

Php - 1. část

Ing. Martin Dostal, Ph.D.
madostal@kiv.zcu.cz
github.com/madostal/kiv-web

Obsah

- Php - úvod a jak to funguje
 - proměnné
 - funkce
 - cykly
 - a mnoho dalšího

Php - úvod

- Interpretovaný programovací jazyk
- Určený především pro programování dynamických internetových stránek.
- Kód lze vložit přímo do HTML.
- PHP skripty jsou prováděny na straně serveru.
- K uživateli je přenášén až výsledek jejich činnosti. = např. HTML

Php

- Nezávislý na platformě, skripty fungují bez úprav na různých operačních systémech
- Rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory
- Přístup k většině databázových serverů např. přes PDO
- Podpora celé řady protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP)
- Kombinace LAMP – Linux + Apache + MySQL + PHP – často používána k tvorbě webových aplikací

Historie

- relativně nezajímavá
- viz. samostudium

Php - úvod

- skript s koncovkou .php
- index.php - spustí se po otevření adresáře nebo domény
- php kód uzavřený do **<?php ... ?>**
nebo s pomocí "short tags" **<? ?>** (nedoporučeno)
nebo "asp tags" **<% %>** (nedoporučeno)

Php - Hello world - index.php

```
<!DOCTYPE html>  
<html>  
<body>  
<?php  
    echo "Hello world!";  
?>  
</body>  
</html>
```

Jak spustit hello world

- potřebujeme webový server s podporou Php
- vhodné vyžít předkonfigurovaný balík:
 - Wamp
 - Xamp
 - atd.
- po instalaci přejdeme na URL:
 - <http://localhost/>
 - <http://localhost/phpmyadmin>
 - nástroj pro manipulaci s lokální databází

Php

- Jazyk PHP je dynamicky typový, tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty
- Pole jsou heterogenní, mohou tedy obsahovat jakékoli údaje, stejně tak jako jejich indexy
- Komentáře jako v Javě: `/* ... */` a `//...`
- `;` jako oddělovač příkazů
- Volání funkcí: `date("H:i, jS F");`
- Proměnné s dolarem `$i = 5;`

Proměnné

- Písmena, čísla, podtržítka, (dolary)
- Nesmí začínat číslicí
- Case sensitive (výjimkou jsou jména funkcí)
- Proměnná může mít stejné jméno jako funkce – nepoužívat
- Deklarace - proměnná je vytvořena, pokud jí prvně přiřadíme hodnotu

Proměnné - příklad

```
<?php
```

```
    $var = 'Bob';
```

```
    $Var = 'Joe';
```

```
    echo "$var, $Var";      // outputs "Bob, Joe"
```

```
    $4site = 'not yet';    // invalid; starts with a number
```

```
    $_4site = 'not yet';  // valid; starts with an underscore
```

```
    $täyte = 'mansikka';  // valid; 'ä' is (Extended) ASCII 228
```

```
?>
```

PHP - podpora datových typů

- Integer
- Double
- String
- Boolean
- Array
- Object
- Nedefinované proměnné mají hodnotu NULL (dat. typ NULL)
- Dat. typ resource – manipulace s db nebo třeba obrázkem

Php

- Slabě typovaný jazyk (`$a=0; $b=0.0;`)
- Typ proměnné lze změnit (`$b= "ahoj";`)
- Přetypování (`$b = (int) $a;`)

- Variabilní proměnné (pro zajímavost):
- `$a = 5;`
- `$kde_hledat = "a"; // mam se podivat do promenne $a`
- Hodnota hledané proměnné: `$$kde_hledat`
 - přes první `$kde_hledat` se dostane k "a"
 - druhý dolar ze stringu udělá proměnnou `$a`

Konstanty

- Velkými písmeny
- `define("KONSTANTA", 100);`
- vhodné např. pro přihlašovací údaje k databázi

Příklad - zpracování formuláře

Odešleme formulář s polem jmeno a jeho hodnotou "Karel"

```
<input type="text" name="jmeno" />
```

Přístup k tomuto parametru:

Zkrácený: `$jmeno` // musí být povoleno v `php.ini`, nedělat !!!

Standardní: `$_POST["jmeno"]` // doporučeno

Rozsah platnosti proměnných

- Místa ve skriptu, kde je proměnná viditelná
- Vestavěné superglobální proměnné jsou viditelné v celém skriptu
- Globální proměnné deklarované ve skriptu jsou viditelné v celém skriptu kromě vnitřku funkcí
- Proměnná použitá uvnitř funkce je lokální
- Proměnná použitá uvnitř funkce, která je deklarovaná jako globální, odkazuje na globální proměnnou
- Superglobální proměnné - `$GLOBALS`, `$_SERVER`, `$_GET`, `$_POST`, `$_COOKIE`, `$_FILES`, `$_ENV`, `$_REQUEST`, `$_SESSION`

Operátory (1)

- Aritmetické - +,-,*,/,%
- Zřetězení – \$a.\$b
- Přřazení - =
- \$b = 6 + (\$a = 5); // 11
- +=, -=, *=, /=, %=, . =
- Pre/Post-Increment/Decrement
- a=5; echo ++\$a; echo \$a++;
- Reference
- \$a=5; \$b=\$a; \$a=7; echo \$a.” “.\$b;
- \$a=5; \$b=&\$a; \$a=7; echo \$a.” “.\$b;

Operátory (2)

- Porovnávání
 - == (shodné hodnoty) x === (shodné typy)
 - !=, <>, <, > <=, >=
 - !, &&, ||, and, or (and a or má nižší prioritu)
- Bitové operátory
 - &, |, ~, ^, >>, <<
- Ternární operátor
 - Podmínka ? True hodnota : false hodnota;
- Potlačení výpisu chyby
 - \$a = @(57/0);

Funkce pro práci s proměnnými

- `string gettype(mixed var);`
- `bool settype(mixed var, string type);`
- `is_array()`, `is_double()`, `is_float()`, `is_real`, `is_long()`, `is_int()`, `is_integer()`, `is_string`, `is_object()`
- Stav proměnné
- `bool isset(mixed var);`
- `void unset(mixed var);`
- `bool empty(mixed var);`

Funkce pro práci s proměnnými

- Ekvivalent přetypování
 - `int intval(mixed var);`
 - `float doubleval(mixed var);`
 - `string strval(mixed var);`
 - `mixed` není datový typ php

Řídicí struktury - if

```
if ($cena < 1000) {  
    echo "laciné";  
}  
elseif ($cena < 10000) {  
    echo "dražší";  
}  
else {  
    echo "nejdražší";  
}
```

Řídicí struktury - switch

Switch (\$prom)

```
{  
    case 'a':  
        příkaz;  
        break;  
    default:  
        příkaz;  
        break;  
}
```

Cykly – while, for, do, foreach !!!

- while (podmínka) { příkazy; }
- for (\$d=50; \$d<=250; \$d+=50) { příkazy; }
- do { příkazy; } while (podmínka);
- foreach (\$auta as \$auto) { příkazy }
 - výpis všech prvků v poli
 - lze použít foreach (\$auta as \$key => \$auto)
 - key je klíč v poli

Řídicí struktury

- Ukončení smyčky: `break`
- Skok na další iteraci: `continue`
- Ukončení php skriptu: `exit`

Soubory, pole a funkce

Otevření souboru (1)

Otevření souboru:

- `$fw = fopen("cesta/soubor", mód);`

Root webserveru:

- `$_SERVER['DOCUMENT_ROOT'];`

Módy otevření souboru:

- `r` – read; `r+` – read + write; `w` – write; `w+` – write +read
- `a` – append; `a+` – append and read; `b` – binární mód
- Pozor na `safemod` a přístupová práva
- Při neúspěchu vráceno `false`

Otevření souboru (2)

```
$fw = @fopen("cesta/soubor", r);  
if (!$fw) {  
    echo "Nelze otevřít soubor...";  
    exit;  
}
```

Zápis do souboru, zavření souboru

- Otevření souboru pro zápis
 - `$fw = @fopen("cesta/soubor", w);`
- Zápis do souboru
 - `fwrite($fw, $retezec);`
 - `fputs($fw, $retezec);`
- Zavření souboru
 - `fclose($fw);`

Čtení ze souboru

Otevření souboru pro čtení: `$fw = fopen("cesta/soubor", r);`

Test konce souboru: `feof($fr);`

Čtení řádky:

- `fgets($fw, 999)` – max 998 bytů
- `fgetss` – podobné, vyhazuje HTML/PHP tagy
- `fgetcsv` – třetí parametr jako oddělovač

Čtení ze souboru - 2

- Čtení celého souboru
 - `readfile(soubor), $filearray = file($fr)`
- Čtení znaků
 - `fgetc($fr)`
- Čtení bytů
 - `fread($fr, délka);`

Další užitečné funkce

Test existence souboru

`file_exists(soubor)`

Zjištění velikosti souboru

`filesize(soubor)`

Vymazání souboru

`unlink(soubor)`

Další užitečné funkce - 2.

Navigace v souboru

rewind(\$fw) – na začátek souboru

ftell(\$fw) – zjištění pozice

fseek(\$fw, offset, start) – nastavení pozice v souboru

Upload souboru

Viz. cvičení - použití superglobální proměnné `$_FILES`

Klasická pole

```
$produkty = array('pneu', 'výfuk', 'sklo');
```

```
$produkty[1]; // výfuk
```

```
$cisla = range(1,10); // pole čísel 1-10
```

```
Změna prvku pole: $produkty[1] = 'karosérie'
```

```
Lze přidat nový prvek: $produkty[3] = 'karosérie'
```

```
Procházení pole: for, foreach
```

Asociativní pole !!!

Klíče místo indexů

```
$ceny = array('pneu'=>1000, 'výfuk'=>3000, 'sklo'=>2500);
```

Přístup k prvku: `$ceny['výfuk']`

Přidání nového prvku: `$ceny['karosérie'] = 10000;`

Procházení pole – `foreach`, `while`, `list`, `each`, `reset`

Vícerozměrná pole (1)

```
$produkty = array( array( 'PN', 'pneumatika', 1000),  
                  array( 'VY', 'výfuk', 3000),  
                  array( 'CS', 'čelní sklo', 5000));  
echo $produkty[0][1];
```

Vícerozměrná pole (2)

Vícerozměrné asociativní pole

```
$produkty = array( array( kod=>'PN',  
                        popis=>'pneumatika',  
                        cena=>1000),  
                  array( kod=>'VY',  
                        popis=>'výfuk',  
                        cena=> 3000),  
                  array( kod=>'CS',
```

Řazení polí

```
$produkty = array('pneu', 'výfuk', 'sklo');  
sort($produkty);
```

Case sensitive, velká písmena před malými

Asociativní pole

Podle hodnoty: `asort($ceny);`

Podle klíče: `ksort($ceny);`

Reverzní řazení: `rsort()`, `arsort()`, `krsort()`

Řazení definované uživatelem

```
usort(pole, funkce);
```

Pole - další funkce

Náhodné zamíchání pole: `shuffle(pole);`

Zásobníkové vkládání a vybírání: `array_push`, `array_pop`

Načtení pole ze souboru:

```
$objednavky = file(soubor); // + explode
```

Zjištění velikosti pole

`count()`

`sizeof()`

`array_count_values()` – vrací pole, klíče jsou původní hodnoty, hodnoty jsou počty původních hodnot

Řetězce (1)

Odříznutí bílých znaků: `trim()`, `ltrim()`, `chop()`

Tisk řetězců

`echo` – rozdíl mezi “...” a ‘...’

`printf` – navíc formátování

`sprintf` – vrací zformátovaný řetězec

Převody znaků: `strtoupper()`, `strtolower()`

Přidání escape znaků při vkládání do db:

`AddSlashes()`, `StripSlashes()`

Rozdělování/slučování řetězců: `explode()`, `implode()/join()`, `strtok()`

Řetězce (2)

Podřetězce

substr(řetězec, start, délka)

Záporná čísla znamenají od konce

Porovnávání

==

strcmp(\$ret1, \$ret2) – vrací <0, 0, >0

Délka řetězce: strlen(\$ret)

Hledání podřetězce v řetězci: strstr(\$retezec, \$hledany);

Pozice v řetězci: strpos(\$retezec, \$hledany)

Nahrazení: str_replace

Require a include (1)

Oba umožňují vložení kódu z jiného souboru

Kód se parsuje – `<?php ... ?>`

Rozdíly

Starší verze – `include` proveden tolikrát kolikrát je vložen a `require` jednou (nově `require_once`, `include_once`)

Při neexistenci způsobuje `include` warning, ale `require` fatal error => `require` použít u životně důležitých funkcí

Soubory vložené pomocí `include` mohou vracet hodnotu

Soubor vložený `require` je načten ikdyž je v podmínce a ta není splněná

Proměnné dostupné v hlavním i ve vloženém souboru

Require a include (2)

Hlavní soubor:

```
<?php require('header.php'); ?>
```

Obsah

```
<?php require('footer.php'); ?>
```

header.php

Začíná <html>

logo, menu atd.

footer.php

Patička – např. copyright

Končí </html>

Require a include (3)

openfile.php:

```
<?php
    @ $fp = fopen($name, $mode);
    if (!$fp) {
        echo "Nelze...";
        return 0;
    }
    else { return 1; }
?>
```

main.php:

```
<?php
$name = 'file.txt';
$mode = 'r';
$result = include('openfile.php');
if ($result == 1) {
    ... lze použít $fp...
}
?>
```

Funkce

Jména funkce nejsou case sensitive

Konvence – lower case

Volání: `fname(parametry);`

Deklarace: `function fname() {...}`

Funkce lze deklarovat odděleně + `require`

Uvnitř funkce lze ukončit php kód a psát html

Jména funkcí

Lze: `jmeno()`, `jmeno2()`, `jmeno_tri()`, `_jmenoctyri()`

Nelze: `5jmeno()`, `jmeno-sest()`, `fopen()`

Parametry

Povinné: `function create_table($ data) {...}`

Volitelné: `function create_table($table, $border=1) {...}`

Rozsah platnosti lokálních proměnných je pouze uvnitř funkce

Globální proměnné nejsou viditelné uvnitř funkcí (lze pomocí `global $var;`)

Předávání parametrů

- Hodnotou: `function fname($par) {}`
- Odkazem: `function fname(&$par) {}`

Vracení hodnot:

`return 5;`

Zdroje

- <http://www.w3schools.com/php/>
- <http://www.jakpsatweb.cz/>
- <http://www.vrana.cz/>
 - část PHP triky <http://php.vrana.cz/>