



Listen and Learn

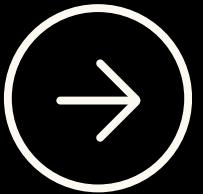
# PHP OPERATORS

Presented by PaSig



# PHP OPERATORS

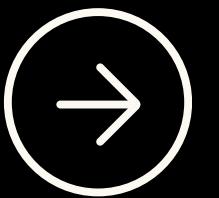
OPERATORS ARE USED TO PERFORM OPERATIONS ON VARIABLES AND VALUES.  
PHP DIVIDES THE OPERATORS IN THE FOLLOWING GROUPS:



- 1 Arithmetic operators
- 2 Assignment operators
- 3 Comparison operators
- 4 Increment/Decrement operators
- 5 Logical operators
- 6 String operators
- 7 Array operators
- 8 Conditional assignment operators

# PHP ARITHMETIC OPERATORS

THE PHP ARITHMETIC OPERATORS ARE USED WITH NUMERIC VALUES TO PERFORM COMMON ARITHMETICAL OPERATIONS, SUCH AS ADDITION, SUBTRACTION, MULTIPLICATION ETC.



Operator	Name	Example	Result
+	Addition	$\$x + \$y$	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \$y$	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \$y$	Product of $\$x$ and $\$y$
/	Division	$\$x / \$y$	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \$y$	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \$y$	Result of raising $\$x$ to the $\$y$ 'th power

# EXAMPLE OF ARITHMETIC OPERATIONS:

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
$y = 6;

echo $x + $y;
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
$y = 6;

echo $x - $y;
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
$y = 6;

echo $x * $y;
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
$y = 6;

echo $x / $y;
?>

</body>
</html>
```

1.666666666667

```
<!DOCTYPE html> 2
<html>
<body>

<?php
$x = 10;
$x /= 5;

echo $x;
?>

</body>
</html>
```

16

4

60

```
<!DOCTYPE html> 3
<html>
<body>

<?php
$x = 15;
$x %= 4;

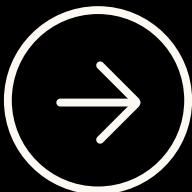
echo $x;
?>

</body>
</html>
```

# PHP ASSIGNMENT OPERATORS

THE PHP ASSIGNMENT OPERATORS ARE USED WITH NUMERIC VALUES TO WRITE A VALUE TO A VARIABLE.

THE BASIC ASSIGNMENT OPERATOR IN PHP IS "`=`". IT MEANS THAT THE LEFT OPERAND GETS SET TO THE VALUE OF THE ASSIGNMENT EXPRESSION ON THE RIGHT.



Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

# EXAMPLES OF PHP ASSIGNMENT OPERATOR:

## Assignment

`x = y`

`x += y`

`x -= y`

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 10;
echo $x;
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 20;
$x += 100;

echo $x;
?>

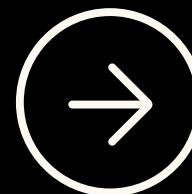
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 50;
$x -= 30;

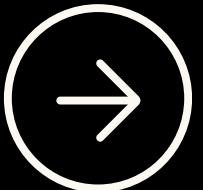
echo $x;
?>

</body>
</html>
```



# PHP COMPARISON OPERATORS

THE PHP COMPARISON OPERATORS ARE USED TO COMPARE TWO VALUES (NUMBER OR STRING):



Operator	Name	Example	Result
<code>==</code>	Equal	<code>\$x == \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code>
<code>===</code>	Identical	<code>\$x === \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code> , and they are of the same type
<code>!=</code>	Not equal	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>&lt;&gt;</code>	Not equal	<code>\$x &lt;&gt; \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Not identical	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code> , or they are not of the same type
<code>&gt;</code>	Greater than	<code>\$x &gt; \$y</code>	Returns true if <code>\$x</code> is greater than <code>\$y</code>
<code>&lt;</code>	Less than	<code>\$x &lt; \$y</code>	Returns true if <code>\$x</code> is less than <code>\$y</code>
<code>&gt;=</code>	Greater than or equal to	<code>\$x &gt;= \$y</code>	Returns true if <code>\$x</code> is greater than or equal to <code>\$y</code>
<code>&lt;=</code>	Less than or equal to	<code>\$x &lt;= \$y</code>	Returns true if <code>\$x</code> is less than or equal to <code>\$y</code>
<code>&lt;=&gt;</code>	Spaceship	<code>\$x &lt;=&gt; \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if <code>\$x</code> is less than, equal to, or greater than <code>\$y</code> . Introduced in PHP 7.

# EXAMPLES OF PHP COMPARISON OPERATOR:

## Operator

`==`

`== =`

```
<!DOCTYPE html>
<html>
<body>

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

var_dump($x == $y); // returns true because values are equal
?>
```

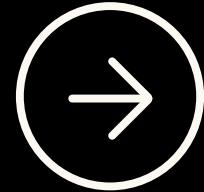
```
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
```

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

```
var_dump($x === $y); // returns false because types are not equal
?>
```

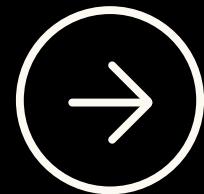
```
</body>
</html>
```



# PHP INCREMENT / DECREMENT OPERATORS

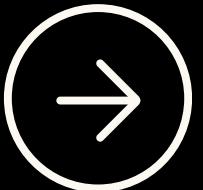
THE PHP INCREMENT OPERATORS ARE USED TO INCREMENT A VARIABLE'S VALUE.

THE PHP DECREMENT OPERATORS ARE USED TO DECREMENT A VARIABLE'S VALUE.



Operator	Same as...	Description
<code>++\$x</code>	Pre-increment	Increments <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x++</code>	Post-increment	Returns <code>\$x</code> , then increments <code>\$x</code> by one
<code>--\$x</code>	Pre-decrement	Decrements <code>\$x</code> by one, then returns <code>\$x</code>
<code>\$x--</code>	Post-decrement	Returns <code>\$x</code> , then decrements <code>\$x</code> by one

# EXAMPLES OF PHP INCREMENT / DECREMENT OPERATORS



## Operator

`++$x`

`$x++`

`--$x`

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
$x = 10;
echo ++$x;
?>
```

```
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
$x = 10;
echo $x++;
?>
```

```
</body>
</html>
```

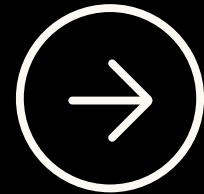
```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
$x = 10;
echo --$x;
?>
```

```
</body>
</html>
```

# PHP LOGICAL OPERATORS

THE PHP LOGICAL OPERATORS ARE USED TO COMBINE CONDITIONAL STATEMENTS.



Operator	Name	Example	Result
and	And	<code>\$x and \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
or	Or	<code>\$x or \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
xor	Xor	<code>\$x xor \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true, but not both
<code>&amp;&amp;</code>	And	<code>\$x &amp;&amp; \$y</code>	True if both <code>\$x</code> and <code>\$y</code> are true
<code>  </code>	Or	<code>\$x    \$y</code>	True if either <code>\$x</code> or <code>\$y</code> is true
!	Not	<code>!\$x</code>	True if <code>\$x</code> is not true

# EXAMPLES OF PHP LOGICAL OPERATOR:

Operator

and

or

```
<!DOCTYPE html>
<html>
<body>

<h1>The or Operator</h1>

<p>Write a message if at least one of the conditions are true.</p>
```

```
<?php
$x = 100;
$y = 50;

if ($x == 100 or $y == 80) {
    echo "Hello world!";
}
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

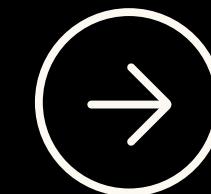
<h1>The and Operator</h1>

<p>Write a message if both conditions are true.</p>
```

```
<?php
$x = 100;
$y = 50;

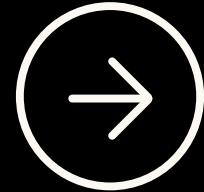
if ($x == 100 and $y == 50) {
    echo "Hello world!";
}
?>

</body>
</html>
```



# PHP STRING OPERATORS

PHP HAS TWO OPERATORS THAT ARE SPECIALLY DESIGNED FOR STRINGS.



Operator	Name	Example	Result
.	Concatenation	<code>\$txt1 . \$txt2</code>	Concatenation of <code>\$txt1</code> and <code>\$txt2</code>
.=	Concatenation assignment	<code>\$txt1 .= \$txt2</code>	Appends <code>\$txt2</code> to <code>\$txt1</code>

# EXAMPLES OF PHP STRING OPERATOR:

Operator
.
.=

```
<!DOCTYPE html>
<html>
<body>

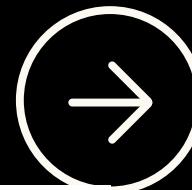
<?php
$txt1 = "Hello";
$txt2 = " world!";
$txt1 .= $txt2;
echo $txt1;
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

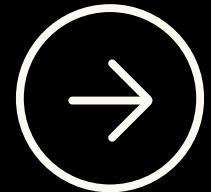
<?php
$txt1 = "Hello";
$txt2 = " world!";
echo $txt1 . $txt2;
?>

</body>
</html>
```



# PHP ARRAY OPERATORS

THE PHP ARRAY OPERATORS ARE USED TO COMPARE ARRAYS.



Operator	Name	Example	Result
+	Union	<code>\$x + \$y</code>	Union of <code>\$x</code> and <code>\$y</code>
<code>==</code>	Equality	<code>\$x == \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs
<code>===</code>	Identity	<code>\$x === \$y</code>	Returns true if <code>\$x</code> and <code>\$y</code> have the same key/value pairs in the same order and of the same types
<code>!=</code>	Inequality	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>&lt;&gt;</code>	Inequality	<code>\$x &lt;&gt; \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Non-identity	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not identical to <code>\$y</code>

# EXAMPLES OF PHP ARRAY OPERATOR

Operator

+

==

```
<!DOCTYPE html>
>
>

<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");

var_dump($x == $y);
?>

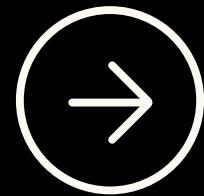
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");

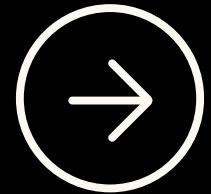
print_r($x + $y); // union of $x and $y
?>

</body>
</html>
```



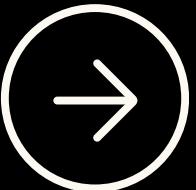
# PHP CONDITIONAL ASSIGNMENT OPERATORS

THE PHP CONDITIONAL ASSIGNMENT OPERATORS ARE USED TO SET A VALUE DEPENDING ON CONDITIONS:



Operator	Name	Example	Result
?:	Ternary	<code>\$x = expr1 ? expr2 : expr3</code>	Returns the value of \$x. The value of \$x is expr2 if expr1 = TRUE. The value of \$x is expr3 if expr1 = FALSE
??	Null coalescing	<code>\$x = expr1 ?? expr2</code>	Returns the value of \$x. The value of \$x is expr1 if expr1 exists, and is not NULL. If expr1 does not exist, or is NULL, the value of \$x is expr2.  <b>Introduced in PHP 7</b>

# EXAMPLE OF PHP CONDITIONAL ASSIGNMENT OPERATORS



## Operator

?:

```
<!DOCTYPE html>
<html>
<body>

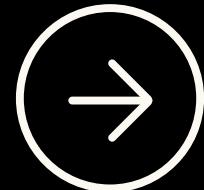
<?php
    // if empty($user) = TRUE, set $status = "anonymous"
    echo $status = (empty($user)) ? "anonymous" : "logged in";
    echo("<br>");

    $user = "John Doe";
    // if empty($user) = FALSE, set $status = "logged in"
    echo $status = (empty($user)) ? "anonymous" : "logged in";
?>

</body>
</html>
```

# EXERCISES;

```
php  
  
<?php  
$num1 = 10;  
$num2 = 5;  
  
echo "Guess the operator used: 10 ? 5 = " . ($num1 + $num2) . "<br>";  
echo "Addition: 10 + 5 = " . ($num1 + $num2) . "<br>";  
echo "Subtraction: 10 - 5 = " . ($num1 - $num2) . "<br>";  
echo "Multiplication: 10 * 5 = " . ($num1 * $num2) . "<br>";  
echo "Division: 10 / 5 = " . ($num1 / $num2) . "<br>";  
??>
```



```
php  
  
<?php  
$num = 10;  
  
echo "Guess the operator used: \$num += 5<br>"; // Guess here!  
$num += 5; // Adds 5 to $num  
echo "After +=: $num<br>";  
  
$num *= 2; // Multiplies $num by 2  
echo "After *=: $num<br>";  
??>
```

```
<?php  
$num1 = 10;  
$num2 = 20;  
  
echo "Guess the operator used: 10 ? 20 = " . var_export($num1 == $num2, true) . "<br>";  
echo "Equal (==): " . var_export($num1 == $num2, true) . "<br>";  
echo "Not equal (!=): " . var_export($num1 != $num2, true) . "<br>";  
echo "Less than (<): " . var_export($num1 < $num2, true) . "<br>";  
?>
```

php

```
<?php  
$array1 = ["a" => 1, "b" => 2];  
$array2 = ["b" => 3, "c" => 4];  
  
echo "Guess the operator used: \$array1 ? \$array2<br>"; // Guess here!  
$union = $array1 + $array2; // Merge arrays  
echo "Union (+): ";  
print_r($union);  
echo "<br>";  
  
$isEqual = $array1 == $array2;  
echo "Equality (==): " . var_export($isEqual, true) . "<br>";  
?>
```

```
<?php  
$num = 5;  
  
echo "Guess the operator used: ?\$num = " . ++$num . "<br>";  
echo "Pre-increment (++): \$num<br>";  
  
--$num; // Decrement $num by 1  
echo "After pre-decrement (--): \$num<br>";  
?>
```