PHP 5 Operators

What is Operators in PHP

Operators are symbols that tell the PHP processor to perform certain actions. For example, the addition (+) symbol is an operator that tells PHP to add two variables or values, while the greater-than (>) symbol is an operator that tells PHP to compare two values. Operators are used to perform operations on variables and values.

PHP divides the operators in the following groups:

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

PHP Arithmetic Operators

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

Example:

```
<?php

$x = 10;

$y = 4;

echo($x + $y);

echo "<br>";
```

```
echo($x - $y);
echo "<br>";
echo($x * $y);
echo "<br>";
echo($x / $y);
echo "<br>";
echo($x % $y);
?>
```

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.

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`

```
<?php
$x = 10;
echo $x;
echo "<br>";
$x = 20;
$x += 30;
echo $x;
echo "<br>";
```

```
$x = 50;

$x -= 20;

echo $x;

echo "<br>";

 $x = 5;

$x *= 25;

echo $x;

echo "<br>";

$x = 50;

$x /= 10;

echo $x;

echo "<br>";

$x = 100;

$x %= 15;

echo $x;

?>
```

PHP Comparison Operators

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

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

```
<?php

$x = 25;
$y = 35;
$z = "25";

var_dump($x == $z);

echo "<br>";

var_dump($x === $z);

echo "<br>";

var_dump($x != $y);

echo "<br>";

var_dump($x !== $z);

echo "<br>";

var_dump($x < $y);

echo "<br>";

var_dump($x > $y);

echo "<br>";

var_dump($x <= $y);

echo "<br>";

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

```
?>
```

PHP 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

```
<?php
$x = 10;
echo ++$x;
echo "<br>";
echo $x;
echo "<hr>";
$x = 10;
echo $x++;
echo "<br>";
echo $x;
echo "<hr>";
$x = 10;
echo --$x;
echo "<br>";
echo $x;
echo "<hr>";
```

```
$x = 10;

echo $x--;

echo "<br>";

echo $x;

?>
```

PHP 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

```
<?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.";

}

?>
```

PHP String Operators

There are two operators which are specifically designed for strings.

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

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

PHP Array Operators

The array operators are used to compare arrays:

Operator	Name	Example	Result
+	Union	`$x + $y`	Union of $x and $y
==	Equality	`$x == $y`	True if $x and $y have the same key/value pairs
===	Identity	`$x === $y`	True if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	`$x != $y`	True if $x is not equal to $y
<>	Inequality	`$x <> $y`	True if $x is not equal to $y
!==	Non-identity	`$x !== $y`	True if $x is not identical to $y

```
<?php
$x = array("a" => "Red", "b" => "Green", "c" => "Blue");
$y = array("u" => "Yellow", "v" => "Orange", "w" => "Pink");
$z = $x + $y; // Union of $x and $y
var_dump($z);
echo "<hr>";

var_dump($x == $y);
echo "<br>";

var_dump($x === $y);
echo "<br>";

var_dump($x != $y);
echo "<br>";

var_dump($x <> $y);
echo "<br>";

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

Thanks...