



PHP - základy

Ing. Galuszková Ivana 2020

PHP (Personal Home Page)

- » skriptovací jazyk na straně serveru
 - » opensource, multiplatformní
 - » komunikace s SQL databází
- **dynamické stránky** (reagují na tlačítka a myš, zpracovávají data zadávaná uživatelem, zobrazované informace se mění apod.)
- klientské
 - serverové



Jak to funguje?

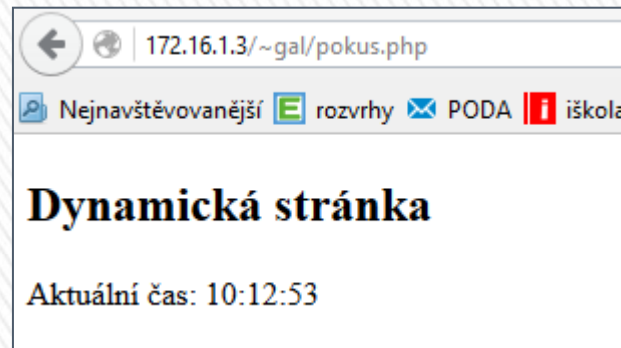
- » PHP kód se zapisuje v textovém tvaru
- » PHP interpret (běžícím na webovém serveru) kód překládá a posílá do klienta (prohlížeče) hotový html kód
- » nezatěžuje vlastní počítač – náročný na komunikaci se serverem
- » zrychluje načítání stránek
- » vyšší ochrana zdrojových kódů
- » nenáročnost na HW a SW klienta



Dynamická stránka

```
<!DOCTYPE html>
<html lang='cs'>
  <head>
    <title></title>
    <meta charset='utf-8'>
  </head>
  <body>
    <h2>Dynamická stránka</h2>
    <p>
      Aktuální čas:
      <?php
        echo date("h:m:s");
      ?>
    </p>
  </body>
</html>
```

původní php kód v editoru



stránka v prohlížeči



php kód v prohlížeči →

```
1 <!DOCTYPE html>
2 <html lang='cs'>
3   <head>
4     <title></title>
5     <meta charset='utf-8'>
6   </head>
7   <body>
8     <h2>Dynamická stránka</h2>
9     <p>
10      Aktuální čas:
11      10:12:53    </p>
12   </body>
13 </html>
```


Co potřebujeme

- » textový editor pro psaní kódu (pspad ...)
- » webový server (apache)
- » databáze SQL

Práce na serveru: Soubory ukládáme do public_html na uživatelském účtu (nastavit vhodná práva i pro nadřazenou složku)

Prohlížeč - URL (automaticky najede do složky public_html):

http://172.16.1.4/~x1a_prijmeni

Pro lokální použití: XAMPP – soubory ukládáme do složky
xampp/htdocs/mojeslozka

Prohlížeč - URL:

<http://localhost/mojeslozka>



Vložení kódu php do html

Způsoby vložení php scriptu

```
<?php  
// PHP kód  
?>
```

```
<SCRIPT LANGUAGE="php"> php  
// PHP kód  
</SCRIPT>
```



Syntaxe jazyka PHP

- » všechny příkazy píšeme do skriptu označeného php značkami
- » soubor s příponou .php
- » blok příkazů { }
- » příkazy jsou ukončeny středníkem
- » komentáře:

// jednořádkový komentář

/* víceřádkový

komentář */



Proměnné

- » **název proměnné začíná značkou \$**
- » **nedeklarují se**
- » libovolná délka
- » obsahují písmena, číslice a podtržítko
- » číslice nemůže být prvním znakem po \$
- » jsou case-sensitive
- » typ proměnné se určí, když do ní přiřadíme hodnotu, automaticky se může konvertovat na jiný typ



Datové typy

Základní typy :

integer - celé číslo

double, float, real - reálné číslo

boolean – logické hodnoty *

string - řetězec v uvozovkách nebo apostrofech

array – pole, počáteční index je 0, pokud se neurčí jinak

object – objekt

* true - nenulová hodnota výrazu

false - číslo 0, prázdný řetězec, pole, objekt a ukazatel null



Příkaz echo

» vypíše obsah proměnné, řetězec, číslo, html tag....

» možné zápisy:

echo "Vítejte v naší nabídce";

echo \$promenna;

echo "<h2>Můj nadpis</h2>";



Operátory

- » aritmetické: `+` `-` `*` `/` `%`
- » přiřazovací: `=` `+=` `-=` `*=` `/=` `%=`
- » inkrementace, dekrementace: `++` `--`
- » relační: `==` `!=` `>=` `<=` `>` `<`
- » shoda hodnoty a datového typu:
 - `===` stejná hodnota i typ
 - `!==` stejná hodnota nebo stejný typ
- » logické: `!` `&&` `||` `not` `and` `or`
- » podmíněný operátor: `podmínka?výraz1:výraz2`
- » operátor tečka `.` - spojování řetězců
- » operátor čárka `,` odděluje výrazy – v hlavičce cyklu `for`
- » bitové operátory



Větvení

```
if (výraz) { ..... }
```

```
if (výraz) { ..... } else { ..... }
```

```
if (výraz) { ..... } elseif { ..... }
```

```
else { ..... }
```

switch (výraz) ← na rozdíl od C je možné použít libovolný typ

```
{
```

```
    case hodnota1: příkazy; break;
```

```
    case hodnota2: příkazy; break;
```

```
    default: příkazy;
```

```
}
```



Smyčky

- » s podmínkou na začátku

`while(výraz) { ... }`

- » s podmínkou na konci

`do { ... } while(výraz);`

- » s parametrem

`for (inicializace; podmínka; inkrementace) { ... }`

- » pro procházení pole

`foreach ($pole as $hodnota) { ... }`

`foreach ($pole as $index => $hodnota) { ... }`



Přerušení cyklu

break – úplné ukončení nejbližšího cyklu

continue – ukončí aktuální průchod, cyklus pokračuje

break 2;

continue pocet;

- opustí několik cyklů najednou



Pole

- » prvky se automaticky číslují od 0
- » může obsahovat prvky různých datových typů

Definice pole:

`$pole = array(1, 3, 5, 7);`

`$poleA = array("Petr", "Velký", 18, 1.8);`

`$poleC[5] = 2;` - vloží na místo s indexem 5

`$poleC[] = 4;` - vloží na první volnou pozici

`$klient["platba"] = 15000;` asociativní pole (použití textu u indexu)

Funkce pro práci s polem: `count()`, `max()`, `sort()`, `implode()`, ...



Procházení celého pole po prvcích

Máme hodnoty uložené v poli \$seznam:

cyklus for: (podobně jako v jazyce C)

```
for( $i = 0; $i < count($seznam); $i++ )  
{  
    /* zpracování prvku pole: $seznam[$i] */  
}
```

cyklus foreach:

```
foreach( $seznam as $x )  
{  
    /* zpracování prvku pole: $x */  
}
```



Formulář

- » definován parametr „**action**“ (název skriptu, který zpracuje formulář)
- » definován parametr „**method**“ (post, get)
- » všechny prvky musí být identifikovány jménem
- » musí obsahovat odesílací prvek („**submit**“)

Obsah formuláře se po odeslání tlačítkem “submit” ukládá do globálního asociativního pole **\$_POST** resp. **\$_GET**



Zpracování formuláře

Data odeslané z formuláře jsou uloženy v proměnných:

» `$_POST ["jmeno_prvku_formulare"]`

– odeslané metodou post

» `$_GET ["jmeno_prvku_formulare"]`

– odeslané metodou get

Př. `<?php`

`$nazev_v_php = $_POST["nazev_ve_form"];`

`...`

`?>`



Možnosti kontroly – HTML 5

» použití nových typů pro prvek **input**

→ automatická kontrola např. pro číslo ...

```
<input type=" " .....">
```

number, url, tel, email, date, week, color, range, ...

» použití atributů pro prvky

required, min, max, step, placeholder, checked, datalist, pattern ...



Možnosti kontroly v php

Př 1. //bylo z formuláře něco odesláno?

```
if ( $_POST ) {.....}
```

Př 2. //bylo stisknuto tlačítko se jménem odeslat?

```
if ( isset($_POST["odeslat"]) ) { ..... }
```

Př 3. //bylo vyplněno město?

```
if ( $mesto == NULL ) { ..... }
```



Možnosti kontroly v php

Př 4. //ošetření zaškrtnutí prvku "checkbox"

```
if ( isset($_POST["souhlas"]) ) { $souhlas = "ano"; }  
else { $souhlas = "ne"; }
```

Př 5. //datum - lokalizace

```
date_default_timezone_set('Europe/Berlin' );  
echo date("d.m.Y H:i:s");
```

Př 6. //texty

```
$jmeno = ucfirst(strtolower($jmeno));  
// varianta mb_strtolower()
```



Další užitečné funkce

- » **ucfirst()** - změnění první písmeno na velké
- » **ucwords()** - změnění první písmeno každého slova na velké písmeno
- » **date()** - aktuální datum a čas
- » **is_numeric(\$prom)** - testuje, zda se jedná o číslo
- » **htmlentities()** – kontrola odesílaných informací
- » **htmlspecialchars()** – kontrola odesílaných informací

