03 Konstrukt 1

PVA4 Programování a vývoj aplikací

Adam Fišer

Obsah

Základní syntaxe

- `<?php` ... obsah ... `?>`
- výjimečně se lze setkat s `<?` ... obsah ... `?>`, ale není zaručena plná kompatibilita

```
1  <?php
2
3    // PHP Kód
4
5  ?>
```

- Středník `;` je nejdůležitější částí PHP skriptu. Ukončuje deklaraci funkce a příkazu.
- Nejčastější chyba v zdrojovém kódu = chybí středník

Komentáře

- Jednořádkový komentář: `//`
- Víceřádkový komentář: `/* ... */`

```php
<?php

  // Jednořádkový komentář

  /*
   * Víceřádkový
   * komentář
   */

  /*
    Víceřádkový
    komentář
  */

?>
```

Dokumentační komnetáře `Optional`

- Dokumentační komentáře jsou speciální komentáře, které se používají k dokumentaci kódu
- Dokumentační komentáře začínají `/**` a končí `*/`
- Dokumentační komentáře mohou obsahovat speciální značky, které jsou následovány `@`

```php
<?php

  /**
   * Toto je dokumentační komentář
   *
   * @param string $param1
   * @return string
   */
```

Dokumentační komentáře

Ukázka dokumentačního komentáře v kódu pro Doctrrine ORM

```php
/**
 * Operation - poplatek
 * @var Operation
 * @ORM\ManyToOne(targetEntity="Operation")
 * @ORM\JoinColumn(referencedColumnName="id", nullable=true)
 */
private ?Operation $operation;

// ...

/**
 * @author  Adam Fišer <adam.fiser@wanex.cz>
 * @licence MIT
 * @param int $id
 */
function getTrain(int $id): Train
{
    //some code
    return $Train;
}
```

Konvence

- PHP je case-sensitive jazyk
- Názvy **proměnných** a funkcí by měly být psány v **camelCase**
- Názvy **tříd** by měly být psány v **PascalCase**
- Názvy **konstant** by měly být psány **velkými písmeny** s podtržítky

Význam	camelCase	CamelCase	PascalCase	CONSTANT_CASE
Je viditelný	isVisible	IsVisible	IsVisible	IS_VISIBLE
Počet pokusů	numberOfAttempts	NumberOfAttempts	NumberOfAttempts	NUMBER_OF_ATTEMPTS
Datum vytvoření	createdAt	CreatedAt	CreatedAt	CREATED_AT

Proměnné

Proměnné

- Proměnné v PHP začínají znakem `$` a následuje název proměnné
- (Znak `$` lze zapsat Pravý ALT + Ů)
- Proměnné mohou obsahovat písmena, číslice a podtržítko
- Proměnné nesmí začínat číslem
- Proměnné jsou case-sensitive - rozlišují se velikost písmen
- Proměnné by měly být pojmenovány tak, aby bylo jasné, co obsahují
- Přiřazení hodnoty pomocí znaku rovnítko `=`
- Proměnné se ukončují středníkem `;`
- Mohou být přetypovány
- Mohou být deklarovány bez hodnoty

Deklarace proměnné

```php
<?php

$string = "Textový řetězec";
$stringInt = '14'; // stále textový řetězec
$integer = 14;
$boolean = TRUE;
$array = ['pole','jiné pole'];

$uvozovky = "Textový výstup";
$apostrof = 'Textový výstup';

$vystupViceRadku = "Dlouhý text\\ndruhý řádek";

?>
```

Rozsah platnosti proměnných

- Globální proměnné jsou dostupné v celém skriptu
- Lokální proměnné jsou dostupné pouze v rámci funkce, ve které byly deklarovány
- Statické proměnné si pamatují svou hodnotu i po skončení funkce
- Proměnné v PHP jsou dynamicky typované
- Některé předdefinované proměnné v PHP jsou "superglobální", což znamená, že jsou vždy přístupné bez ohledu na rozsah - a můžete k nim přistupovat z libovolné funkce, třídy nebo souboru, aniž byste museli dělat cokoli speciálního.

`$GLOBALS`, `$_SERVER`, `$_GET`, `$_POST`, `$_FILES`, `$_COOKIE`, `$_SESSION`, `$_REQUEST`, `$_ENV`

Operátory

- Operátory jsou použity k provedení operací na proměnných a hodnotách
- PHP obsahuje mnoho operátorů, jako jsou aritmetické, relační, logické, inkrementační/dekrementační, řetězcové, pole, porovnávací, identické, ternární, atd.

Operátory

Matematické

- Přičítání: `$a + $b`
- Odečítání: `$a - $b`
- Násobení: `$a * $b`
- Dělení: `$a / $b`
- Modulo: `$a % $b`
- Exponent: `$a ** $b`

Porovnávací

- Rovná se: `$a == $b`
- Nerovná se: `$a != $b`
- Rovná se a je stejného typu: `$a === $b` (3x rovnítko)
- Nerovná se nebo není stejného typu: `$a !== $b`
- Větší než: `$a > $b`
- Menší než: `$a < $b`
- Větší nebo rovno: `$a >= $b`
- Menší nebo rovno: `$a <= $b`
- Porovnání: `$a <=> $b`

Uvozovky

V PHP můžete použít jednoduché nebo dvojité uvozovky k definici řetězců.

- V jednoduchých uvozovkách se proměnné a escape sekvence nepoužívají.
- Řetězce s jednoduchými uvozovkami: nezpracovává speciální znaky uvnitř uvozovek.
- V dvojitých uvozovkách můžete použít proměnné a escape sekvence pro zpracování speciálních znaků.

```php
<?php
  $name = "Adam";
  echo "Jmenuji se $name"; // Jmenuji se Adam
  echo 'Jmenuji se $name'; // Jmenuji se $name
?>
```

Escapování

- Escapování je proces, kdy se speciální znaky převedou na znaky, které lze v řetězci zobrazit
- Používá se zpětné lomítko `\`
- `\'` Apostrof
- `\"` Uvozovky
- `\\` Zpětné lomítko
- `\?` Otazník
- `\&` Ampersand

```
1   echo "<a href=\"https://dest.url\">My Link</a>";
2   echo '<a href="https://dest.url">My Link</a>';
```

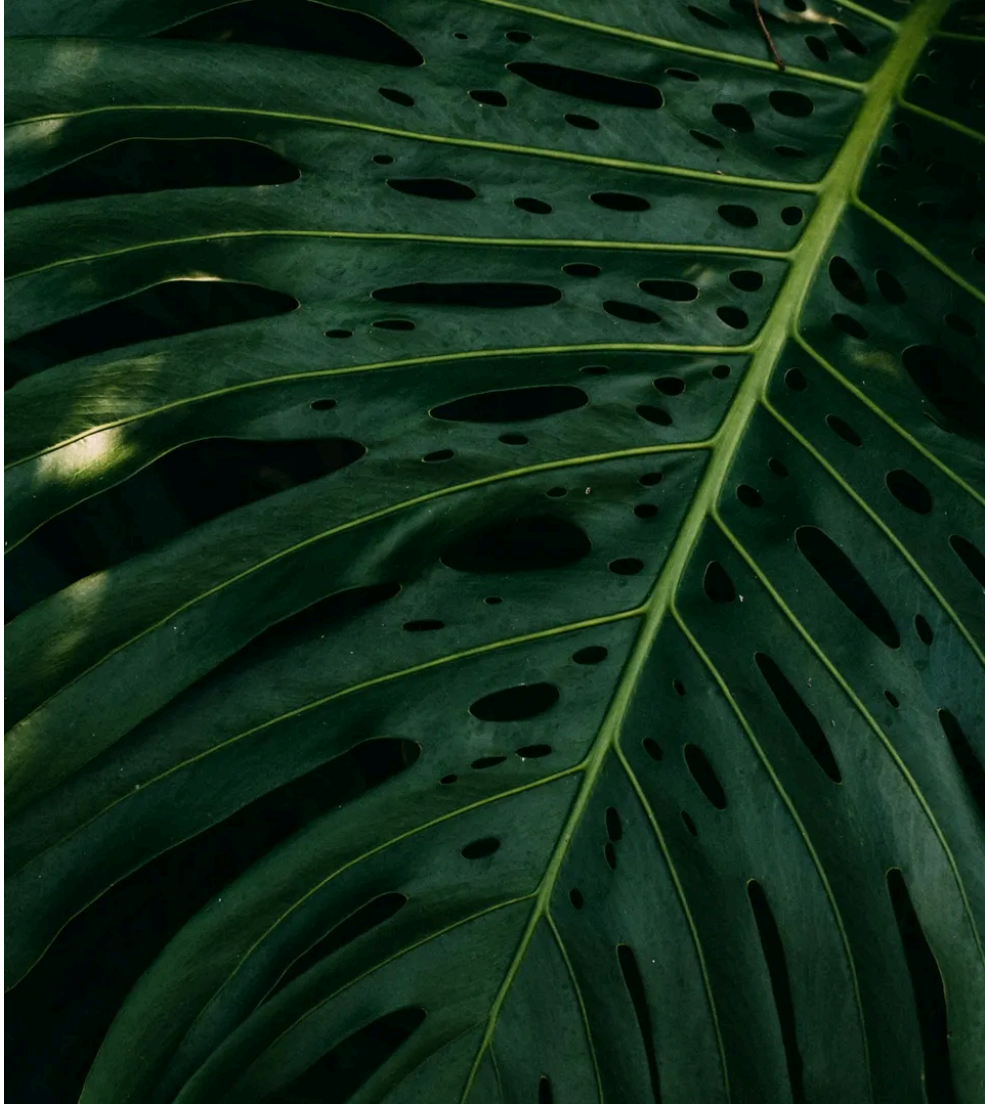
Keywords

- PHP má několik klíčových slov, která nelze použít jako názvy proměnných, funkcí nebo tříd
- Klíčová slova jsou rezervována pro speciální účely a nemohou být použita jinak

```
abstract    and    array    as    break    callable    case    catch    class    clone    const    continue
declare    default    die    do    echo    else    elseif    empty    enddeclare    endfor    endforeach
endif    endswitch    endwhile    eval    exit    extends    final    finally    for    foreach    function
global    goto    if    implements    include    include_once    instanceof    insteadof    interface    isset
list    namespace    new    or    print    private    protected    public    require    require_once    return
static    switch    throw    trait    try    unset    use    var    while    xor    yield
```

Výstupy

- Výstupy v PHP se provádí pomocí příkazů `echo`, `print` a `var_dump`
- používají se k zobrazení informací na obrazovce
- Výstupy mohou být textové, číselné, logické, pole, objekt, atd.



Příkaz `echo`

- Příkaz `echo` vypíše jeden nebo více výrazů bez dalších nových řádků nebo mezer.
- Nevrací návratovou hodnotu
- Může přijímat více parametrů – reálně se nepoužívá

```
1   echo "Hello World!";
```

Příkaz `echo`

```php
// Deklarace proměnné s hodnotou 1$
$dolar="1$";

// Výstupy
echo $dolar; // 1$
echo "$dolar"; // 1$
echo '$dolar'; // $dolar
echo "mám jen" . $dolar . "<br />"; // mám 1$<br />
echo "nemám ani $dolar"; // nemám ani 1$
echo($dolar); // $1
echo "výstup","více","parametrů"; // výstupvíceparametrů
```

Příkaz `print`

- Příkaz `print` vypíše jeden výraz
- Vrací hodnotu 1, což znamená, že může být použit jako výraz
- Může přijímat pouze jeden parametr
- Má nižší prioritu než `echo`
- Pomalejší než echo

```
1   print "Hello World!";
```

Příkaz `var_dump`

- Příkaz `var_dump` vypíše informace o proměnné nebo výrazu
- Vrací datový typ a hodnotu proměnné
- Nepoužívá se pro běžný výpis, ale pro účely testování a při vývoji

```php
<?php
  $x = 5;
  $y = 10.5;
  $z = "Hello World!";
  var_dump($x); // int(5)
  var_dump($y); // float(10.5)
  var_dump($z); // string(12) "Hello World!"
?>
```

Děkuji za pozornost

Otázky?

Repository / Prezentace