Programování v PHP

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

Syntaxe 1. část
BI-PHP

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Strucny prehled zakladni syntaxe PHP - 1. cast</title>
    </head>
    <body>
        <?php

            // komentare
            // do konce radku
            /* od - do */
            #  do konce radku

            // promenne se nedeklaruji; typ urcen prave prirazenou hodnotou

            $i = 5;

            $i = "ABC";

            $jmeno = "Franta";

            // echo - vypis dat do vystupu (do vytvarene webovske stranky)

            echo "Ahoj<br>";

            echo "Jmenuje se $jmeno\nDalsi<br>"; // nahradit promennou jeji
                                                 // hodnotou (tez escape
                                                 // sekvence)

            echo 'Jmenuje se $jmeno\n<br>';      // vypsat retezec doslovne

           $i = 2; $j = 3;
           echo "Soucin = ".$i*$j."<br>";  // vyraz nelze psat do retezce


            // misto echo lze tez pouzit funkci print (minimalni rozdily)

            // pro formatovany vystup slouzi funkce printf, napr.:
            $cena = 2.5;
            printf("Cena=%5.2f",$cena); // vypise Cena=2.50 (dve des. mista)

            // seznam formatovacich znaku - help PHP, funkce sprintf
            // sprintf - jako printf, ale vystup jde do stringu

            // test existence promennych
            echo "Je nastaveno:--".isset($jmeno)."--<br>";

            // isset - neexistujici promenna, nebo promenna s hodnotou null
            //         vraci false

            echo "Je nastaveno:--".isset($nejni)."--<br>";
            // pro false nezobrazi nic, pro true 1

            unset($jmeno);  // zrusi promennou
```

```
$jmeno = "Eva";

// definice dlouheho textu (HEREDOC)
// ohranicujici (ukoncovaci) retezec je uveden mezi <<< a Enter
echo <<<MUJRETEZEC
<p>Toto je velice, velice, velice dlouhy text,
ktery pisi na dalsi radek; Jmeno=$jmeno</p>
MUJRETEZEC;
// ukoncovaci retezec MUSI byt na samostatnem radku bez mezer,
// tabelatoru apod; Maximalne jej muze nasledovat strednik!!

// definice dlouheho textu (NOWDOC)
// jako HEREDOC, ale uvodni ohranicujici retezec je v apostrofech
// <<<'MUJRETEZEC' ... (u HEREDOC mohou byt uvozovky)
// NOWDOC neprovadi nahrazovani promennych ... (jako 'ab$citacX')

// Misto HEREDOC a NOWDOC lze ovsem pouzit viceradkovy text,
// ohraniceny " nebo ' pouze na 1. a poslednim radku (bude ale
// obsahovat whitespaces (mezery, nove radky, ...)


// typy promennych
$i = 1;                 // cele cislo
$r = 3.1415;            // realne cislo
$r = 2.8e3;

$retezec = "Nazev";  // retezec

$pravda = true;
$pravda = $i > 0;

// pole - viz dale
// objekty - viz dale

// resources - obvykle zdroje OS (napr. otevreny soubor,
                // otevrene connection k databazi, ...)

// definice konstant
define("PI",3.141592);     // definice
$delka = 2*PI*10;          // pouziti

// operatory

// aritmeticke operatory + - * / % (modulo - zbytek po deleni)

echo (3+"4")."<br>"; // automaticka konverze

// slozene operatory += -= *= ...
$k = 2;
$k += 3;      // $k = $k + 3; // totez

// operatory incrementace ++ a dekrementace --
$i = 5;
$k = 3 + $i++;    // postincrement - NAPRED pouzije, POTOM zvysi;
                  // ++$i - preincrement - opacne
echo "\$i = $i; \$k = $k<br>";
```

```
// relacni operatory < <= > >= != (nerovno) == (rovno),
// === (identicka rovnost) !== (identicka nerovnost)

// bezna chyba:
if ($i = 1){          // chtel $i == 1
    // ...            // provede vzdy!!
};

$b = "1" == 1;   // je rovno (provede konverzi)
$b = "1" === 1;  // neni IDENTICKY rovno

// operator prirazeni - vracenou hodnotou je prirazovana hodnota
$i = 1;

$i = $j = $k = 5;     // operator je asociativni zprava - priradi
                      // hodnotu 5 do vsech promennych

$i = ($j = ($k = 5));// takto provede


// prirazeni hodnotou
$j = 1;
$i = $j; // zkopiruje data
$j++;    // zmeni pouze $j

echo "\$i = $i; \$j = $j<br>";

// prirazeni odkazem (referenci)
$j = 1;
$i =& $j;        // priradi referenci (adresu) promenne $j;
                 // ukazuje na stejne misto v pameti
// $i = &$j;     // totez

$i++;            // zvysi i $i - jde o odkaz
echo "\$i = $i; \$j = $j<br>";

// pouziti vice znaku $
$nazev = "name";
$name = "Karel";
echo $$nazev."<br>";    // Karel ($$nazev zpracovava zprava)

// operatory logicke && i and; || i or; ! i not
// and, or not - nizsi priorita!! (viz help PHP, operatory)

$b = ($i > 5) and ($j > 1);

// operatory bitove & - bitove AND, | - bitove OR, ^ - bitove XOR,
// ~ - bitove NOT
$i = 3;          // ..0 0011
$j = 5;          // ..0 0101
$k = $i & $j;    // ..0 0001

// operatory bitovych posunu >> vpravo  a << vlevo
$k = 4;          // ...0 0100;
$i = $k << 2;    // .001 0000;   // posun o 2 bity vlevo

// operatory pretypovani (casting)
$i = "123";
$k = (int)$i;    // prevod na integer; (int) - operator castingu
```

```
            // ternalni (podmineny) operator
            $k = ($i > $j)?$i:$j;

            // neprehanet!!
            $k += ($i-- > ++$j)?--$i - $j++:$i-- + ++$j;

            // automaticke konverze (tez explicitni)
            $i = (int)"ABC";      // vrati nulu!!
            echo "$i<br>";
            $i = (int)"123ABC";   // vrati 123!!
            $i = (int)"ABC4";     // vrati nulu!!

            // konverze specialnich hodnot na cisla:
            // true -> 1, false -> 0, null -> 0, resource - > cislo (jeho ID)

            // konverze na logicke hodnoty:
            // false: 0, 0.0, "" - prazdny retezec, false,
            //        pole bez prvku (prazdne); objekt a resource vzdy true!!


            // funkce empty - vraci true, lze-li jeji argument (promenna!!) --
            // interpretovat jako false, nebo kdyz dana promenna neexistuje

            $k = "";
            $i = empty($k);
            echo "--$i--<br>";
        ?>
    </body>
</html>
```