9. Logické operace v PHP

Často se stává, že potřebujeme kombinovat více podmínek dohromady. Pro tyto účely existují v PHP tzv. **logické operace**. **Logická operace vyhodnocuje vždy hodnotu ano či ne, resp. true nebo false, nebo také jedničku či nulu**.

Logická negace (!)

Jedná se o nejjednodušší operaci, tedy obrácení podmínky. Provádí vlastně jen to, že z pravdivé podmínky udělá nepravdivou a naopak. V PHP se logická negace zapisuje tak, že podmínce přiřadíme znak vykřičník – !

```
<?php
var_dump(10 > 20);
// Tato podmínka je nepravdivá. Vypíše: bool(false)
echo '<br>';
// Zajistí, aby se další výpis vypsal na další řádku.
var_dump(!(10 > 20));
// Podmínka je pravdivá, protože je použita operace negace, která otočila pravdivost podmínky.
// Vypíše: bool(true)
?>
```

Podmínku, kterou chcete obrátit (negovat) je vždy dobré umisťovat do závorek.

Logické operace and a or

Tyto logické operace slouží ke spojení dvou podmínek tak, aby se výsledek tvářil jako jedna podmínka. Obě operace se liší tím, jak vypočítavají výslednou hodnotu podmínky v závislosti na obou původních podmínkách, ze kterých jsou složeny.

Logickou operaci **and lze jinak také vyjádřit slovním spojením „a zároveň“**. Tato operace považuje dvě podmínky za pravdivé jedině tehdy, pokud jsou obě pravdivé. Oproti tomu logické operaci **or (nebo) stačí, aby alespoň jedna ze dvou podmínek platila**.

Logická operace **and**			Logická operace **or**		
Podmínka 1	**Podmínka 2**	**Výsledek**	**Podmínka 1**	**Podmínka 2**	**Výsledek**
pravda	pravda	pravda	pravda	pravda	pravda
pravda	nepravda	nepravda	pravda	nepravda	pravda
nepravda	pravda	nepravda	nepravda	pravda	pravda
nepravda	nepravda	nepravda	nepravda	nepravda	nepravda

Příklad s logickou operací **and**:

```
<?php
$znamka = 1;
// Zde zadáme školní známku. Vyzkoušejte si sem zadávat různá čísla.
var_dump( $znamka >= 1 and $znamka <= 5 );
```

```
// Podmínka je pravdivá, pokud je v proměnné $znamka uložena platná školní známka.
?>
```

Pokud budete do proměnné místo současné jedničky zadávat známky jedna až pět, bude stále výsledkem pravda. Jakmile zadáte známku nižší než jedna nebo vyšší než pět, stane se minimálně jedna ze dvou porovnávaných podmínek nepravdivou a tím pádem celý výsledek funkce var_dump bude nepravda.

Příklad s logickou operací **or**:

```
<?php
$a = 2;
var_dump($a > 10 or $a == 2);
// Výsledek je pravdivá podmínka, protože alespoň jedna podmínka je pravdivá. Vypíše: bool(true)
?>
```

Dvojí zápis logické operace and a or

Kromě klasických slov and a or můžete v PHP použít také zápis **&&** (pravý Alt+C) a **||** (pravý Alt+W, programátoři tento znak nazývají „roura“). Tak že např. tyto dva zápisy jsou totožné:

```
$a > $b and $a != 3
//stejné jako:
$a > $b && $a != 3
```

I když jsou obojí zápisy možné, je mezi mini nepatrný rozdíl a to v tzv. prioritě (přednosti). **Slova and a or mají tzv. nižší – příznivou prioritu**, obvykle odpovídající programátorovým záměrům. Obecně je doporučováno raději používat právě tyto slovní vyjádření.

Neúplné vyhodnocování logických výrazů pomocí and a or

PHP se obecně snaží postupovat co nejúsporněji. **Ctí zásadu „zdravé lenosti“**. Právě v případě logické operace and je to velmi zřejmé. Jak již bylo napsáno, je u logické operace and potřeba k pravdivému výsledku, aby obě dvě části podmínky byly pravdivé. V každém jiném případě bude výsledek nepravdivý. Vysvětlení je v následujícím příkladu:

```
<?php
$a = 1;
$b = 2;
$c = 20;
$d = 20;
var_dump ($a > $b and $c == $d);
// Vypíše: bool(false)
?>
```

PHP postupuje tak, že jakmile zjistí, že je první část podmínky nepravdivá (není pravda, že proměnná a je větší než proměnná b), nemusí se již vůbec zabývat tím, co je ve druhé části podmínky (tedy to, že je proměnná c rovna

proměnné d). Touto částí se již PHP vůbec nezabývá. Tím se skripty nepatrně zrychlují. V našem případě to nebude zcela jistě patrné, ovšem u složitějších programů to může hrát významnou roli. Tento způsob postupu se nevyužívá jen kvůli rychlosti zpracování, ale také z jiných důvodů, kdy je např. na proměnné navázán další složitější skript apod. (viz dále).