

# Osnove PHP-a

Nizovi



# PHP nizovi

Ako imate iskustva s drugim programskim jezicima ili strukturama podataka općenito, možda ste upoznati s dvije strukture podataka koje su vrlo česte i korisne:

- liste
  - poredani skup elemenata
- mape
  - skup elemenata identificiranih ključevima

U PHP-u ne postoje liste i mape, postoje nizovi. Niz je struktura podataka koja implementira obje, listu i mapu.

# PHP nizovi – inicijalizacija

Postoje različite opcije za inicijalizaciju niza. Možete inicijalizirati prazan niz ili možete inicijalizirati niz s podacima. Postoje i različiti načini pisanja istih podataka u nizove.

**\$empty1** i **\$empty2** dva su načina kreiranja praznog niza.

Kasnije ćete vidjeti da se listama rukuje kao mapama. Interno je niz **\$names1** i **\$names2** mapa, a ključevi su poredani brojevi.

Ključevi niza mogu biti bilo koje alfanumeričke vrijednosti, poput slova, znakova ili brojeva. Vrijednosti niza mogu biti bilo što: slova, znakovi, brojevi, logičke vrijednosti, drugi nizovi i slično.

```
1  <?php
2      // Inicijalizacija praznog niza
3      $empty1 = [];
4      $empty2 = array();
5      // Inicijalizacija niza (lista) s podacima
6      $names1 = ['Harry', 'Ron', 'Hermione'];
7      $names2 = array('Harry', 'Ron', 'Hermione');
8      // Inicijalizacija niza (mapa) s podacima
9      $status1 = [
10         'name' => 'James Potter',
11         'status' => 'dead'
12     ];
13     $status2 = array(
14         'name' => 'James Potter',
15         'status' => 'dead'
16     );
17  ?>
```

PHPOsnove/Primjeri/Nizovi/inicijalizacija\_niza.php

# PHP nizovi – višedimenzionalni

Ovaj niz je mapa koja sadrži dva niza (mape). Svaka mapa sadrži različite vrijednosti kao što su tekstualni podaci, decimalne vrijednosti i logičke vrijednosti.

```
1  <?php
2  // Višedimenzionalni niz
3  $books = [
4      '1984' => [
5          'author' => 'George Orwell',
6          'finished' => true,
7          'rate' => 9.5
8      ],
9      'Romeo and Juliet' => [
10         'author' => 'William Shakespeare',
11         'finished' => false
12     ]
13 ];
14 ?>
```

PHPOsnove/Primjeri/Nizovi/višedimenzionalni\_niz.php

# PHP nizovi – populacija

Nizovi nisu nepromjenjivi, odnosno mogu se mijenjati nakon inicijalizacije.

Sadržaj niza možete promijeniti neovisno o tome tretirate li ga kao mapu ili kao listu.

Ako niz tretirate kao listu, dodajete novi element na kraj niza.

Ako niz tretirate kao mapu, potrebno je odrediti ključ koji želite nadjačati.

```
1  <?php
2  // Niz kao lista
3  $names = ['Harry', 'Ron', 'Hermione'];
4  // Niza kao mapa
5  $status = [
6      'name' => 'James Potter',
7      'status' => 'dead',
8      'age' => 24
9  ];
10 // Dodavanje novog elementa na kraj niza
11 $names[] = 'Neville';
12 // Nadjačavanje ključa u nizu
13 $status['age'] = 32;
14
15 print_r($names, $status);
16 ?>
```

PHPOsnove/Primjeri/Nizovi/populacija\_niza.php

# PHP nizovi – uklanjanje elemenata

Ako trebate ukloniti element iz niza, umjesto dodavanja ili ažuriranja, možete upotrijebiti funkciju **unset**.

Novi niz **\$status** sadrži samo ključ **name**.

```
1  <?php
2  // Inicijalizacija niza
3  $status = [
4      'name' => 'James Potter',
5      'status' => 'dead'
6  ];
7  // Uklanjanje elementa iz niza
8  unset($status['status']);
9
10 print_r ($status);
11 ?>
```

PHPOsnove/Primjeri/Nizovi/uklanjanje\_elementa\_niza.php

# PHP nizovi – pristup elementima

Pristup elementima niza je jednostavan i vrši se preko ključeva.

Kod nizova-mapa to je jednostavnije zato što su ključevi unaprijed definirani.

Kod nizova-lista prvo morate razumjeti kako liste funkcioniraju.

Već znate da se liste interno tretiraju kao mape s numeričkim ključevima. Prvi ključ je uvijek **0**; Dakle, niz s **n** elemenata imat će ključeve od **0** do **n-1**.

Pokušaj pristupa ključu koji ne postoji u nizu vratit će vrijednost **null** i ispisati obavijest da u kodu radite nešto pogrešno.

```
1 <?php
2 // Inicijalizacija niza kao lista
3 $names = ['Harry', 'Ron', 'Hermione'];
4 // Pristup elementu
5 print_r($names[1]);
6 // Pristup elementu koji ne postoji
7 print_r($names[4]);
8
9 // Inicijalizacija niza kao mapa
10 $status = [
11     'name' => 'James Potter',
12     'status' => 'dead',
13     'age' => 24
14 ];
15 // Pristup elementu
16 print_r($status['name']);
17 // Pristup elementu koji ne postoji
18 print_r($status['birthday']);
19 ?>
```

PHPOsnove/Primjeri/Nizovi/pristup\_elementima\_niza.php

# PHP nizovi – *empty* i *isset* funkcije

**Empty** i **isset** su korisne funkcije za ispitivanje sadržaja niza.

Ako želite saznati sadrži li niz elemente ili je prazan, to možete učiniti pomoću funkcije **empty**.

Funkcija **isset** vraća *true* ili *false* ovisno o tome postoji li traženi element u nizu ili ne.

```
1  <?php
2  $array = [];
3  $names = ['Harry', 'Ron', 'Hermione'];
4
5  var_dump(empty($array)); // true
6  var_dump(empty($names)); // false
7  var_dump(isset($names[2])); // true
8  var_dump(isset($names[3])); // false
9  ?>
```

PHPOsnove/Primjeri/Nizovi/empty\_isset\_funkcije.php



# PHP nizovi – traženje elemenata

Jedna od najčešće korištenih funkcija u traženju elemenata u nizu je ***in\_array***.

Ova funkcija uzima dvije vrijednosti, vrijednost koju želite tražiti i niz u kojem tražite.

Funkcija vraća *true* ako je vrijednost u nizu, a u suprotnom je *false*.

Ponekad je korisnija funkcija ***array\_search***.

Ova funkcija djeluje na isti način, osim što, umjesto da vrati logičku vrijednost, vraća ključ tamo gdje se nalazi vrijednost koju tražimo, ili *false* ako vrijednost nije pronađena.

```
1  <?php
2  $names = ['Harry', 'Ron', 'Hermione'];
3
4  $containsHermione = in_array('Hermione', $names);
5  var_dump($containsHermione); // true
6
7  $containsSnape = in_array('Snape', $names);
8  var_dump($containsSnape); // false
9
10 $wheresRon = array_search('Ron', $names);
11 var_dump($wheresRon); // 1
12 |
13 $wheresVoldemort = array_search('Voldemort', $names);
14 var_dump($wheresVoldemort); // false
15 ?>
```

PHPOsnove/Primjeri/Nizovi/traženje\_elementata\_niza.php

# PHP nizovi – sortiranje

Niz se može sortirati na različite načine.

Postoji mogućnost da je redoslijed koji vam je potreban različit od trenutnog.

Niz je prema zadanim postavkama sortiran prema redoslijedu u kojem su dodani elementi, ali niz možete sortirati prema njegovom ključu ili prema vrijednosti, uzlazno i silazno.

Nadalje, kad sortirate niz prema vrijednostima, možete odabrati da sačuvate njihove ključeve ili generirate nove.

Potpuni popis funkcija nalazi se na službenim internetskim stranicama u dokumentaciji.

<http://php.net/manual/en/array.sorting.php>

Ove funkcije uvijek uzimaju jedan argument, niz i ne vraćaju ništa. Umjesto toga, one izravno sortiraju niz koji im prosljeđujemo.

# PHP nizovi – sortiranje

Prvo inicijaliziramo niz i dodijelimo ga **\$properties**.

Tada kreiramo tri varijable koje su kopije izvornog niza.

Zašto to radimo?

Ako sortiramo izvorni niz, više nećemo imati izvorni sadržaj. To nije ono što želimo u ovom konkretnom primjeru, jer želimo vidjeti kako različite funkcije sortiranja utječu na isti niz. Na kraju izvodimo tri različite vrste sortiranja i ispisujemo rezultate.

```
1  <?php
2  // Inicijalizacija niza
3  $properties = [
4      'firstname' => 'Tom',
5      'surname' => 'Riddle',
6      'house' => 'Slytherin'
7  ];
8  // dupliciranje niza u nove varijable
9  $properties1 = $properties2 = $properties3 = $properties;
10 // sortiranje po vrijednosti uzlazno
11 // ključevi se resetiraju
12 sort($properties1);
13 var_dump($properties1);
14 // sortiranje po vrijednosti uzlazno
15 // zadržava prvotne ključeve
16 asort($properties3);
17 var_dump($properties3);
18 // sortiranje po ključu uzlazno
19 ksort($properties2);
20 var_dump($properties2);
21 ?>
```

PHPOsnove/Primjeri/Nizovi/sortiranje\_niza.php

# PHP nizovi – ostale funkcije

Postoji oko 80 različitih funkcija povezanih s nizovima. Kao što možete pretpostaviti, za neke od njih nikada nećete ni čuti jer one imaju vrlo specifične svrhe.

Kompletan popis možete naći na:

<http://php.net/manual/en/book.array.php>

Možemo dobiti popis ključeva niza funkcijom **array\_keys** i popis njegovih vrijednosti funkcijom **array\_values**.

Možemo dobiti broj elemenata u nizu funkcijom **count**.

I možemo spojiti dva ili više niza u jedan funkcijom **array\_merge**.

```
1 <?php
2 $properties = [
3     'firstname' => 'Tom',
4     'surname' => 'Riddle',
5     'house' => 'Slytherin'
6 ];
7 // Dohvaćanje ključeva elemenata niza
8 $keys = array_keys($properties);
9 var_dump($keys);
10 // Dohvaćanje vrijednosti elemenata niza
11 $values = array_values($properties);
12 var_dump($values);
13 // Brojanje elemenata niza
14 $size = count($properties);
15 var_dump($size); // 3
16 // Spajanje dva niza u jedan
17 $good = ['Harry', 'Ron', 'Hermione'];
18 $bad = ['Dudley', 'Vernon', 'Petunia'];
19 $all = array_merge($good, $bad);
20 var_dump($all);
21 ?>
```

PHPOsnove/Primjeri/Nizovi/ostale\_funkcije\_niza.php

**Hvala na pažnji!**

