

5. Proměnná v PHP

Proměnné patří mezi základní stavební kameny každého programovacího jazyka.

Definice – proměnná je název pro nějaké místo, do kterého pak můžeme dosadit libovolnou hodnotu.

Příklad – vytváříte program, ve kterém potřebujete několikrát pracovat s číslem 4000000. Jednoduchým přiřazením tohoto čísla do proměnné (např. s názvem x), již pak v samotném běhu programu používáte pouze název této proměnné:

```
<?php
$x = 4000000;
//proměnné x jsme přiřadili číslo 4000000
echo $x * 2;
//výsledkem bude číslo 8000000 – tedy 4 miliony krát dvě
?>
```

V PHP se proměnná značí symbolem **dolaru – \$** (pravý Alt a ů), za kterým následuje jméno proměnné (např. \$pokus představuje proměnnou s názvem pokus). Vkládání hodnot do proměnné se nazývá **přiřazování hodnot proměnné**.

Do proměnné samozřejmě nemusíme vždy přiřazovat pouze číslo. Může to být i libovolný text neboli řetězec:

```
<?php
$adresa = 'Palackého 125, Nová Paka, 542 12';
//vytvořili jsme proměnnou s názvem adresa, která obsahuje delší text
echo $adresa;
?>
```

Proměnné se mohou **během programu měnit**. Tedy např.:

```
<?php
$adresa = 'Palackého 125, Nová Paka, 542 12';
echo $adresa;
echo '<br />';
//vypíše adresu a zařídí, aby se vše následující vypsalo na nový řádek
$adresa = 'Sokolovská 28, Praha 8, 111 50';
echo $adresa;
echo '<br />';
//vypíše adresu a zařídí, aby se vše následující vypsalo na nový řádek
$adresa = 'Kamenická 85, Písek 5, 245 20';
echo $adresa;
echo '<br />';
//vypíše adresu a zařídí, aby se vše následující vypsalo na nový řádek
?>
```

Tento příklad vypíše jednotlivé adresy vždy na nový řádek:

Palackého 125, Nová Paka, 542 12
Sokolovská 28, Praha 8, 111 50
Kamenická 85, Písek 5, 245 20

Názvy proměnných

Každá proměnná má svůj název. Můžeme používat **písmena a také znak podtržítka** (_). Ovšem název **nesmí začínat číslicí!**

PHP je programovacím jazykem, který je tzv. „**Case sensitive**“, neboli je citlivý na velká či malá písmena. Můžete tedy používat jak malá tak velká písmena, ale pamatujte, že je v každém z nich rozdíl.

České znaky lze také používat, ale velmi důrazně se to nedoporučuje. Obecně lze říci, že můžete používat všechny znaky, které mají ASCII hodnotu od 127 do 255 (ASCII je standardizovaný způsob, který přiděluje každému znaku určité číslo – hodnotu ASCII).

Používejte pokud možno názvy, které alespoň trochu symbolizují to, co daný skript bude vykonávat nebo daná proměnná obsahuje. V programu se pak budete později lépe orientovat.

Když proměnná neexistuje

Pokud se pokusíte v PHP spustit skript s proměnnou, která neexistuje, mohou nastat dvě varianty možností:

1. buď je PHP nastaveno tak, aby vypsaló chybové hlášení (o těchto nastaveních později..)
2. je PHP nastaveno tak, že nevypisuje chybové hlášení – pak se nic nevypíše

```
<?php
$pokus=1;
//vytvořili jsme proměnnou s názvem pokus, která obsahuje číslo 1
echo $pkus;
//zde je napsáno místo pokus pkus, tedy takováto proměnná neexistuje
?>
```

Pokud je PHP nastaveno na vypisování chybových hlášení v tomto případě vypíše:

Notice: Undefined variable: c in **C:\wamp\www\index.php** on line **4**

(cesta je samozřejmě dle daného umístění souboru)

Zrušení proměnné

Pokud jednou proměnné přiřadíte určitou hodnotu, bude tato proměnná existovat až do konce skriptu PHP. To sice nemusí, ale také může vadit. Pokud tedy potřebujeme určitou proměnnou zrušit, použijeme tzv. konstrukci **unset**, za kterou se do závorek vypíše jméno proměnné, kterou potřebujeme zrušit.

```
unset ($a)
```

Příklad:

```
<?php
$a = 1;
// Proměnná $a teď obsahuje jako hodnotu číslo 1
echo $a;
// Vypíše hodnotu proměnné $a tedy 1
unset($a);
// konstrukce unset zruší proměnnou $a. Tedy nyní již proměnná $a už
neexistuje.
echo $a;
// Nyní (pokud je tak PHP nastaveno) vypíše stránka chybu
?>
```

Proměnné uvnitř řetězce

Následující příklad demonstruje užití vytvořené **proměnné uvnitř** nějakého textu – řetězce.

```
<?php
$jmeno = 'Alík';
// Vytvoříme proměnnou $jmeno a dáme jí jako hodnotu řetězec 'Alík'
echo "Můj pes se jmenuje $jmeno";
// Vypíše: Můj pes se jmenuje Alík
?>
```

Další příklad ukazuje, jak je možné spojit proměnnou s určitým textem:

PHP je benevolentní k umístění znaku dolaru – funguje i varianta, kdy dolar není uvnitř závorek:

```
echo "Máme doma ${jmeno}a";
```

Spojování řetězců

Spojování řetězců je vlastně **spojování kusů textu**. Pro spojování textu se v PHP používá tečka (.), vložená mezi tyto řetězce. Např:

```
<?php
echo 'a'.'b'.'c';
// vypíše se abc
?>
```

Podobně můžete **spojovat proměnné**:

```
<?php
$znak = 'x';
echo $znak . $znak . $znak;
//Vypíše xxx
?>
```

Další variantou může být **připojování** nových hodnot k již vytvořené proměnné:

```
<?php
$a = '1';
// Proměnná $a obsahuje text: 1
$a .= '2';
// Proměnná $a nyní připojí text: 2
$a .= '3';
// Proměnná $a nyní připojí text: 3
echo $a;
// Nakonec se vypíše: 123
?>
```

Operace s jednotlivými znaky

Každý řetězec má určitý počet znaků. PHP si jednotlivé znaky umí očíslovat. Příklad:

```
$a = abcd;
//Pořadí znaků je tedy a=0, b=1, c=2, d=3 atd.
```

První znak má pořadí 0, druhý 1, třetí 2 atd. Tedy oproti klasickému číslování se u PHP ale i u jiných programovacích jazyků **začíná nulou**. Pokud potřebujeme nahradit daný znak z určité proměnné, vypadá to následovně:

```
<?php
$a = '12345';
$a[2] = 'x';
// znak na pozici 2 v řetězci je přepsán znakem x.
echo $a;
// Vypíše: 12x45
?>
```

Požadovaná pozice se vkládá do hranatých závorek [a] (klávesa pravý Alt+F a pravý Alt+G). Nicméně může se používat také složených závorek – { a } (klávesa pravý Alt+B a pravý Alt+N). Oba zápisy fungují stejně. Ovšem je mezinárodně dohodnuto, že verze PHP 6 bude podporovat již jen první způsob zápisu.

Pokud znak neexistuje

Pokud program požaduje vypsání znaku na pozici, kde nic není (přesněji řečeno, která neexistuje), nevypisuje se nic nebo chybové hlášení (záleží na nastavení výpisu chyb PHP serveru).

```
<?php
$a = 'ABCDE';
echo $a[5];
// Pokus o výpis znaku na pozici 5.
// Nevypíše nic nebo chybové hlášení (dle nastavení PHP)
?>
```

Připsání znaku v případě, že neexistuje znak na dané pozici

Pokud se pokusíte změnit znak na neexistující pozici, dojde k připsání tohoto znaku do řetězce:

```
<?php
$a = 'abcd';
$a[4] = 'x';
// Písmeno x se dopíše na 4, tedy v tomto případě neexistující pozici.
echo $a;
// Vypíše: abcdx
?>
```

Vyzkoušejte ještě variantu, kdy nebude daná pozice navazovat na řetězec. Např. zadáte pozici 8:

```
<?php
$a = 'abcd';
$a[8] = 'x';
// Písmeno x chceme vypsát na 8. pozici. 4. až 7. ale neexistuje.
echo $a;
// Vypíše: abcd x. Řetězec prostě vytvoří mezeru
?>
```

Určení pořadí znaku pomocí proměnné

O něco složitěji vypadá následující příklad. Namísto přímého uvedení pořadového čísla znaku je použita proměnná:

```
<?php
$a = 'ABCDE';
$poradi = 3;
echo $a[$poradi];
// Vypíše znak na pozici 3, tedy D
?>
```