



PHP tutoriál (základy PHP snadno a rychle)

Druhá, vylepšená offline verze.

Připravil Štěpán Mátl, <http://khamos.wz.cz>

Chceš se naučit základy PHP? V tom případě si prostuduj tento rychlý průvodce. Nejdříve něco málo o tomto programovacím jazyku. Dnes se nachází ve své páté verzi. Původně PHP znamenalo Personal Home Page, dnes je toto slovo zkratkou pro Hypertext Preprocessor. Jedná se o jazyk vzniklý z Perlu a nezávislý na platformě, což znamená, že ho lze používat na většině dnes existujících operačních systémech. Původně byl určen pouze pro osobní potřeby autora, avšak postupně se stal velice oblíbeným jazykem, používaným u webových aplikací a dynamicky generovaných stránek.

Téměř každý moderní web jen alespoň z části tvořen PHP skripty. PHP velmi usnadňuje práci a lze ho uplatnit u každé maličkosti. Jak se zapisuje?

```
<?php ?>

// nebo touto zkrácenou verzí (doporučuji používat tu první)
<? ?>
```

Vše, co je mezi dvěma šípkami a otazníky, je serverem považováno za PHP skript. Možná se divíte, proč tento kód není vidět ve zdrojovém kódu stránek. Je to tím, že PHP pracuje na straně serveru. Tzn., že server nejdříve zpracuje skript a poté předá prohlížeči výsledek. Ve zdrojovém kódu se tak zobrazí např. jen výsledek součtu dvou čísel. Ve výše uvedeném příkladu jsem použil rovnou i komentář. Ten slouží výhradně pro potřeby autora a pro lepší orientaci v kódu. Existují dva typy komentářů.

```
<?php
// toto je komentář jednořádkový

/* a toto
komentář
víceřádkový */
?>
```

A hurá do toho...

Nyní se již pustíme do první "aplikace". Následující kód si vlož normálně mezi tagy <body> a </body>.

```
<?php
echo "Tento text byl vypsán přes PHP!";
?>
```

Příkaz echo() slouží pro vypsání nějakého textu, proměnné nebo čehokoli jiného. Stejný výsledek vytvoří příkaz print(). Ve zdrojovém kódu (po nahrání na server, na disku ti to bez nainstalovaného Apache serveru nebude fungovat!) se objeví text Tento text byl vypsán přes PHP! Všimni si jedné věci. Každý příkaz musí být ukončen středníkem. Je to zkratka pravidlo syntaxe, bez středníků

server vypíše chybu.

Proměnné

Jako proměnná se obecně označuje ta hodnota, která není konstantní (pevně určená). Prvním znakem názvu proměnné nesmí být velké písmeno, číslo nebo jiný znak. Dalšími znaky mohou být pouze písmena, číslice a podtržítko. Proměnné začínají dolarem (\$) a zapisují se následovně.

```
<?php
$den1="sobotu";
$den2="neděli";

// takto hodnotu proměnné vypíšeme
echo "V $den1 a v $den2 se nejde do školy.";
?>
```

Výsledek: V sobotu a v neděli se nejde do školy. S proměnnými lze samozřejmě provádět spousty operací - můžeme je sčítat, násobit, odčítat, dělit, porovnávat atd. Nejčastěji je ale používáme v podmínkách.

Podmínky

```
<?php
if ($sekce=="") {
echo "Nacházíš se na úvodní stránce mého webu.";
}

if ($sekce=="download") {
echo "Nacházíš se v sekci Download.";
}

// jiný způsob zápisu podmínky
if ($sekce=="kontakt"):
echo "Zde najdeš kontakt na mne.";
endif;

/* pokud chceme zobrazit něco jen na úvodní stránce,
můžeme to udělat třeba takhle */
if ($sekce=="") {
echo "Novinky: Dnes se událo...";
}
else {
echo "Abys zjistil co je nového, musíš na úvodní stránku.";
}
?>
```

Díky podmínkám můžeš mít celý web pouze v jedné stránce. Podmínka if, else rozhoduje, podobně jako klasická podmínka if, zda platí požadované rovnosti. Pokud ano, provede se první část skriptu za if, pokud ne, provede se skript za else. Podmínky lze větvit do mnoha úrovní. Lze využít i zápis elseif, který bude pokračovat tak dlouho, dokud se něco nebude rovnat - if (podmínky) { skript }

elseif (podmínky) { skript } ... else { poslední skript }. Odkazy na sekce webu budou vypadat takto:

```
<a href="index.php?sekce=">Homepage</a>
<a href="index.php?sekce=download">Download</a>
<a href="index.php?sekce=kontakt">Kontakt</a>
```

atd.

Cykly

Ted' něco k cyklům. Ty vlastně ušetřují práci s podmínkami - zadaný skript se bude provádět tak dlouho, dokud bude platit požadovaná rovnost (nebo dokud platit nebude).

```
<?php
$i=0;
while ($i<100) {
    $i++;
    echo "$i";
}

// stejný výsledek nám dá cyklus for
for ($i=1; $i<101; $i++) {
    /* můžeme použít tuto verzi příkazu echo se závorkami,
    nebo klasickou s uvozovkami */
    echo ($i);
}
?>
```

Výše uvedený cyklus se bude provádět tak dlouho, dokud hodnota proměnné \$i nebude rovna 100. Každé proměnné musíš deklarovat nějakou výchozí hodnotu (ta naše má hodnotu 0). Při každém provedení cyklu se hodnota \$i zvětší o 1. To dělá zápis \$i++;. Při každém provedení cyklu se také hodnota \$i vypíše, takže nám vznikne řada čísel od 1 do 100. Cyklus for vypíše to samé, akorát s tím rozdílem, že čísla by se vypsala od nuly po 99. Proto má \$i výchozí hodnotu 1. Pokud by byl v prvním cyklu zápis \$i++; umístěn až za příkazem echo(), čísla by se rovněž vypisovala od nuly.

Funkce

Osobně je používám minimálně. Funkce zkracují skript, lze je např. použít pro vypsání textů s proměnnými částmi, nebo deklaraci skriptů, které lze následně hromadně provést. Nebudu se tu jimi více zabírat, ukážeme si pouze tu nejjednodušší.

```
<?php
// deklarace funkce
function prijezd($kdy) {
    echo "Přijedeme v $kdy";
}

// volání funkce
prijezd("neděli"); // výsledek: Přijedeme v neděli
prijezd("pátek"); // výsledek: Přijedeme v pátek
?>
```

Pole

Pole naopak používám docela často. Jedná se o skupinu prvků, které spolu nějak souvisí. Lze je použít například pro vytvoření kalendáře atp. Rovněž se tu o nich nebudu rozepisovat, následující příklad vysvětluje vše.

```
<?php
// vytvoření pole
$vsedni_dny=array("pondělí", "úterý", "středa", "čtvrtek",
"pátek");

// vypsání části pole
echo ($vsedni_dny[0]); // vypíše: pondělí
echo ($vsedni_dny[1]); // vypíše: úterý
?>
```

Řetězce

Ty nám umožňují pracovat s textem. Text můžeme např. rozdělit na jednotlivé věty, obrátit, vypsát prvních deset znaků, zjistit délku textu a spoustu dalších věcí. Více informací nalezneš v PHP dokumentaci (odkaz na ni níže).

```
<?php
$text="Khamos";
// zjištění počtu znaků
echo(strlen($text)); // výsledek: 6

// obrácení textu
echo(strrev($text)); // výsledek: somahK

// odstranění tagů
echo(strip_tags("Ahoj!")); // výsledek: Ahoj!

// všechna písmena velká
echo(strtoupper($text)); // výsledek: KHAMOS

// rozdělení textu
$pozdravy="Ahoj.Čau.Zdarec";
$rozdelené_pozdravy=explode(".", $pozdravy);
echo ($rozdelené_pozdravy[0]); // výsledek: Ahoj
?>
```

Načítání obsahu z jiných souborů

Pokud nechceš mít celý web v jednom souboru, jistě se setkáš s problémem, kdy je třeba vložit do více stránek stejný obsah. Například menu. K tomu v PHP slouží následující dva příkazy. Kód se vloží tam, kam chceš aby se ti menu načetlo.

```
<?php
include "menu.php";

/* nebo alternativa s tím rozdílem, že pokud
požadovaný soubor nebude nalezen, skript
se nebude provádět dál */
require "menu.php";
?>
```

A to je z mého tutoriálu vše. Dál už musíš jen zkoušet a nalézat nové a nové možnosti :). Probral jsem všechny nejdůležitější vlastnosti PHP, které jistě každý programátor mnohokrát použije. Vynechal jsem cookies, regulární výrazy, práci s formulářem a se soubory a další méně důležité věci. Vše se dá vyhledat na stránkách PHP dokumentace, kterou najdeš na www.php.net.