

Web programiranje

Vežbe 4 - Nizovi u jeziku PHP

Jedan od složenih PHP tipova podataka u su nizovi. U upotrebi su dve vrste nizova:

- nizovi sa numeričkim indeksiranjem
- asocijativni nizovi (na nekim platformama poznatih pod nazivom “*mape*”)

Ovde će biti obrađene obe vrste.

Nizovi sa numeričkim indeksiranjem

Definisanje niza

```
<?php
$hrana = array('sarma','cvarci','kajmak','prebranac','jagnjetina');
print_r($hrana);
?>
```

ili sa potpuno istim efektom:

```
<?php
$hrana[0]='sarma';
$hrana[1]='cvarci';
$hrana[2]='kajmak';
$hrana[3]='prebranac';
$hrana[4]='jagnjetina';
print_r($hrana);
?>
```

ili

```
<?php
$hrana[]='sarma';
$hrana[]='cvarci';
$hrana[]='kajmak';
$hrana[]='prebranac';
$hrana[]='jagnjetina';
print_r($hrana);
```

Dodavanje člana

Način 1 (navođenje indeksa):

```
$hrana[5]='gibanica';
```

Način 2 (dodavanje člana iza poslednjeg postojećeg indeksa):

```
$hrana[]='gibanica';
```

Modifikacija postojećeg člana niza

```
$hrana[3]='corbast pasulj';
```

Brisanje člana niza

```
unset($hrana[3]);
```

Brisanje celog niza

```
unset($hrana);
```

Primer:

```
$hrana = array('sarma', 'cvarci', 'kajmak');  
print_r($hrana);  
unset($hrana[1]);  
$hrana[] = 'corbast pasulj';  
print_r($hrana);
```

daje sledeći izlaz:

```
Array ( [0] => sarma [1] => cvarci [2] => kajmak )  
Array ( [0] => sarma [2] => kajmak [3] => corbast pasulj )
```

Asocijativni nizovi

Asocijativni nizovi umesto celih brojeva kao ključ koriste string:

```
$obrok = array( 'dorucak' => 'cvarci',  
              'rucak' => 'corbast pasulj',  
              'vecera' => 'pihtije' );  
print_r($obrok);
```

daje sledeći izlaz:

```
Array ( [dorucak] => cvarci [rucak] => corbast pasulj [vecera] => pihtije )
```

Potpuno isto se postiže i sledećim kodom:

```
$obrok['dorucak'] = 'cvarci';  
$obrok['rucak'] = 'corbast pasulj';  
$obrok['vecera'] = 'pihtije';  
print_r($obrok);
```

Iz prethodnog koda se vidi i kako se vrši dodavanje novih asocijativnih članova. Brisanje i modifikacija se kod asocijativnog niza rade na potpuno isti način kao i kod niza sa numeričkim indeksom:

```
unset($hrana['rucak']); // Brisanje  
$hrana['vecera'] = 'leskovacka muckalica'; // Modifikacija
```

Neke funkcije za rad sa nizovima

Broj elemenata niza, bio on običan ili asocijativan, dobija se funkcijom **count(\$niz)**, koja ima i alias **sizeof(\$niz)**.

Dodavanje i brisanje na početku i na kraju niza

Funkcija **array_push** dodaje jedan ili više elemenata na kraj niza:

```
int array_push ( array &$amp;array, mixed $var [, mixed $... ] )
```

```
<?php

```

Funkcija **array_pop()** uzima element sa kraja niza:

```
mixed array_pop ( array &$amp;array )
```

```
<?php

```

štampa:

```
kajmak
Array ( [0] => sarma [1] => cvarci )
```

Pored dodavanja elemenata na kraj niza i skidanja elemenata sa kraja niza, postoji mogućnost da se isto to učini i na početku niza, i to pomoću funkcija **array_shift** i **array_unshift**, koje imaju sledeći prototip:

```
mixed array_shift ( array &$amp;array )
int array_unshift ( array &$amp;array, mixed $var [, mixed $... ] )
```

Funkcije **array_push()** i **array_unshift()** rade ne rade sa asocijativnim nizovima, već samo sa običnim. Za proširivanje asocijativnog niza treba koristiti uobičajenu sintaksu: **\$niz[\$ključ] = \$vrednost**.

Konverzija string->niz i niz->string

Funkcija **explode** je vrlo korisna kada treba razdvojiti elemente nekog stringa razgraničene bilo kojim separatorom i smestiti ih u niz:

```
// defnisi CSV string (Comma Separated Values)
$boje = 'plava, zelena, crvena, bela, crna';
// podela na individualne reci
$boje = explode(' ', $boje);
print_r($boje);
```

U gornjem primeru je očigledno da je promenljiva **\$boje** u liniji 4 redefinisana, što je sasvim moguće u jezicima sa dinamičkim tipiziranjem, kao što je PHP.

Funkcija **implode** izvršava obrnutu radnju, tj. spaja elemente nekog niza u string čiji su delovi povezani navedenim separatorom:

Sortiranje nizova

Funkcije za sortiranje nizova su:

```
bool sort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
bool rsort ( array &$array [, int $sort_flags = SORT_REGULAR ] )
```

Funkcija `sort` sortira članove niza u direktnom, a `rsort` u reverznom redosledu. Parametar **\$sort_flags** može imati vrednosti: `SORT_REGULAR` - compare items normally (don't change types), `SORT_NUMERIC` - compare items numerically, `SORT_STRING` - compare items as strings, `SORT_LOCALE_STRING` - compare items as strings, based on the current locale.

```
// defnisi CSV string (Comma Separated Values)
$boje = array('plava', 'zelena', 'crvena', 'bela', 'crna');
// vraca niz sortiran abecednim redom
sort($boje);
print_r($boje);
print "<br />";
// vraca niz sortiran reverznim poretkom
rsort($boje);
print_r($boje);
```

izlaz:

```
Array ( [0] => bela [1] => crna [2] => crvena [3] => plava [4] => zelena )
Array ( [0] => zelena [1] => plava [2] => crvena [3] => crna [4] => bela )
```

Međutim, funkcija **sort()** prilikom sortiranja asocijativnog niza postavlja celobrojne indekse, što uglavnom nije ono što se želi kada se govori o sortiranju asocijativnog niza:

```
<html>
  <head></head>
  <body>
    <ul>
      <?php
        $skracenice["IBM"]="International Business Machines";
        $skracenice["MS"]="Microsoft";
        $skracenice["CA"]="Computer Associated";
        $skracenice["WHO"]="World Health Organization";
        $skracenice["UK"]="United Kingdon";
        $skracenice["BA"]="Something Random";

        sort($skracenice);

        foreach($skracenice as $kljuc => $vrednost) {
          echo "<li>$kljuc = $vrednost </li>";
        }
      ?>
    </ul>
  </body>
</html>
```

izlaz iz skripte bi bio sledeći:

```
0 = Computer Associated
1 = International Business Machines
2 = Microsoft
3 = Something Random
4 = United Kingdom
5 = World Health Organization
```

Kao rešenje ovog problema, tj. kako bi se očuvale vrednosti ključeva, dovoljno je **sort()** funkciju zameniti funkcijom **asort()**, pri čemu tako izmenjen gornji primer daje sledeći izlaz:

```
CA = Computer Associated
IBM = International Business Machines
MS = Microsoft
BA = Something Random
UK = United Kingdom
WHO = World Health Organization
```

Ukoliko se želi sortiranje po ključevima, a ne po vrednostima, koristi se funkcija **ksort()**.

Iterisanje kroz niz (foreach petlja)

U poslednjoj skripti primećuje se još jedan izraz kontrole toka koji umnogome olakšava rad sa nizovima, bili oni sa celobrojnim indeksima ili asocijativni. Sintaksa za niz sa celobrojnim indeksima je:

```
foreach ($niz as $clan) {
    // radi nesto sa $clan
}
```

dok za asocijativni niz sintaksa izgleda ovako:

```
foreach ($niz as $kljuc => $vrednost) {
    // radi nesto sa $kljuc i/ili $vrednost
}
```

kao u gornjem primeru.

Vrlo lako se mogu dobiti i posebni nizovi za ključeve i vrednosti jednog asocijativnog niza. To se postiže funkcijama `array_keys()` i `array_values()`:

```
<?php
// definisi niz
$obrok = array( 'dorucak' => 'cvarci', 'rucak' => 'corbast pasulj', 'vecera' => 'pihtije' );

// vraca array ('dorucak', 'rucak', 'vecera') sa celobrojnim indeksima
$resultat = array_keys($obrok);
print_r($resultat);
print "<br />";

// vraca array ('cvarci', 'corbast pasulj', 'pihtije') sa celobrojnim indeksima
$resultat = array_values($obrok);
print_r($resultat);
?>
```

Korišćenje nizova u obradi formi

Nizovi su veoma korisni i u obradi formi, i to posebno onih sa poljima za označavanje (*checkboxlist*) ili listama sa višestukom selekcijom. Niz se koristi kako bi “uhvatio” sve selektovane vrednosti. Sledeći primer to ilustruje:

```
<html>
  <head></head>
  <body>

    <?php
    // proverava submit-a
    if (!isset($_POST['submit'])) {
    // prikazi formu
    ?>

    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">
      <input type="checkbox" name="muzicar[]" value="Bon Jovi">Bon Jovi
      <input type="checkbox" name="muzicar[]" value="KUD Idijoti">KUD Idijoti
      <input type="checkbox" name="muzicar[]" value="Nervozni Postar">Nervozni postar
      <input type="checkbox" name="muzicar[]" value="MJ">MJ
      <input type="checkbox" name="muzicar[]" value="Jethro Tull">Jethro Tull
      <input type="checkbox" name="muzicar[]" value="Rokeri s Moravu">Rokeri s Moravu
      <input type="submit" name="submit" value="Izaberi">
    </form>

    <?php
    }
    else {
    // ili prikazi listu izabranih muzicara
    // foreach petlja ispisuje elemente
    if (is_array($_POST['muzicar'])) {
      echo 'Izabrano je: <br />';
      foreach ($_POST['muzicar'] as $a) {
        echo "<i>$a</i><br />";
      }
    }
    else {
      echo 'Nista nije izabrano.';
    }
  }
  ?>

</body>
</html>
```

Zadaci za vežbanje

1. Modifikovati primer sa prošlog termina koji pravi tabelu čije se dimenzije unose preko forme, tako da svaka parna vrsta ima sivu, a neparna belu pozadinu.
2. Napisati funkciju koja radi kao unapređena verzija funkcije implode, tj. sve stringove osim poslednjeg odvoja zapektama, a poslednji veznikom "i". Na primer:

`array("slanina", "kupus", "paprika")` prelazi u `"slanina, kupus i paprika"`

Rešenja zadataka

Prvi zadatak:

```
<html>
  <head>
    <style type="text/css">
      .parna-vrsta {
        background-color:silver;
      }
      .neparna-vrsta {
        background-color:white;
      }
    </style>
  </head>
  <body>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
      Uneti broj vrsta <input name="br_vrsta" type="text" size="4">
      i kolona <input name="br_kolona" type="text" size="4">
      <input type="submit" name="posalji" value="Nacrtaj tabelu">
    </form>

    <?php
    $stilovi = array('parna-vrsta', 'neparna-vrsta');

    if (isset($_POST['posalji'])) {
      echo "<table width = 90% border='1'>";
      // setovanje varijabli sa ulaza
      $br_vrsta = $_POST['br_vrsta'];
      $br_kolona = $_POST['br_kolona'];
      // petlja za pravljenje vrsta
      for ($m = 1; $m <= $br_vrsta; $m++) {
        $stil = $stilovi[$m % 2];
        echo "<tr class='$stil'>";
        // petlja koja pravi kolone
        for ($n = 1; $n <= $br_kolona; $n++) {
          echo "<td>A($m, $n)</td> ";
        }
        echo "</tr> ";
      }
      echo "</table> ";
    }
  </body>
</html>
```


Drugi zadatak:

```
<?php
function unapredjeni_implode($niz) {
    switch (count($niz)) {
        case 0:
            return '';
            break;
        case 1:
            return $niz[0];
            break;
        case 2:
            return $niz[0].' i '.$niz[1];
            break;
        default:
            // izbaci zadnji clan niza i dodeli promenljivoj $zadnji
            $zadnji = array_pop($niz);
            return implode(', ', $niz).' i '.$zadnji;
            break;
    }
}

$boje = array ('crvena','zelena','plava','crna');
echo unapredjeni_implode($boje);
?>
```