

Web programiranje

Vežbe 4 - Funkcije

Funkcije

Funkcije se definišu koristeći ključnu reč **function** slično kao u nekim drugim programskim jezicima, recimo C-u ili Javi, kao:

```
function ime_funkcije(parametri)
{
    telo funkcije
    return povratna_vrednost;
}
```

Za razliku od Jave, parametri funkcije se mogu prenositi i po vrednosti i po referenci. Primer prenošenja parametara po vrednosti je:

```
<?php
function puna_cena($cena, $porez) {
    $total = $cena + $cena*$porez/100;
    return $total;
}

function obim_kruga($poluprecnik) {
    return 2*$poluprecnik*pi();
}

echo "Puna cena je ". puna_cena(100, 18);
echo "<br />Obim prvog kruga je ".sprintf("%.3f",obim_kruga(6));
echo "<br />Obim drugog kruga je ".sprintf("%.3f",obim_kruga(7));
?>
```

Zadatak: Napisati funkciju *promeni_case(\$tekst, \$kod)* koja u zavisnosti od koda koji može biti "V" ili "M", menja tekst u velika slova ili mala slova, respektivno.

```
<?php

function promeni_case($str, $kod='') {
    switch(strtoupper($kod)) {
        case 'V':
            return strtoupper($str);
            break;
        case 'M':
            return strtolower($str);
            break;
        default:
            return $str;
            break;
    }
}
```

```
print promeni_case("Prirodno-matematicki fakultet Kragujevac<br/>", "M");
print promeni_case("Prirodno-matematicki fakultet Kragujevac<br/>", "V");
print promeni_case("Prirodno-matematicki fakultet Kragujevac<br/>");
?>
```

U gornjem primeru vidi se i kako se u PHP-u tretiraju podrazumevane vrednosti argumenata. Kao i u programskom jeziku Java, svi argumenti desno od prvog podrazumevanog argumenta takođe moraju imati podrazumevane vrednosti.

Zadatak:

Napisati funkciju koja računa prosečnu vrednost niza brojeva.

```
<?php
function average($numbers){
// inicijalizacija
$sum = 0;

//broj elemenata u nizu
$size = count($numbers);

// iteriraj kroz niz i saberi brojeve
for ($i = 0; $i < $size; $i++){
$sum += $numbers[$i];
}

$prosek = $sum/$size;
return $prosek;
}

$mean = average(array(96, 93, 97));
echo $mean;
?>
```

Funkcija može da vrati i niz, a ne samo jednu vrednost. Sledeći primer to demonstrira.

Zadatak: Korisnik unosi 8 e-mail adresa u 8 tekst polja unutar HTML forme. Napisati funkciju koja daje listu jedinstvenih domena izvučenih iz polja forme.

```
<html>
<head>
<title>Adrese i domeni</title>
</head>
<body>
<?php
// Definicija funkcije
function jedinstveni_domeni($lista) {
    $domeni = array();
    foreach ($lista as $adresa) {
        $arr = explode("@", $adresa);
        if (isset ($arr[1])) $domeni[] = trim($arr[1]);
    }
    return array_unique($domeni);
}
```

```
// Ako forma jos nije poslata
if (!isset ($_POST['posalji']))
{
    echo "<form method='POST' action=\".$_SERVER['PHP_SELF'].\">";
    for ($i=1; $i<=8; $i++)
        echo "Adresa $i: <input type='text' name='adrese[' . $i . ']' /> <br/>";
    echo "<input type='submit' name='posalji' value='Posalji' />";
    echo "</form>";
}
// Obrada poslatih vrednosti
else
{
    $adrese = $_POST['adrese'];
    if ( is_array($adrese) )
    {
        foreach (jedinstveni_domeni($adrese) as $domen) {
            echo "<p>$domen</p>";
        }
    }
}
?>
</body>
</html>
```

Prosleđivanje parametara po referenci

U PHP-u je moguće proslediti parametar po referenci i na taj način ga učiniti mutabilnim. Mehanizam je sličan navođenju ključne reči VAR ispred parametra u programskom jeziku Pascal. Znak koji PHP koristi za referencu je &:

```
<?php
function zameni(&$a, &$b) {
    $temp = $a;
    $a = $b;
    $b = $temp;
}

$a = "Mika";
$b = "Pera";
echo "Pravi redosled: $a i $b.<br />";
zameni($a, $b);
echo "Obrnuti redosled: $a i $b.<br />";
?>
```

Drugi primer radi sortiranje niza korišćenjem jednostavnog algoritma *Bubble sort*, pri čemu se niz sortira “u mestu”. Pravo sortiranje niza naravno treba raditi ugrađenom funkcijom **sort()**, koja je višestruko brža.

```
<?php

function zameni(&$a, &$b) {
    $temp = $a;
    $a = $b;
    $b = $temp;
}

function sortiraj(&$niz) {
    for($i=0; $i<count($niz)-1; $i++)
        for($j=$i+1; $j<count($niz); $j++)
            if ($niz[$i] > $niz[$j]) zameni($niz[$i], $niz[$j]);
}

$niz = array(1,5,1,7,4,6,8,2,4,1);
echo "Nesortirani niz je: ".implode(' ', $niz)."<br />";
sortiraj($niz);
echo "Sortirani niz je: ".implode(' ', $niz)."<br />";
?>
```

Pomoćne funkcije za rad sa argumentima

Skript jezici kao što je PHP omogućavaju i neke dodatne funkcionalnosti u odnosu na jezike sa statičkim tipiziranjem. Jedna od mogućnosti je da se funkcija definiše bez argumenata, a da joj se pošalje proizvoljno mnogo:

```
<?php
// definicija funkcije
function neka_funkcija() {
    // uzmi broj argumenata
    $numArgs = func_num_args();

    // uzmi argumente
    $args = func_get_args();

    // stampaj argumente
    print "Poslati su mi sledeci argumenti: ";
    for ($x = 0; $x < $numArgs; $x++) {
        print "<br />Argument $x: ";
        // ako je poslat niz, iterisi kroz njega i stampaj sadržaj
        if (is_array($args[$x])) {
            print " NIZ ";
            foreach ($args[$x] as $index => $element) {
                print " $index => $element ";
            }
        }
        else {
            print " $args[$x] ";
        }
    }
}
```

```
// pozovi funkciju sa raznorodnim argumentima  
neka_funkcija("red", "green", "blue", array(4,5), "yellow");  
?>
```

Prosečna vrednost niza:

```
<?php  
function average(){  
    // inicijalizacija  
    $sum = 0;  
  
    //broj elemenata u nizu  
    $size = func_num_args();  
  
    // iteriraj kroz niz i saberi brojeve  
    for ($i = 0; $i < $size; $i++){  
        $sum += func_get_arg($i);  
    }  
  
    $prosek = $sum/$size;  
    // vrati srednju vrednost  
    return $prosek;  
}  
  
$mean = average(96, 93, 97);  
echo $mean;  
?>
```

Globalne promenljive

Sledeći primer demonstrira kako se iz funkcije operiše nekom globalnom promenljivom:

```
<?php  
  
$danas = "Utorak";  
  
function dan() {  
    // varijabla je globalna  
    global $danas;  
    $danas = "Subota";  
    // stampaj varijablu  
    print "Unutar fukcije je $danas. <br />";  
}  
  
// print the variable  
print "Pre poziva funkcije je $danas. <br />";  
dan();  
print "Posle poziva funkcije je $danas. <br />";  
?>
```

Ukoliko bi se izostavila linija “**global \$danas**” promenljiva **\$danas** bila bi lokalna za funkciju.