Logical Operators

This lesson covers logical operators and how we can use them to evaluate conditions in PHP.

We'll cover the following

What are Logical Operators?

Logical operators are used to combine conditional statements. This means that the program can take a decision based on multiple conditions. You will learn more about conditional statements later. But for now, suffice it to say, that conditional statements are those that can either be true or false.

Types of Logical Operators

There are two logical operators:

- `&&` or `and`
- `||` or `or`

The And Operator

The `and` or `&&` operator returns `true` if all the statements are `true` and `false` if one or more statements are false. Say we have two statements `a` and `b`. The following table illustrates this behavior.

`a`	`b`	`a&&b` or `a and b`

true	true	true
true	false	false
false	true	false
false	false	false

Run the code below to see how this works.

```
<?php
$x = 5;
$y = 4;
$z = 2;

if ($x > $y && $x > $z) echo "x is greater than y and z\n";

if ($z < $y and $z < $x) echo "z is smaller than y and z\n";
?>
```

The Or Operator

The `or` operator returns `true` if one or more of the conditional statements are `true` and `false` if all the conditions are `false`. The following table illustrates this behavior.

`a`	`b`	`a||b` or `a or b`
true	true	true
true	false	true
false	true	true
false	false	false

Run the examples below to see how this works:

```
<?php
$x = true;
$y = false;
$z = true;
echo ($x || $y);
?>
```

At least one of the two variables $x or $y has to be true for 1 to be printed in the console

```
<?php
$x = 5;
$y = 6;
$z = 2;

if ($x > $y || $x > $z) echo "x is either greater than y or z";
?>
```

$x has to be greater than $y or $z for the if-condition to be considered true

Difference Between `and/or` against `&&/||`

> **Note:** The `&&` and `||` operators have higher precedence than `and` and `or` respectively. See table below:

Since the precedence of `or` and `and` is lower, the `=` operator is executed first.

Evaluation	Result	Evaluated as
$e = false || true	True	$e = (false || true)
$e = false or true	False	($e = false) or true

```
<?php
$e = false or true; // evaluated as ($e = false) || true
echo "When using or, the value of e is: ".$e;
echo "\n";

$e = false || true; // evaluated as $e = (false || true)

echo "When using ||, the value of e is: ".$e;
?>
```

Because of this, it’s safer to use `&&` and `||` instead of `and` and `or` respectively.

Quick Quiz on Logical Operators

1

What is the result of the following code?

```
<?php
$a = true;
$b = false;
$c = true;
if ( $a && $b || $c || (!$c))
    echo "inside if";
else
    echo "inside else";
?>
```

2

What will be the value of `$e` when the following code is executed?

```
<?php
$e = false and true;
?>
```

Retake Quiz

In the next lesson, we will discuss assignment operators.