

# Osnove PHP-a



# PHP funkcije

Funkcija je blok koda za višekratnu upotrebu koja, uz unos, izvodi neke radnje i po želji vraća neki rezultat.

Već znate nekoliko unaprijed definiranih funkcija kao što su **empty**, **in\_array** ili **var\_dump**. Te funkcije dolaze s PHP-om, ali možete vrlo lako izraditi vlastite.

Funkcije možete upotrijebiti kada imate dijelove koda u aplikaciji koje je potrebno nekoliko puta izvršiti ili za definiranje neke funkcionalnosti.

PHP ne podržava preopterećene funkcije. Preopterećenje se odnosi na sposobnost deklariranja dvije ili više funkcija s istim nazivom, ali različitim argumentima. Argumente možete deklarirati čak i ako ne znate koje tipove podataka prihvataju, tako da PHP ne bi mogao odlučiti koju će funkciju koristiti.

# PHP funkcije – deklariranje i pozivanje

Funkcija ima naziv, argumente (opcionalno) i sadrži blok koda.

Po želji, može definirati koju vrijednost treba vratiti.

Naziv funkcije mora slijediti ista pravila kao i naziv varijable, odnosno mora započeti slovom ili podvlakom, a može sadržavati sva slova, brojeve ili podvlake.

Naziv funkcije ne smije biti rezervirana riječ, kao što su imena unaprijed definiranih funkcija.

Pozivanje funkcije se radi tako da se napiše naziv funkcije i na kraju doda zagrada.

```
1  <?php
2  // deklariranje funkcije
3  function writeHello(){
4      return "Hello World";
5  }
6  // pozivanje funkcije
7  writeHello();
8
9  ?>
```

PHPOsnove/Primjeri/Funkcije/deklariranje\_pozivanje.php

# PHP funkcije – vježba 1

- Proizvoljno deklarirajte funkciju koja vraća neki tekst.
- Pozovite funkciju i vraćenu vrijednost dodijelite varijabli.
- Ispišite vrijednost varijable.

# PHP funkcije – argumenti

Funkcija dobiva podatke izvana putem argumenata. Možete definirati bilo koji broj argumenata, uključujući 0 (nijedan).

Argumenti trebaju imati naziv kako bi se mogli koristiti unutar funkcije; ne mogu postojati dva argumenta s istim nazivom.

Prilikom pozivanja funkcije, podatke argumentima morate poslati istim redoslijedom kao što je deklarirano.

Funkcija može sadržavati neobavezne argumente, odnosno niste obavezni poslati vrijednost za te argumente. Kad deklarirate funkciju, trebate navesti zadane vrijednosti za te argumente. Dakle, u slučaju da korisnik ne daje vrijednost, funkcija će koristiti zadanu.

```
1  <?php
2  function addNumbers($a, $b, $printResult = false) {
3      $sum = $a + $b;
4      if ($printResult) {
5          echo 'The result is ' . $sum;
6      }
7      return $sum;
8  }
9
10 $sum1 = addNumbers(1, 2);
11 $sum1 = addNumbers(3, 4, false);
12 $sum1 = addNumbers(5, 6, true); // it will print the result
13
14 ?>
```

PHPOsnove/Primjeri/Funkcije/argumenti.php

# PHP funkcije – vježba 2

- Proizvoljno deklarirajte funkciju koja ima dva argumenta (**name** i **surname**). Funkcija treba konkatenerati podatke iz argumenata tako da između postoji razmak i dodijeliti ih lokalnoj varijabli, zatim treba vrijednost u varijabli pretvoriti u velika slova i to vratiti kao rezultat.
- Pozovite funkciju i vraćenu vrijednost dodijelite varijabli.
- Ispišite vrijednost varijable.

# PHP funkcije – doseg varijabli

Po dosegu varijable dijelimo na:

- **lokalne varijable** - vidljive samo unutar funkcije
- **globalne varijable** - varijable definirane izvan funkcija, vidljive su unutar cijele datoteke i svih uključenih datoteka.

```
1  <?php
2  // Globalna varijabla
3  $a = 'Algebra';
4
5  function variableScope() {
6      // Lokalna varijabla
7      $a = 'Backend developer';
8      // Ispis lokalne varijable
9      echo $a;
10 }
11 // Ispis Globalne varijable
12 echo $a;
13
14 ?>
```

PHPOsnove/Primjeri/Funkcije/doseg\_varijabli.php

# PHP funkcije – varijabla static

Druga važna značajka dosega varijabli je statička varijabla. Statička varijabla postoji samo u lokalnom dosegu funkcija, ali ne gubi vrijednost kada izvršenje programa napusti funkciju.

Varijabla **\$a** se inicijalizira samo u prvom pozivu funkcije i svaki put kada se pozove funkcija, ispisat će vrijednost **\$a** i povećavati je.

```
1  <?php
2  function staticScope()
3  {
4      static $a = 0;
5      echo $a;
6      $a++;
7  }
8  // Višestruko pozivanje funkcije
9  staticScope();
10 staticScope();
11 staticScope();
12 ?>
```

PHPOsnove/Primjeri/Funkcije/static.php



# PHP funkcije – kao varijable

PHP podržava koncept funkcija kao varijable. To znači da će, ako je vrijednost varijable jednaka nazivu funkcije, PHP tražiti funkciju s istim nazivom i pokušati je izvršiti.

Između ostalog, ovo se može koristiti za implementaciju povratnih poziva.

Ovaj način pozivanja funkcija neće raditi s ugrađenim PHP funkcijama kao što su: *echo*, *print*, *unset*, *isset*, *empty*, *include*, *require* i slične.

```
1  <?php
2  // This is a wrapper function around echo
3  function echoIt($string)
4  {
5      echo $string;
6  }
7
8  $func = 'echoIt';
9  $func('test'); // This calls echoIt()
10 ?>
```

PHPOsnove/Primjeri/Funkcije/kao\_varijable.php

# PHP funkcije – vježba 3

- Proizvoljno deklarirajte funkciju koja ima jedan argument (**number**). Funkcija treba imati lokalnu varijablu u koju će pribrojiti vrijednost argumenta number te istu vratiti kao rezultat. Vrijednost treba biti zadržana kod svakog poziva funkcije.
- Deklarirajte funkciju kao varijablu.
- Pozovite funkciju pomoću varijable, a kao vrijednost argumenta pošaljite slučajan broj u rasponu od 1 do 10 te ispišite rezultat.
- Ponovite postupak pet puta.

**Hvala na pažnji!**

