PHP

PHP: Hypertext Preprocessor
Personal Home Page

WEB 2013-2014

PHP – obecně (1)

- Skriptovací programovací jazyk
- Určený především pro programování dynamických internetových stránek
- Kód lze začlenit do HTML
- PHP skripty jsou prováděny na straně serveru
- K uživateli je přenášen až výsledek jejich činnosti
- Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java)

PHP – obecně (2)

- 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 (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL)
- Podpora celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP)
- Kombinace – Apache+MySQL+PHP – často používána k tvorbě webových aplikací
- Od verze 5 OOP podobné Javě

PHP - historie

- 1994 – Rasmus Lerdorf – pro osobní potřebu, vydána sada skriptů - *Personal Home Page Tools*
- Stále ještě v 1994 – PHP spojeno s programem Form Interpreter stejného autora - PHP/FI 2.0
- 1997 - Zeev Suraski a Andi Gutmans – přepsali parser – základ PHP3
- Současně byl název změněn na *PHP Hypertext Preprocessor*
- 1998 – PHP3 – rychlejší, více funkcí, běželo i pod WIN
- 2000 – PHP4
- 2004 – PHP5 – vylepšené OOP podobné Javě

Ukázka kódu

```
<table>
    <tr>
        <th>Číslo</td>
        <th>Druhá mocnina</td>
    </tr>

<?php
    $i=1;
    while ($i<=10)
    {
       echo "<tr><td>".$i."</td><td>".$i*$i."</td></tr>\n";
       $i++;
    }
?>

</table>
```

Vlastnosti jazyka 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
- Koncovka souborů: .php
- PHP kód uzavřený do <?php ... ?>, mimo HTML kód
- Komentáře jako v Javě: /* … */ a //… (#)
- ; jako oddělovač příkazů
- Volání funkcí: date('H:i, jS F');
- Proměnné s dolarem `$i = 5;`

Styly PHP značek

- `<?php echo '<p>Text</p>'; ?>`
  - Preferovaný styl
  - Nelze zakázat na straně serveru
  - Může být použit v XML dokumentech
- `<? echo '<p>Text</p>'; ?>`
  - Podle SGML instrukcí
  - Lze zakázat na straně serveru (*short_tag*)
- `<script language="php">`
  `echo '<p>Text</p>';`
  `</script>`
  - Jako JavaScript
  - Pokud HTML editor má problémy s jinými styly PHP značek
- `<% echo '<p>Text</p>'; %>`
  - Lze zakázat na straně serveru (*asp_tags*)

Přístup k proměnným formuláře

- Odešleme formulář s polem *jmeno* a jeho hodnotou *'Karel'*
- Přístup k tomuto parametru:
  - Zkrácený: `$jmeno`
    - Musí být povoleno na straně serveru
    - Problémy s laděním
  - Střední: `$_POST['jmeno']`
    - Doporučovaný
    - Od PHP v. 4.1.0
  - Dlouhý: `$HTTP_POST_VARS['jmeno']`
    - Zastaralý

Proměnné

- Identifikátory:
  - 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

Datové typy (1)

- PHP podporuje následující datové typy
  - Integer
  - Double
  - String
  - Boolean
  - Array
  - Object
- Nedefinované proměnné mají hodnotu NULL (dat. typ NULL)
- Dat. typ *resource* – manipulace s db

Datové typy (2)

- Slabě typovaný jazyk (`a=0; b=0.0;`)
- Typ proměnné lze změnit (`b='ahoj';`)
- Přetypování (`b = (double) a;`)
- Variabilní proměnné:

```
$varname = 'prom';
$$varname = 5
<==>
$prom = 5
```

Konstanty

- Velkými písmeny
- `define('KONSTANTA', 100);`
- `phpinfo();` - php konstanty

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`
- Přiřazení - `=`
  - `$b = 6 + ($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
  - `$a = @(57/0);`
- Operátor spuštění procesu
  - `$out = `dir c:`;`

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()`
- Stavu proměnné
  - `bool isset(mixed var);`
  - `void unset(mixed var);`
  - `bool empty(mixed var);`
- 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;
   case ‘b’:
      příkaz;
      break;
   default:
      příkaz;
      break;
}
```

Řídicí struktury – while, for, do

```
while (podmínka)
{
   příkazy;
}

for ($d=50; $d<=250; $d+=50)
{
   příkazy;
}

do
{
   příkazy;
} while (podmínka);
```

Řídicí struktury

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