PHP

Zdroje: inspirováno
http://www.w3schools.com/php

PHP

- PHP: Hypertext Preprocessor
- Interpretovaný jazyk určený především pro webové serverové aplikace, který má i samostatný interpret
- Velmi dobrá podpora pro spolupráci s ostatními službami a standardy
  - databáze, mail, grafika, obrázky, pdf, …
- Multiplatformní – běží na Windows, Linuxu, Androidu
- Procedurální i objektový přístup k tvorbě kódu

- Dynamicky typovaný jazyk
  - jednodušší deklarace proměnných
  - není nutné se starat o nastavení typu proměnné
- Aktuální verze 8.x
  - plně 64 bitová podpora (celá čísla, velké soubory)
  - nové operátory
    - spaceship $a <==> &b (0 - a=b, -1 – a<b, 1 – a>b)
    - coalescence $a ?? $b (a pokud ex. a není null, jinak b)
- Základní odkaz - www.php.net
- Dokumentace
  - https://www.php.net/manual/en/langref.php

Způsoby užití

- Návrh ve vývojovém prostředí (IDE)
  - NetBeans
  - http://phptester.net/
- Vždy nutný PHP parser (interpreter)
- Server-side scripting pro web
  - hlavní užití
  - ve webserveru přes CGI nebo serverový modul
- Samostatný překladač (CLI)
  - úroveň příkazové řádky

Základy

- Case sensitive
- Proměnné začínají na $
- Datové typy
  - String, Integer, Float (double), Boolean, Array, Object, NULL (null)
- http://www.w3schools.com/php/php_variables.asp

Prvky jazyka

- Operátory
  - http://www.w3schools.com/php/php_operators.asp
- Cykly
  - http://www.w3schools.com/php/php_looping.asp
- Funkce
  - http://www.w3schools.com/php/php_functions.asp
- Pole, asociativní pole (foreach), multidimenzionální pole
  - http://www.w3schools.com/php/showphp.asp?filename=demo_array_assoc
  - http://www.w3schools.com/php/showphp.asp?filename=demo_array_multi

Rozšíření jazyka

- Hlavní síla PHP
  - echo phpinfo();
- PECL – PHP Extension Community Library
  - http://pecl.php.net/packages.php
  - web services
  - SOAP, XML-RPC
  - PDF, matematika, grafy,
  - další

Základy

- Způsob zápisu
  - středník za každým příkazem
  - sekvenci příkazů tvořících celek (blok) uzavírat do { a }

```
<?php        // označení začátku programu v PHP
     echo “Hello world!“;          // program
?>           // nepovinné ukončení
```

- Komentáře
  - řádkový - //
  - víceřádkový - /* … */

- Každý příkaz na novou řádku
- Příklad

```
<?php
        // toto je řádkový komentář
        /*
        echo “Hello world!“.“ – “. “Ahoj svete!”;
        */
?>
```

Operátory jazyka

- Umožňují provádět operace s určitým typem dat
  - unární, binární
  - základní úkony, které lze s daným typem dat provádět
- Aritmetické

  +, -, *, /, %, **, ++, --
- Přiřazovací

  =, +=, -=, *=, /=, %=
- Textové

  ., .=

- Logické

  &&, and, ||, or, xor, !

- Porovnávací

  ==, === (typová kontrola), !=, <>, >=, <=, >, <

- Pro práci s polem

  +, ==, ===, !=, <>, !==

Proměnné

- Místo v paměti obsahující data určitého typu
  - v PHP sám překladač rozhoduje o typu obsahu
- Syntaxe
  - vždy začínají znakem $
  - nejlépe používat písmena bez diakritiky
  - v názvech nepoužívat mezery
- Názvy by měly vyjadřovat, co je obsahem
  - $x, $násobitel, $delitel, $parametr 1, $par_1
- Deklarace a inicializace proměnné
  - její první výskyt a první přiřazení hodnoty

- Naplnění proměnné (přiřazení hodnoty do proměnné)

```
$pozdrav = “Hello world!“;
$cislo = 2*3;
```

- Výpis proměnné
  - příkaz echo, print()

```
<?php
        $pozdrav=“Hello world“;
        echo $pozdrav.“!“;
        print($pozdrav.“!“);
?>
```

- Pokud je proměnná deklarována ve funkci, tak má lokální platnost

```
$x=2; // globální proměnná

function myTest() {
    $x = 5; // lokální proměnná
    echo $x;
}

myTest();
echo $x;
```

Datové typy

- String - $a=“text”;
- Integer - $a=25; +- 2 mld
- Float - $a=3.14; $a=3.5E7; //desetinná tečka
- Boolean - $a=true/false;
- Array (pole) - $a=array(“Jarda",“Pepa",“Tonda");
- Object – mimo kurz
- Null - $a=null; // prázdná hodnota

Řetězce

- Počet znaků v řetězci

```
strlen("Hello world!"); // 12
```

- Počet slov v textu

```
str_word_count("Hello world!"); // 2
```

- Nalezení textu v řetězci

```
strpos("Hello world!", "world"); // 6
```

- Náhrada textu

```
str_replace("world", "Dolly", "Hello world!"); // Hello Dolly!
```

Podmínka

- Větvení programu dle hodnoty podmínky
- Několik forem syntaxe příkazu IF
  - ```
    if ($a < 0) {
        echo „Záporné číslo!";
    }
    ```
  - ```
    if ($a < 0) {
        echo „Záporné číslo!";
    } else {
        echo „Nezáporné číslo!";
    }
    ```

```
if ($a < 0) {
    echo „Záporné číslo!";
} elseif ($a > 0) {
    echo „Kladné číslo!";
} else {
    echo „Nula!";
}
```

- Switch – více cest větvení

```
$auto = “Skoda";
switch ($auto) {
    case “Skoda":
        echo “OK!";
        break;
    case “Ford":
        echo “Nic moc!";
        break;
    default:
        echo “Co mas vlastne za auto?";
}
```

Cyklus s pevným počtem opakování

- Cyklus typu FOR
  - známe počet opakování

```
for ($x = 0; $x <= 10; $x++) {
    echo „Krok: $x \n";
}
```

- Cyklus FOREACH – speciální varianta FOR
  - pro procházení kolekcí – ty typicky indexovány od 0
  - vytvoření náhradní proměnné pro prvek kolekce

```
$auta = array("Skoda", "Ford", "BMW", “Kia");
foreach ($auta as $prvek) {
    echo "$prvek \n";
}
```

Cyklus s podmínkou

- Neznáme počet opakování
  - je nutno v každém kroku testovat podmínku ukončení
- Podmínka na začátku

```
$x = 1;
while($x <= 5) {
    echo "Krok: $x \n";
    $x++;
}
```

- Podmínka na konci

proběhne aspoň jednou

```
$x = 1;
do {
    echo „Krok: $x \n";
    $x++;
} while ($x <= 5);
```

Funkce

- Opakovatelně použitelné části programu
  - použití parametrizace
  - názvy nejsou case sensitive

```
function pozdrav() { // definice funkce bez parametrů
    echo „Nazdar!";
}
pozdrav(); // volání funkce
```

- Použití parametrů zvyšuje použitelnost funkce

```
function pozdrav($text) { // definice funkce s parametrem
    echo $text."!";
}
pozdrav(“Ahoj”); // volání funkce
```

- Možné definovat návratovou hodnotu
  - tu lze pak uložit do proměnné nebo přímo použít

```
function soucet($x, $y) {
    $z = $x + $y;
    return $z;
}
$a = 10;
$b = soucet(5,$a);
echo soucet(10, 20) . "\n";
```

- Zadání přednastavené (default) hodnoty par.

```
function soucet($x=100, $y=200) { …
echo soucet();
```

Pole

- Kolekce z více prvků i různého typu (v PHP)
  - prvky přístupné přes index pole – vždy od 0!

```
$auta = array("Skoda", "Ford", "BMW", "Kia");
echo $auta[2];
$auta[4]=10; // nelze ve všech jazycích
```

- Délka pole

```
count($auta);
```

- Pole polí

```
$auta = array(array("Skoda“,10),array("Ford",20),
array("BMW",30));
echo $auta[2][1];
```

Asociativní pole

- Pole, kde indexy nejsou čísla od 0, ale texty
  - dvojice klíč => hodnota

```
$vek = array("Petr"=>"35", "Jan"=>"37", "Pepa"=>"43");
echo $vek["Jan“];
```

- Průchod asociativním polem

```
foreach($vek as $klic => $hodnota) {
    echo $klic ." = " . $hodnota."\n";
}
```

Řazení polí

- Podle hodnoty
  - od nejmenší - sort($auta);
  - od největší - rsort($auta);
- Asociativní pole
  - podle klíče - ksort($vek);
  - podle klíče reverzně - krsort($vek);
  - podle hodnoty - asort($vek);
  - podle hodnoty reverzně - arsort($vek);

Datum a čas

- Princip jako v Excelu
  - timestamp – časové razítko od 1.1.1970
- Příkaz date
  - lze použít pro datum i čas, vše dáno modifikátory formátu
  - echo "Dnes je " . date("d.m.Y") . "\n";
  - echo "Je právě " . date("H:i:s") . "\n";
  - $c=date("H")+10;
- Výčet modifikátorů větší
  - https://secure.php.net/manual/en/function.date.php

Vstup a výstup

- Klávesnice – načtení vstupu od uživatele
  - ve Windows problematické
  - příkaz stream_get_line(STDIN, 1024, "\n");
- Monitor
  - příkazy echo, print, print_r, var_dump