2; $b = ++$a; // $a=3,  $b=3
```

The increment operators are used to increment a variable's value.

Assignment Operators

Assignment operators are used to write values to variables.

```
$num1 = 5;
$num2 = $num1;
```

$num1 and **$num2** now contain the value of 5.

Assignments can also be used in conjunction with arithmetic operators.

Assignment	Same as...	Description
x+=y	x = x + y	**Addition**
x-=y	x = x - y	**Subtraction**
x*=y	x = x * y	**Multiplication**
x/=y	x = x / y	**Division**
x%=y	x = x % y	**Modulus**

Example:

```
<?php
 $x = 50;
 $x += 100;
 echo $x;

 // Outputs: 150
?>
```

Try It Yourself

Tap **Try It Yourself** to play around with the code!

Comparison Operators

Comparison operators compare two values (numbers or strings).
Comparison operators are used inside conditional statements, and evaluate to either **TRUE** or **FALSE**.

Operator	Name	Example	Result
==	Equal	**$x == $y**	Returns true if **$x is equal to $y**
===	Identical	**$x === $y**	Returns true if **$x is equal to $y**, and they are of the same type
!=	Not equal	**$x != $y**	Returns true if **$x is not equal to $y**
<>	Not equal	**$x <> $y**	Returns true if **$x is not equal to $y**
!==	Not identical	**$x !== $y**	Returns true if **$x is not equal to $y,** or they are not of the same type

Be careful using == and === ; the first one doesn't check the type of data.

Comparison Operators

Additional comparison operators:

Operator	Name	Example	Result
>	Greater than	$x > $y	Returns true if **$x is greater than $y**
<	Less than	$x < $y	Returns true if **$x is less than $y**
>=	Greater than or equal to	$x >= $y	Returns true if **$x is greater than or equal to $y**
<=	Less than or equal to	$x <= $y	Returns true if **$x is less than or equal to $y**

> The PHP comparison operators are used to compare two values (number or string).

Logical Operators

Logical operators are used to combine conditional statements.

Operator	Name	Example	Result
and	And	$x and $y	True if **both $x and $y are true**
or	Or	$x or $y	True if **either $x or $y is true**
xor	Xor	$x xor $y	True if **either $x or $y is true, but not both**
&&	And	$x && $y	True if **both $x and $y are true**
||	Or	$x || $y	True if **either $x or $y is true**
!	Not	!$x	True if **$x is not true**

> You can combine as many terms as you want. Use parentheses () for precedence.

End.