OPERATORS

Operators are used to perform operations on variables and values.

1.PHP Arithmetic Operators

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

Addition +:

```
$sum = 5 + 3;  // $sum is 8
```

Subtraction -:

```
$difference = 10 - 4;  // $difference is 6
```

Multiplication *:

```
$product = 3 * 7;  // $product is 21
```

Division /:

```
$quotient = 15 / 3;  // $quotient is 5
```

Modulus % (Remainder):

```
$remainder = 10 % 3;  // $remainder is 1
```

2.PHP Assignment Operators:

Assignment operators assign values to variables.

Assignment =:

```
$x = 10;  // Assigns the value 10 to $x
```

Addition Assignment +=:

```
$x += 5;  // Equivalent to $x = $x + 5
```

Subtraction Assignment -=:

```
$y -= 3;  // Equivalent to $y = $y - 3
```

Multiplication Assignment *=:

```
$z *= 2;  // Equivalent to $z = $z * 2
```

Division Assignment /=:

```
$a /= 4;  // Equivalent to $a = $a / 4
```

3.PHP Increment and Decrement Operators:

These operators increase or decrease the value of a variable.

Increment ++:

```
$count = 5;
$count++;  // $count is now 6
```

Decrement --:

```
$total = 8;
$total--;  // $total is now 7
```

4.PHP Comparison Operators:

Comparison operators are used to compare values.

Equal ==:

```
$num1 = 10;
$num2 = "10";
$isEqual = ($num1 == $num2);  // $isEqual is true (loose equality)
```

Identical ===:

```
$num1 = 10;
$num2 = "10";
```

```
$isIdentical = ($num1 === $num2);  // $isIdentical is false (strict equality)
```

Not Equal != or <>:

```
$a = 5;

$b = 7;

$notEqual = ($a != $b);  // $notEqual is true
```

Not Identical !==:

```
$x = 10;

$y = "10";

$notIdentical = ($x !== $y);  // $notIdentical is true
```

Greater Than > and Less Than <:

```
$val1 = 15;

$val2 = 10;

$isGreater = ($val1 > $val2);  // $isGreater is true
```

Greater Than or Equal To >= and Less Than or Equal To <=:

```
$num1 = 20;

$num2 = 20;

$isGreaterOrEqual = ($num1 >= $num2);  // $isGreaterOrEqual is true
```

5.PHP Logical Operators:

Logical operators are used to combine or manipulate boolean values.

Logical AND && or and:

```
$x = true;

$y = false;

$result = ($x && $y);  // $result is false
```

Logical OR || or or:

```
$a = true;
$b = false;
$output = ($a || $b);  // $output is true
```

Logical NOT !:

```
$isTrue = true;
$isNotTrue = !$isTrue;  // $isNotTrue is false
```

6.PHP String Operators:

String operators are used to concatenate strings.

Concatenation .:

```
$str1 = "Hello, ";
$str2 = "World!";
$combinedStr = $str1 . $str2;  // $combinedStr is "Hello, World!"
```

7. PHP Ternary Operator:

The ternary operator is a shorthand way to write simple if-else statements.

Ternary ? :

```
$age = 20;
$message = ($age >= 18) ? "Adult" : "Minor";
```

8. PHP Null Coalescing Operator:

Introduced in PHP 7, the null coalescing operator (??) provides a concise way to handle null values.

Null Coalescing ??

```
$username = $_GET['user'] ?? 'Guest';
```

These are the most commonly used operators in PHP. Understanding how to use these operators is crucial for writing efficient and expressive PHP code.