PHP OPERATORS

Introduction

- Operators are symbols that tell the PHP processor to perform certain actions.

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- String operators
- Array operators
- Conditional assignment operators
- Logical operators

Arithmetic operators

- The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Description	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

Arithmetic operators(contd.)

```
<?php
$x = 10;
$y = 4;
echo($x + $y); // 0utputs: 14
echo "<br>";
echo($x - $y); // 0utputs: 6
echo "<br>";
echo($x * $y); // 0utputs: 40
echo "<br>";
echo($x / $y); // 0utputs: 2.5
echo "<br>";
echo($x % $y); // 0utputs: 2
echo "<br>";
echo($x ** $y); // 0utputs: 10000
?>
```

Assignment operators

- The PHP assignment operators are used with numeric values to write a value to a variable.

Operator	Description	Example	Is The Same As
=	Assign	$x = $y	$x = $y
+=	Add and assign	$x += $y	$x = $x + $y
-=	Subtract and assign	$x -= $y	$x = $x - $y
*=	Multiply and assign	$x *= $y	$x = $x * $y
/=	Divide and assign quotient	$x /= $y	$x = $x / $y
%=	Divide and assign modulus	$x %= $y	$x = $x % $y

Assignment operators(contd.)

```
<?php
$x = 10;
echo $x; // Outputs: 10
echo "<br>";
$x = 20;
$x += 30;
echo $x; // Outputs: 50
echo "<br>";
$x = 50;
$x -= 20;
echo $x; // Outputs: 30
echo "<br>";
$x = 5;
$x *= 25;
echo $x; // Outputs: 125
echo "<br>";
$x = 50;
$x /= 10;
echo $x; // Outputs: 5
echo "<br>";
$x = 100;
$x %= 15;
echo $x; // Outputs: 10
?>
```

Assignment operators(contd.)

OUTPUT:

10

50

30

125

5

10

Comparison operators

- The comparison operators are used to compare two values in a Boolean fashion.

Comparison operators(contd.)

Operator	Name	Example	Result
==	Equal	$x == $y	True if $x is equal to $y
===	Identical	$x === $y	True if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	True if $x is not equal to $y
<>	Not equal	$x <> $y	True if $x is not equal to $y
!==	Not identical	$x !== $y	True if $x is not equal to $y, or they are not of the same type
<	Less than	$x < $y	True if $x is less than $y
>	Greater than	$x > $y	True if $x is greater than $y
>=	Greater than or equal to	$x >= $y	True if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	True if $x is less than or equal to $y

Comparison operators(contd.)

```
<?php
$x = 25;
$y = 35;
$z = "25";
var_dump($x == $z);  // Outputs: boolean true
var_dump($x === $z); // Outputs: boolean false
var_dump($x != $y);  // Outputs: boolean true
var_dump($x !== $z); // Outputs: boolean true
var_dump($x < $y);   // Outputs: boolean true
var_dump($x > $y);   // Outputs: boolean false
var_dump($x <= $y);  // Outputs: boolean true
var_dump($x >= $y);  // Outputs: boolean false
?>
```

Incrementing and Decrementing Operators

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

Operator	Name	Effect
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

Incrementing and Decrementing Operators(contd.)

```
<html>
<?php
$x = 10;
echo ++$x; // Outputs: 11
echo $x;   // Outputs: 11
echo "<br>";
$x = 10;
echo $x++; // Outputs: 10
echo $x;   // Outputs: 11
echo "<br>";
$x = 10;
echo --$x; // Outputs: 9
echo $x;   // Outputs: 9
echo "<br>";
$x = 10;
echo $x--; // Outputs: 10
echo $x;   // Outputs: 9
?>
</html>
```

Incrementing and Decrementing Operators(contd.)

OUTPUT:

1111

1011

99

109

String Operators

- The string operators are used to perform the operation on strings.

Operator	Description	Example	Result
.	Concatenation	$str1 . $str2	Concatenation of $str1 and $str2
.=	Concatenation assignment	$str1 .= $str2	Appends the $str2 to the $str1

String Operators(contd.)

```
<?php
$x = "Hello";
$y = " World!";
echo $x . $y; // Outputs: Hello World!
echo "<br>";
$x .= $y;
echo $x; // Outputs: Hello World!
?>
```

OUTPUT:

Hello World!

Hello World!

PHP Conditional Assignment Operators

- The PHP conditional assignment operators are used to set a value depending on conditions:

?:	Ternary	$x = *expr1* ? *expr2* : *expr3*	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	$x = *expr1* ?? *expr2*	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*. Introduced in PHP 7

PHP Conditional Assignment Operators(contd.) – Ternary operator

```
<?php
echo $status = (empty($user)) ? "anonymous" : $user;
echo "<br>";

$user = "Michael";
echo $status = (empty($user)) ? "anonymous" : $user;
?>
```

OUTPUT:

anonymous

Michael

PHP Conditional Assignment Operators(contd.) – Null coalescing

```
<?php
echo $status = $user ?? 'anonymous';
echo "<br>";

$user = "Michael";
echo $status = $user ?? 'anonymous';
?>
```

OUTPUT:

anonymous

Michael

Logical Operators

- The logical operators are typically 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

Logical Operators(contd.)

```
<?php
$year = 2014;
// Leap years are divisible by 400 or by 4 but not 100
if(($year % 400 == 0) || (($year % 100 != 0) && ($year % 4 == 0))){
    echo "$year is a leap year.";
} else{
    echo "$year is not a leap year.";
}
?>
```

OUTPUT:

2014 is not a leap year.