

7. Podmínky v PHP

Činnost většiny programů (a to nejen v PHP) je řízena tzv. podmínkou.

Podmínka je jednoduše řečeno zápis, který PHP vyhodnotí a zjistí, zda je pravdivý nebo ne. Další program pak může pokračovat a to jak pro stav, kdy je podmínka pravdivá (je splněna) či naopak nepravdivá (není splněna).

Podmíněný příkaz **if**

Podmínky v PHP jsou řešeny pomocí **příkazu if**. Dělíme jej na úplný a neúplný.

Úplný podmíněný příkaz **if**

Úplný podmíněný příkaz je takový, který řeší nejen stav, kdy je daná podmínka splněna, ale i stav, kdy podmínka není splněna. Celá konstrukce příkazu **if** vypadá zjednodušeně takto:

```
if (podmínka)
příkaz který se má vykonat pokud je podmínka pravdivá
else
příkaz který se má vykonat pokud podmínka není pravdivá
```

Jak je z příkladu patrné, podmínka, která se má vyhodnotit, je za slůvkem **if** (volný překlad „pokud“) v kulatých závorkách. Může to být jakýkoliv výraz, kterému PHP rozumí. Může to být nějaká hodnota, třeba číslo, nebo řetězec, proměnná nebo matematická operace, apod. Po vyhodnocení podmínky s výsledkem „pravda“ se vykoná nějaký další krok, umístěný hned na dalším řádku. Pokud však výsledkem vyhodnocení podmínky je nepravda (podmínka není splněna), vykoná se to, co následuje za slovem **else** (volný překlad „jinak“). Příklad:

```
<?php
if (10 > 2)
echo 'Podmínka je pravdivá';
else
echo 'Podmínka není pravdivá';
?>
```

V tomto případě se po spuštění vypíše text Podmínka je pravdivá, protože je pravda, že číslo 10 je větší než 2. Pokud si zkusíte zaměnit obě čísla (tedy – 2 > 10), vypíše se text Podmínka není pravdivá.

Pozn.: všimněte si, že se za závorkami nepíše středník!

Neúplný podmíněný příkaz **if**

Velmi často se v praxi stává, že nepotřebujeme celou funkčnost konstrukce **if**. V mnoha případech nám stačí, aby se v případě pravdivé podmínky něco vykonalo a v případě nepravdivosti nevykonalo nic. V tomto případě můžeme zcela vynechat část konstrukce **else**.

```

<?php
echo 'Toto je kontrola na pravdivost. Pokud bude výsledkem nepravda, nic se
nevypíše.';
// Tento text se vypíše vždy. V tomto případě pouze informuje o činnosti
programu

echo '<br />';
// zde se zařídí, že se vše následující vypíše na nový řádek

if (3 > 2)
//toto je samotná podmínka, která se vyhodnocuje

echo 'Podmínka je pravdivá';
//pokud bude pravdivá, tak se tento text vypíše. Pokud ne, nic dalšího se
nestane
?>

```

V uvedeném příkladu se to, co je součástí konstrukce if (**všimněte si, že konstrukce if končí až se středníkem!**), tzn. text „Podmínka je pravdivá“ vypíše, protože podmínka byla splněna tedy je pravdivá. Pokud však tato podmínka pravdivá nebude, není zde definován příkaz else a tak se nevykoná nic. Zkuste si nyní vyzkoušet tento příklad tak, že zaměníte obě číslíce 3 a 2.

Operátory v podmínce

O operátorech bude zmínka v následující kapitole. Tady je jeden příklad:

```

<?php
$a = 10;
$b = 0;
if ($b != 0)
// Nebude-li proměnná b rovna nule, pak proved' následující výpočet
{
    $vysledek = $a / $b;
    echo('Výsledek dělení: ' . $vysledek);
}
if ($b == 0)
    echo('Nulou nelze dělit!');
?>

```

Zobrazení času s kombinovanou podmínkou:

```

<?php
$presnycas = date('H:i');
$hodina = date('H');
echo ('Na serveru je přesně: '.$presnycas.'

```

```

{
echo 'je dopoledne.';
}
else if ($hodina>12 and $hodina<18)
{
echo 'je odpoledne.';
}
else if ($hodina>18)
{
echo 'je večer.';
}
?>

```

Ukázka načtení obsahu pokud je splněna podmínka (správné heslo):

```

<form action="" method="post">
    Zadej heslo: <input type="password" name="heslo"><br />
    <input type="submit" value="Odeslat heslo" />
</form>
<?php
$heslo = $_POST['heslo'];
echo "Následující obsah se ukáže jedině pokud znáš správné heslo.";
if ($heslo == 'tajne-heslo')
    echo '<h1>Tady je tajný obsah stránky</h1>';
else
    echo '<p>Neznáš heslo, neuvidíš tajný obsah...</p>';
?>

```

Funkce die

V souvislosti s příkazem if se velmi často používá funkce die, která slouží k okamžitému skončení skriptu PHP. Před tím, než se skript PHP ukončí, se ještě naposledy vypíše nějaká zpráva, která je umístěna uvnitř funkce die:

```
die('Ukončení programu...');
```

Často se tato funkce používá jako reakce na nějakou chybu (příklady později).