

2. PHP

1

Základná syntax a štruktúry

Obsah

2

- Premenné, konštanty
- Reťazce
- Príkaz echo
- Komentáre
- Operátory
- Vetvenie výpočtu (IF)
- Cykly
- Funkcie
- Dátum a čas
- Test pre pokročilých

Premenné

3

- Musia začínať znakom **\$**
- napr. **\$pocet**
- V názve premennej môžu byť len písmená, číslice a podčiarkovník
- Hneď za znakom **\$** nesmie byť číslica
- Nedefinujeme typy, PHP si ich (akože) definuje samo (automaticky)
- `$cislo = 123;`
- `$realne = 12.3;` (desatinná čiarka = **bodka**)
- `$ret1 = "Ahoj";` `$ret2 = 'Ahoj';`

Konštanty

4

- Nezačínajú dolárom
- Názov konštanty nesmie začínať číslom
- **define("názov_konštanty", "hodnota");**
- Píšeme zväčša veľkými písmenami, ale nie je to pravidlo
- **define("POCET", 12);**

Príkaz echo

5

- Vypíše zadaný „reťazec“ (prípadne číslo) do výsledného HTML súboru – **echo čosi**;
- Namiesto reťazca môžeme dosadiť premennú alebo konštantu.
- echo **\$pocet**; // výpis premennej
- echo **POCET**; // výpis konštanty
- echo (**"text xyz"**); – reťazec môžeme dať aj do zátvoriek

Reťazce

6

- Môžu byť v apostrofoch ('**retazec1**') alebo úvodzovkách ("**retazec2**")
- **Rozdiel:**
 - v reťazci v úvodzovkách sa premenné a špeciálne znaky vyhodnotia,
 - v reťazci v apostrofoch sa všetky znaky vypíšu ako sú v reťazci.
- Spájajú sa bodkou . (zreťazenie)
Např. \$vysledok = **\$retazec1 . \$retazec2;**
- V reťazcoch **nemôžeme** vypisovať konštanty. Môžeme ich však zreťaziť.
Např. **echo "Počet záznamov je" . POCET;**

Vypisovanie reťazcov

7

- **echo "text xyz";**

Ak do reťazca vložíme premenné, **vyhodnotia sa**.

Napr. echo **"Ahoj \$meno"**; vypíše **Ahoj Jozo**.

- **echo 'text xyz';**

Ak do reťazca vložíme premenné, **NEvyhodnotia sa**, ale budú chápané ako obyčajný text.

Napr. echo **'Ahoj \$meno'**; vypíše **Ahoj \$meno**.

Komentáre

- platia iba v rámci PHP vsuvky `<?php ... ?>`
- `// ...` po koniec riadka alebo po koniec PHP vsuvky (značka `?>`)
- `/*` viacriadkové komentáre `*/`

Operátory

- Číselné
 - **+**, **-**, *****, **/** (podiel v reálnych číslach),
 - **%** (zvyšok po delení celých čísel),
 - **++** (zvýš o 1), **--** (zníž o 1), napr. **\$vysl++**; (**\$vysl = \$vysl + 1**;))
- Prirad'ovacie
 - **=**, **+=**, **-=**, ***=**, **/=**, **%=**, **.=**
 - Sú to skrátene zápisy, napr. **\$vysl += 4**; (**\$vysl = \$vysl + 4**;))
- Logické
 - **!** (negácia), **and** (zároveň), **or** (alebo), **&&** (zároveň), **||** (alebo), **xor**
- Porovnávacie
 - **==**, **!=**, **<**, **>**, **<=**, **>=**, **===**

Porovnávanie

10

- == alebo ===
- != alebo !==
- if (0 == false) – správne/nesprávne?

==	Rovnosť hodnôt
===	Rovnosť hodnôt aj typu – identita
!=	Rôznosť hodnôt
!==	Rôznosť hodnôt aj typu – identita

Vetvenie výpočtu (IF)

11

- **if** (podmienka) príkaz;
- **if** (podmienka) príkaz1;
else príkaz2;
- **if** (podmienka1) príkaz1;
elseif (podmienka2) príkaz2;
else príkaz3;
- Podmienka musí byť vždy v zátvorkách ()
- Ak je príkazov, ktoré sa majú vykonať, viac ako 1, musíme ich dať do blokov pomocou { }

- **while** (podmienka) príkaz;
ak podmienka platí, cyklus beží
- **do** príkaz; **while** (podmienka);
ak podmienka platí, cyklus beží
- **for** (vyraz1; vyraz2; vyraz3) príkaz;
vyraz1 – vykonáva sa iba na začiatku cyklu
(inicializačná hodnota/podmienka)
vyraz2 – ak je TRUE, zoznam príkazov sa vykoná
(podmienka ukončenia)
vyraz3 – príkazy sa vykonávajú na konci každého cyklu
(zvyčajne zvyšovanie počítadla)

Cykly – ukážky

13

zdrojový kód/formát

príklad

```
while (podmienka) {  
    zoznam_prikazov  
}
```

```
$i = 1;  
while ($i <= 10) {  
    echo "$i<br>";  
    $i++;  
}
```

```
do {  
    zoznam_prikazov  
} while (podmienka)
```

```
$i = 1;  
do {  
    echo "$i<br>";  
    $i++;  
} while ($i <= 10);
```

```
for (vyraz1; vyraz2; vyraz3) {  
    zoznam_prikazov;  
}
```

```
for ($i = 1; $i <= 10; $i++) {  
    echo "$i<br>";  
}
```

Existencia premenných

14

- **isset(\$premenna)** – zistí, či premenná existuje (vráti true/false)
- **unset(\$premenna)** – zruší premennú
- **if (isset(\$meno)) echo "Ahoj \$meno";**

Vlastné funkcie

15

```
function menoFunkcie(parametre) {  
  // telo funkcie  
}
```

```
function mocnina3($cislo) {  
  return $cislo * $cislo * $cislo;  
}
```

Ak funkcia nemá parametre, treba uviesť prázdne zátvorky **()**, napr. **function vypis()** { }

Globálne premenné

16

Veľký rozdiel oproti Delphi, Lazarus

```
→ $sucet = 0;  
function pridaj($co) {  
    $sucet += $co;    // $sucet = $sucet + $co;  
}
```

Hodnota premennej **\$sucet** sa nezmení!

Globálne premenné – správne

17

```
$sucet = 0;  
function pridaj($co) {  
    global $sucet;  
    $sucet += $co;    // $sucet = $sucet + $co;  
}
```

Parametre – inicializačná hodnota

18

```
$sucet = 0;  
function zvys($co = 1) {  
    global $sucet;  
    $sucet += $co;  
}
```

```
$sucet = 0;  
zvys(2);  
zvys();
```

Náhodné čísla, dátum a čas

19

- **rand**(*[min, max]*) – Vrátí náhodné číslo z intervalu **0** a **RAND_MAX**, alebo z intervalu **min** – **max**
- **checkdate**(**mesiac**, **den**, **rok**) – vráti TRUE, ak je zadaný dátum platný
- **date**("format" [, **timestamp**]) – vráti reťazec času a dátumu podľa zadaného formátu

príkaz	vypíše sa
<code>date("j.n.Y")</code>	21.2.2013
<code>date("j. F Y, h:i A")</code>	21. February 2013, 8:30 AM
<code>date("G:i:s")</code>	8:30:15

Test pre pokročilých (1)

20

- V prípade, že ho študent splní aspoň na 80%, získa výslednú známku **A** a nemusí absolvovať cvičenia ani skúšku. V prípade, že neuspěje, absolvuje kurz v štandardnom režime.
- Z každej oblasti (PHP, MySQL) musí byť zrejmé, že študent danej problematike rozumie.

Test pre pokročilých (2)

21

- Štvrtok 21.2.2013, H6 (**paralelne s cvičeniami**)
- 8:10 – 9:40, 9:50 – 11:20, 11:30 – 13:00
- Používať môžete rôzne príručky, knihy (len tlačené) ...
- K dispozícii budú oficiálne manuály PHP a MySQL
- Dobrovoľná účasť
- Treba sa vopred (nezáväzne) registrovať cez Moodle (do 20.2. do 20:00)

Test pre pokročilých (3)

22

- Minimálne 2 stránky – budú mať nejakú spoločnú časť (napr. navigácia, pätička...)
- Dizajn netreba riešiť, bude kompletne vyriešený a k dispozícii (HTML, CSS)
- Formulár (napr. či sú zadané povinné položky, či sú položky daného formátu, vypočítať cenu, atď.)
- Autentifikácia pomocou SESSIONS/COOKIES
- Práca s údajmi v tabuľke (SELECT, INSERT, UPDATE, DELETE)
- Práca s databázou pomocou phpMyAdmin, väčšina tabuliek (štruktúra, údaje) bude k dispozícii

Ďakujem za pozornosť 😊

23

Priestor na vaše otázky