

\_\_METHOD\_\_

The class method name. (Added in PHP 5.0.0) The method name is returned as it was declared (case-sensitive).

## 6. PHP – OperatorTypes

**What is Operator?** Simple answer can be given using expression *4 + 5 is equal to 9*. Here 4 and 5 are called operands and + is called operator. PHP language supports following type of operators.

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Assignment Operators
- Conditional (or ternary) Operators

Let's have a look on all operators one by one.

### Arithmetic Operators

The following arithmetic operators are supported by PHP language:

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
+	Adds two operands	A + B will give 30
-	Subtracts second operand from the first	A - B will give -10
*	Multiply both operands	A * B will give 200
/	Divide the numerator by denominator	B / A will give 2
%	Modulus Operator and remainder of after an integer division	B % A will give 0
++	Increment operator, increases integer value by one	A++ will give 11
--	Decrement operator, decreases integer value by one	A-- will give 9

### Example

Try the following example to understand all the arithmetic operators. Copy and paste following PHP program in test.php file and keep it in your PHP Server's document root and browse it using any browser.

```
<html>
<head><title>Arithmetical Operators</title></head>
<body>
<?php
    $a = 42;
    $b = 20;

    $c = $a + $b;
    echo "Addition Operation Result: $c <br/>";
    $c = $a - $b;
    echo "Subtraction Operation Result: $c <br/>";
    $c = $a * $b;
    echo "Multiplication Operation Result: $c <br/>";
    $c = $a / $b;
    echo "Division Operation Result: $c <br/>";
    $c = $a % $b;
    echo "Modulus Operation Result: $c <br/>";
    $c = $a++;
    echo "Increment Operation Result: $c <br/>";
    $c = $a--;
    echo "Decrement Operation Result: $c <br/>";
?>
</body>
</html>
```

This will produce the following result:

```
Addition Operation Result: 62
Subtraction Operation Result: 22
Multiplication Operation Result: 840
Division Operation Result: 2.1
Modulus Operation Result: 2
Increment Operation Result: 42
Decrement Operation Result: 43
```

## Comparison Operators

There are following comparison operators supported by PHP language.

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
==	Checks if the value of two operands are equal or not, if yes, then condition becomes true.	(A == B) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal, then condition becomes true.	(A != B) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes, then condition becomes true.	(A > B) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes, then condition becomes true.	(A < B) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes, then condition becomes true.	(A <= B) is true.

### Example

Try the following example to understand all the comparison operators. Copy and paste the following PHP program in test.php file and keep it in your PHP Server's document root and browse it using any browser.

```
<html>
<head><title>Comparison Operators</title></head>
<body>
<?php
    $a = 42;
    $b = 20;

    if( $a == $b ){
        echo "TEST1 : a is equal to b<br/>";
    }else{
        echo "TEST1 : a is not equal to b<br/>";
    }
```

```

    }

    if( $a > $b ){
        echo "TEST2 : a is greater than b<br/>";
    }else{
        echo "TEST2 : a is not greater than b<br/>";
    }
    if( $a < $b ){
        echo "TEST3 : a is less than b<br/>";
    }else{
        echo "TEST3 : a is not less than b<br/>";
    }
    if( $a != $b ){
        echo "TEST4 : a is not equal to b<br/>";
    }else{
        echo "TEST4 : a is equal to b<br/>";
    }
    if( $a >= $b ){
        echo "TEST5 : a is either greater than or equal to b<br/>";
    }else{
        echo "TEST5 : a is neither greater than nor equal to b<br/>";
    }
    if( $a <= $b ){
        echo "TEST6 : a is either less than or equal to b<br/>";
    }else{
        echo "TEST6 : a is neither less than nor equal to b<br/>";
    }
    ?>
</body>
</html>

```

This will produce the following result:

```

TEST1 : a is not equal to b
TEST2 : a is greater than b
TEST3 : a is not less than b
TEST4 : a is not equal to b
TEST5 : a is either greater than or equal to b

```

```
TEST6 : a is neither less than nor equal to b
```

## Logical Operators

The following logical operators are supported by PHP language.

Assume variable A holds 10 and variable B holds 20 then:

Operator	Description	Example
and	Called Logical AND operator. If both the operands are true, then condition becomes true.	(A and B) is true.
or	Called Logical OR Operator. If any of the two operands are non zero, then condition becomes true.	(A or B) is true.
&&	Called Logical AND operator. If both the operands are non zero, then condition becomes true.	(A && B) is true.
	Called Logical OR Operator. If any of the two operands are non zero, then condition becomes true.	(A    B) is true.
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false.	!(A && B) is false.

### Example

Try the following example to understand all the logical operators. Copy and paste the following PHP program in test.php file and keep it in your PHP Server's document root and browse it using any browser.

```
<html>
<head><title>Logical Operators</title></head>
<body>
<?php
    $a = 42;
```

```
$b = 0;

if( $a && $b ){
    echo "TEST1 : Both a and b are true<br/>";
}else{
    echo "TEST1 : Either a or b is false<br/>";
}

if( $a and $b ){
    echo "TEST2 : Both a and b are true<br/>";
}else{
    echo "TEST2 : Either a or b is false<br/>";
}

if( $a || $b ){
    echo "TEST3 : Either a or b is true<br/>";
}else{
    echo "TEST3 : Both a and b are false<br/>";
}

if( $a or $b ){
    echo "TEST4 : Either a or b is true<br/>";
}else{
    echo "TEST4 : Both a and b are false<br/>";
}

$a = 10;
$b = 20;
if( $a ){
    echo "TEST5 : a is true <br/>";
}else{
    echo "TEST5 : a is false<br/>";
}

if( $b ){
    echo "TEST6 : b is true <br/>";
}else{
    echo "TEST6 : b is false<br/>";
}

if( !$a ){
    echo "TEST7 : a is true <br/>";
}else{
    echo "TEST7 : a is false<br/>";
}
```

```

    }
    if( !$b ){
        echo "TEST8 : b is true <br/>";
    }else{
        echo "TEST8 : b is false<br/>";
    }
}
?>
</body>
</html>

```

This will produce the following result:

```

TEST1 : Either a or b is false
TEST2 : Either a or b is false
TEST3 : Either a or b is true
TEST4 : Either a or b is true
TEST5 : a is true
TEST6 : b is true
TEST7 : a is false
TEST8 : b is false

```

## Assignment Operators

PHP supports the following assignment operators:

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assign the value of A + B into C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A



<code>*=</code>	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	$C *= A$ is equivalent to $C = C * A$
<code>/=</code>	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	$C /= A$ is equivalent to $C = C / A$
<code>%=</code>	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	$C \% = A$ is equivalent to $C = C \% A$

## Example

Try the following example to understand all the assignment operators. Copy and paste the following PHP program in test.php file and keep it in your PHP Server's document root and browse it using any browser.

```
<html>
<head><title>Assignment Operators</title></head>
<body>
<?php
    $a = 42;
    $b = 20;

    $c = $a + $b;    /* Assignment operator */
    echo "Addition Operation Result: $c <br/>";
    $c += $a;    /* c value was 42 + 20 = 62 */
    echo "Add AND Assignment Operation Result: $c <br/>";
    $c -= $a;    /* c value was 42 + 20 + 42 = 104 */
    echo "Subtract AND Assignment Operation Result: $c <br/>";
    $c *= $a;    /* c value was 104 - 42 = 62 */
    echo "Multiply AND Assignment Operation Result: $c <br/>";
    $c /= $a;    /* c value was 62 * 42 = 2604 */
    echo "Division AND Assignment Operation Result: $c <br/>";
    $c %= $a;    /* c value was 2604/42 = 62 */
    echo "Modulus AND Assignment Operation Result: $c <br/>";
?>
</body>
</html>
```

This will produce the following result:

```
Addition Operation Result: 62
Add AND Assignment Operation Result: 104
Subtract AND Assignment Operation Result: 62
Multiply AND Assignment Operation Result: 2604
Division AND Assignment Operation Result: 62
Modulus AND Assignment Operation Result: 20
```

## Conditional Operator

There is one more operator called the conditional operator. It first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

Operator	Description	Example
? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y

Try the following example to understand the conditional operator. Copy and paste the following PHP program in test.php file and keep it in your PHP Server's document root and browse it using any browser.

```
<html>
<head><title>Arithmetical Operators</title><head>
<body>
<?php
    $a = 10;
    $b = 20;

    /* If condition is true then assign a to result otherwise b */
    $result = ($a > $b ) ? $a : $b;
    echo "TEST1 : Value of result is $result<br/>";
    /* If condition is true then assign a to result otherwise b */
    $result = ($a < $b ) ? $a : $b;
    echo "TEST2 : Value of result is $result<br/>";
?>
</body>
</html>
```

This will produce the following result:

TEST1 : Value of result is 20  
TEST2 : Value of result is 10

## Operators Categories

All the operators we have discussed above can be categorized into the following categories:

- Unary prefix operators, which precede a single operand.
- Binary operators, which take two operands and perform a variety of arithmetic and logical operations.
- The conditional operator (a ternary operator), which takes three operands and evaluates either the second or third expression, depending on the evaluation of the first expression.
- Assignment operators, which assign a value to a variable.

## Precedence of PHP Operators

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator: For example,  $x = 7 + 3 * 2$ ; Here  $x$  is assigned 13, not 20 because operator  $*$  has higher precedence than  $+$  so it first get multiplied with  $3*2$  and then adds into 7.

Here operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Category	Operator	Associativity
Unary	! ++ --	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right

Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=	Right to left

