Aplikační vrstva

Úvod do Php

Ing. Martin Dostal

Co to je PHP?

- php soubory se nekompilují, interpret je spouští přímo bez překladu
- php běží na serveru
- php soubor je .txt soubor obsahující php kód:
  - Zkrácený zápis – musí být povoleno na serveru
    - <?
      // php kód
      ?>
  - Úplný zápis
    - <?php
      // php kód
      ?>

PHP – co je potřeba?

- Webový server s nainstalovým PHP interpretem
- Nejvhodnější je použití balíčků, které obsahují webový server, nakonfigurované Php a MySql:
  - EasyPhp: http://easyphp.org/
  - Wamp: http://www.wampserver.com/en/

- Po startu webového serveru se k obsahu dostaneme přes: http://localhost/

Php – příklad hello_world.php

```
<html>
<body>

<?php
   echo "Hello World";
   $a = 5;
   $b = “a”;
   $$b = 3;

   echo “hodnota a je: $a <br/>”;

?>

</body>
</html>
```

Začátek php skriptu

proměnná

výpis

Php – definice konstant

- Proměnné začínají znakem $
  - Př. $auto, $a, $b …
- Konstanty se píší velkými písmeny a definují se s využitím příkazu „define“
  - define(“COMPANY”, “Acme Enterprises”);
  - define(“YELLOW”, “#FFFF00”);
  - define(“PI”, 3.14);
- V kódu je můžeme použít takto:
  - echo “Firma: ”.COMPANY;

Php – zpracování formuláře

- Data z formuláře můžeme získat přes proměnnou: $_POST – obsahuje všechna data z odeslaného formuláře

- Viz. příklad formular.php

- Php umožňuje snadnou práci s asociativními poli:

```
$pole = array();
$pole[“clovek”] = array(“jmeno”=>”martin”,
                        “prijmeni”=>”dostal”);
print_r($pole);
```

PHP – výhody a nevýhody

Výhody

- Snadné a relativně rychlé na programování
- Nic se nepřekládá, jen se soubory nakopírují na webový server a hotovo
- Velké množství tutoriálů a knihoven na webu
- Vhodné pro malé a střední webové aplikace
- Nemusí se deklarovat proměnné a jejich datový typ – proměnné pouze použijeme, přetypování probíhá automaticky

Nevýhody

- Pomalejší než jsp
- Zátěž serveru stoupá lineárně s počtem připojených uživatelů
- Přibližná hranice je kolem 10 000 unikátních IP adres na 1 server za 1 den
- U proměnných se nestanovuje datový typ – není zde tedy žádná kontrola, zda tam neukládám něco jiného

Php – manipulace se SESSION

- Spuštění session: session_start();
- Ukončení session – např. odhlášení:
  - $_SESSION = array();
  - Session_destroy();
- Do session lze uložit např. informace o přihlášeném uživateli
- Uživatel má identifikátor session uložený většinou v cookies

Šablony pro Php

- Šablony nám usnadňují oddělení vzhledu stránky od obsahu
- Lze využít různé existující nástroje – např. Smarty

- Přístup k proměnným ze šablony:

```
{$promenna}
//klasická proměnná

{$promenna[3]}
//4. prvek pole ($promenna[3])

{$promenna.klic}
//prvek pole s indexem klic ($promenna['klic'])
```

Php - Smarty

- Manipulace s proměnnými: http://smarty.ronnieweb.net/promenne.php
- Volání funkcí: {include file='header.tpl}
- První písmeno textu převede na velké: {$promenna|capitalize:true}
- Funkce, která zjistí počet znaků: {$promenna|count_characters:true}
- Zjištění počtu odstavců: {$promenna|count_paragraphs}
- Výpis pole:
  - **PHP:**
  - <?php
    $pole = array('prvni','druhy',
    'treti','ctvrty');
    $smarty->assign('pole',$pole);
    ?>

Smarty:

```
<body>
{foreach from=$pole key=klic item=polozka}
    <p>Polozka cislo {$klic} ma hodnotu
        {$polozka}</p>
{/foreach}
</body>
```

  - **Výstup:**
    Polozka cislo 0 ma hodnotu prvni
    Polozka cislo 1 ma hodnotu druhy
    Polozka cislo 2 ma hodnotu treti
    Polozka cislo 3 ma hodnotu ctvrty

Php frameworky

- Php frameworky jsou nástroje umožňující rychlejší a kvalitnější vývoj webové aplikace.
- Většinou doporučují a kontrolují metodiku vývoje.
- Obsahují různé kontrolní nástroje.
- Automaticky ošetřují např. vstupní data a znemožňují různé typy útoků na webovou aplikaci.

- Příklady:
  - Nette – český a jednoduchý, doporučuji
  - Symfony
  - …

Tipy a triky pro vývoj webových aplikací

Vývoj – tipy a triky

1. Již v návrhu aplikace řešíme oddělení aplikační logiky od vzhledu.
2. Aplikaci vyvíjíme iterativně = nejdříve napíšu základ a vyzkouším. Až následně vylepšuji.
3. Používáme prototypy. Pokud něco nejde vyřešit, chceme to zjistit co nejdříve.
4. Nejdříve řešíme největší problémy – i odděleně a až následně kódujeme jednodušší věci. Pokud se klíčovou vlastnost aplikace nepodaří vyřešit, ostatní věci jsou k ničemu.