

Tehničko veleučilište u Zagrebu

Politehnički specijalistički diplomski stručni studij specijalizacija Informatika

Napredne tehnike programiranja web servisa
(.open-source)



Klijentsko-poslužiteljska arhitektura

- Web kao dio Interneta nastao je na klijentsko-poslužiteljskoj arhitekturi
- Računala koja traže određene informacije, tzv. klijentska računala ili klijenti
- Računala koja te informacije pružaju, tzv. poslužiteljska računala ili serveri

Klijentsko-poslužiteljska arhitektura

- TCP/IP (internetski) protokoli koji se koriste za prijenos preko: HTTP, SMTP, Telnet, DNS, ...
- Za web najvažniji je HTTP i njegova sigurnija inačica HTTPS
- Klijent je specifičan softver kojim krajnji korisnik pristupa sadržajima određenih servera
- Server je specifičan softver koji se izvršava na računalu predviđenom za prihvatanje klijentskih zahtjeva i da odgovora na iste

Klijentsko-poslužiteljska arhitektura

- Osobine klijenta
 - inicijalizira slanje zahtjeva (eng. request)
 - čeka i prima odaziv servera
 - obično se radi o desktop računalima koja imaju GUI za interakciju sa krajnjim korisnikom (osobom)
 - mogu biti i serveri koji traže neku informaciju (request) od drugih servera
 - tada server dobiva ulogu klijenta

Klijentsko-poslužiteljska arhitektura

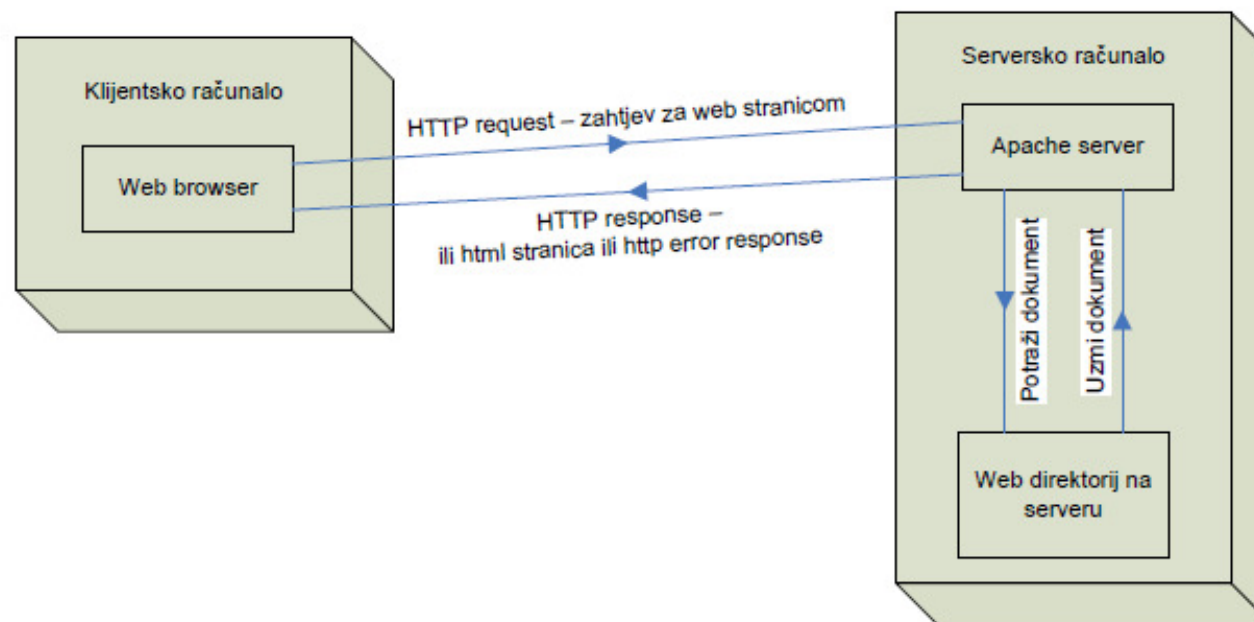
- Osobine servera
 - prima zahtjeve klijenata
 - pasivan je – ne inicijalizira slanje nego samo čeka na zahtjeve
 - po primitku zahtjeva obrađuje zahtjev i primjereno odgovara (eng. response) na klijentov upit/zahtjev
 - obično prima zahtjeve od više klijenata
 - obično ne vrši interakciju sa krajnjim korisnikom, iznimka je administrator

Kako radi web server

- Web server je računalu tj. softver na računalu koji servira web sadržaj (html dokumente, slike, datoteke i ostale web sadržaje)
- HTTP paketi obično dolaze na PORT 80, a HTTPS na port 443.
- Korisnik svojim web preglednikom (browserom ili klijentskim softverom) traži određeni sadržaj (web stranicu) od prikazanog web servera.
- Klijent u tom slučaju šalje HTTP request ili zahtjev za stranicom. HTTP (web) server presreće taj zahtjev i obrađuje ga

Kako radi web server

- Rezultat obrade zahtjeva je HTTP response tj. odgovor i on može biti ili html ili drugi dokument kojeg je klijent zatražio



Kako radi web server

- Apache server je program koji se pokreće u pozadini OS-a. On presreće HTTP request poruke, obrađuje i odgovara na njih
- Obrada se svodi na provjeru i dohvat html dokumenata, slika, css, js i ostalih skriptnih biblioteka i datoteka koji se nalaze u web direktoriju na serverskom računalu
- Ako se radi o skriptnom jeziku apache ili bilo koji drugi web server prvo dohvaća traženu skriptu te ju prosljeđuje dotičnom interpreteru skriptnog jezika

Što je PHP?

- PHP interpreter ili prevodilac je skup izvršnih modula i biblioteka koji naredbe pisane sintaksom PHP-a, prevode (interpretiraju) u html tekst
- PHP je open-source server-side skriptni programski jezik
- PHP je skraćenica od **P**ersonal **H**ome **P**age, a puni naziv mu je PHP HyperText Processor

Što je PHP?

- PHP je server side skriptni jezik, skripte se izvršavaju na serveru
- PHP-a je **open-source jezik**, koristi se besplatno
- Osnovna namjena PHP jezika je stvaranje dinamičkih web stranica i sadržaja (server-side programiranje)

PHP interpreter

- Da bi se PHP povezao sa nekim web serverom koristi se **SAPI** (eng. *Server Application Programming Interface*) modul ili **CGI** (eng. *Common Gateway Interface*) tj. **FastCGI** procesori
- PHP interpreter sastavljen je od PHP parsera, koji čita PHP skripte i SAPI/CGI/FastCGI modula tj. jezgre PHP-a

PHP interpreter

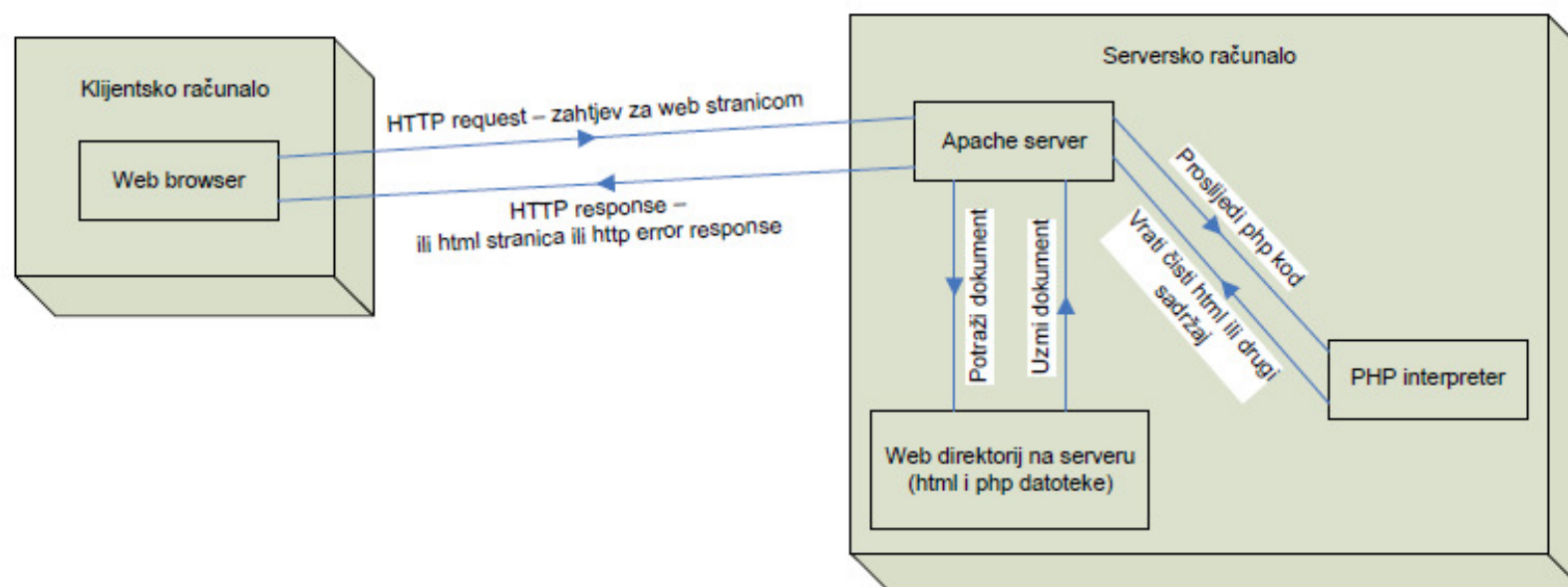
- Pročitane PHP naredbe, funkcije i ostalu sintaksu pretvaraju u html kod
- PHP interpreter koristi već napisane PHP, Perl skripte i biblioteke koje sadrže ugrađene funkcije, globalne varijable i određene postavke parsera



Apache i PHP interpreter

- Klijent šalje zahtjev za PHP stranicom koja sadrži PHP i HTML kod.
- Apache server provjeri da li datoteka postoji i ako da prosljeđuje je PHP interpreteru na obradu.
- Rezultat obrade HTML kod.
- Apache prima rezultat obrade PHP interpretera i šalje ga klijentu.

Apache i PHP interpreter



Što je PHP datoteka?

- PHP može sadržavati text HTML tagove i skripte
- PHP se vraća browseru kao običan HTML dokument
- PHP datoteka ima ekstenziju .php

Zašto PHP?

- PHP se može pogoniti na različitim platformama (MS, UNIX, LINUX..)
- PHP je kompatibilan s većinom WEB servera
- PHP se može slobodno instalirati i koristiti
- Na PHP službenoj strainici možemo naći sve funkcije i skripte koje PHP podržava www.php.net
- PHP se može brzo naučiti i efikasno koristiti kao platforma na serverskoj strani

PHP sintaksa

- PHP izvorni kod se ne može vidjeti odabirom view source u browseru
- Vidimo samo izlaz PHP file-a koj je HTML.
- Skripta je izvršena na serveru prije nego je rezultat poslan nazad prema pregledniku



PHP sintaksa

- PHP skriptni blok uvijek počinje s `<?php` i završava s `?>`. PHP skriptni blok može biti stavljen bilo gdje u dokumentu
- Na serveru sa odabranim-dozvoljenim kraticama možemo startati skriptni blok sa:
— **`<? i završiti s ?>`**
- Zbog maksimalne komapibilnosti, preporuča se korištenje standardne sintakse `<?php ... ?>` u odnosu na skraćeni oblik.

PHP sintaksa

- PHP dokument sadrži HTML tagove.
- U nastavku je jednostavan HTML kod koji šalje tekst "***Dobar dan***" pretraživaču.

```
<html>
<body>
  <?php
    echo "Dobar dan";
  ?>
</body>
</html>
```

PHP sintaksa

- Komentiranje linija koda ili ubacivanje komentara za određeni kod

```
<html>
<body>
  <?php
    // korištenje komentara jedna linija
    # komentar
    /*
      komentar blok
      više linija
    */
  ?>
</body>
</html>
```

PHP varijable

- Varijable koristimo za spremanje vrijednosti kao što su brojevi, stringovi ili rezultati funkcija.
- Varijable se koriste za spremanje vrijednosti kao što su brojevi, stringovi ili matrice.
- Kada je određena varijabla u PHP skripti ona se može koristiti više puta.
- Definicija varijabli počinje s **\$** simbolom:
- **`$var_name = value;`**

PHP varijable

- Varijable u php-u se ne definiraju po tipu podataka
- Varijable se definiraju tako da im ime počinje sa \$ (dolar) znakom, npr.: \$x, \$neka_varijabla, \$var...

PHP varijabla

- Prilikom spremanja određenog niza znakova u varijablu koristimo navodnike
- Kod spremanja broja tipa integer upisujemo samo broj

```
<html>
<body>
  <?php
    $txt = "Dobro jutro i dobar dan!";
    $txt = 'Dobro jutro i dobar dan!';
    $number = 16;
  ?>
</body>
</html>
```

PHP varijable

- Varijable ne moraju biti deklarirane prije nego što ih definiramo.
- U gornjem primjeru vidimo da nismo morali deklarirati varijable, ukazati PHP-u na podatkovne tipove varijabli.
- PHP automatski prevodi varijablu u odgovarajući podatkovni tip ovisno o tome kako smo ju definirali.

PHP variable

- U striktno određenim programskim jezicima morali smo definirati podatkovni tip i ime varijable prije nego što je možemo koristiti.

```
<html>
<body>
  <?php
      $number = 16;
      $number = (int)16;
  ?>
</body>
</html>
```

Pravila označavanja varijabli

- Ime varijable mora započeti sa slovom ili s _ (underscore)
- Ime varijable može sadržavati samo alfa-numeričke znakove (a-Z; 0-9) i _
- Ime varijable ne može sadržavati praznine. Ako je ime varijable sastavljeno od jedne ili više riječi mora bit razdvojeno s _, - ili korištenjem velikih slova npr. \$mojaVarijabla

Tipovi podataka

- Tipovi podataka koje podržava PHP:
 - Cijeli brojevi (integer)
 - Realni brojevi (floating-point numbers)
 - Tekstualni podaci (String)
 - Logičke varijable
 - Nizovi
 - Objekti

Stringovi u PHP-u

- String varijable se koriste za vrijednosti koje sadržavaju karakter tipove podataka.
- Kada kreiramo string možemo ga koristiti na razne načine – direktno u funkcijama ili ga možemo spremiti u varijablu.
- U nastavku kreiramo string \$txt varijablu koja sadrži string "Dobro jutro i dobar dan".

Stringovi u PHP-u

- Primjer

```
<html>
<body>
  <?php
    $txt="Dobro jutro i dobar dan!";
    echo $txt;
  ?>
</body>
</html>
```

Stringovi u PHP-u

- Primjer

```
<html>
<body>
  <?php
    $str_ime="PHP";
    echo "Moje ime je ".$str_ime;
  ?>
</body>
</html>
```

Korištenje duplih navodnika

- Kada koristimo duple navodnike unutar navodnika možemo koristiti posebne znakove
- Posebni znakovi slijede iza znaka backslash (\)

Znak	Značenje
\n	Novi red(LF ili 0x0A u ASCIIu)
\t	Tab razmak (HT ili 0x09 u ASCIIu)
\\	Backslash
\\$	Dolar znak
\"	Dupli navodnik

Korištenje duplih navodnika

- Primjer

```
<html>
<body>
  <?php
    echo 'Arnold once said: "I\'ll be back"';
  ?>
</body>
</html>
```

```
// Izlaz: Arnold once said: "I'll be back"
```


Korištenje duplih navodnika

- Primjer

```
<html>
<body>
  <?php
      echo 'You deleted C:\\*. *?';
  ?>
</body>
</html>
```

```
// Izlaz: You deleted C:\\*. *?
```

Operator konkatencije - spajanja

- Operator konkatencije (.) se koristi za spajanje dvije vrijednosti stringa.
- Koristimo . (operator točka)

```
<html>
<body>
  <?php
      $txt1="Dobro jutro";
      $txt2="studenti";
      echo $txt1 . " " . $txt2;

  ?>
</body>
</html>
```

```
// Izlaz: Dobro jutro studenti
```

PHP operatori

- Operatori se koriste za operacije s vrijednostima.

Operator	Opis	Primjer	Rezultat
+	Zbrajanje	x=2 x+2	4
-	Oduzimanje	x=2 5-x	3
*	Množenje	x=4 x*5	20
/	Dijeljenje	15/5 5/2	3 2.5
%	Ostatak dijeljenja	5%2 10%8 10%2	1 2 0

PHP operatori

- Operatori se koriste za operacije s vrijednostima.

Operator	Opis	Primjer	Rezultat
++	Povećanje	x=5 x++	6
--	Smanjenje	x=5 x--	4

PHP operatori

- Operatori upućivanja – dodjeljivanja

Operator	Primjer	Isto je kao:
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

PHP operatori

- Operatori usporedbe

Operator	Primjer	Isto je kao:
==	is equal to	5==8 (false)
!=	is not equal	5!=8 (true)
>	is greater than	5>8 (false)
<	is less than	5<8 (true)
>=	is greater than or equal to	5>=8 (false)
<=	is less than or equal to	5<=8 (true)

PHP operatori

- Logički operatori

Operator	Primjer	Opis:
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Funkcija DATE i TIME

- date — Format lokalnog vremena servera (time/date)
- Vraća vrijednost stringa ovisno o formatu kojeg definiramo

```
string date ( string $format [, int $timestamp = time() ] )
```

Format	Opis	Primjer
d	Dan u mjesecu (dvije znamenke)	01 do 31
D	Tekstualni prikaz dana	Mon do Sun
j	Dan u mjesecu (bez dvije znamenke)	1 do 31

Funkcija DATE i TIME

Format	Opis	Primjer
I	Puni naziv dana u mjesecu	Sunday do Saturday
F	Puni naziv mjeseca u godini	January do December
m	Numerički prikaz mjeseci u godini	01 do 12
M	Kratak prikaz mjeseca u godini	Jan do Dec
n	Numerički prikaz mjeseci u godini (bez dvije znamenke)	1 do 12
S	sufiks za dane u mjesecu	st, nd, rd ili th

Funkcija DATE i TIME

Format	Opis	Primjer
t	Broj dana u mjesecu	28 do 31
Y	Numerički prikaz godine	2013
y	Dvoznamenkasti prikaz godine	13

Funkcija DATE i TIME

Format	Opis	Primjer
a	prijepodne ili poslijepodne	am ili pm
A	prijepodne ili poslijepodne	AM ili PM
g	12- satni format datuma	1 do 12
G	24- satni format datuma	0 do 23
h	12- satni format datuma (dvoznamenkasti zapis)	01 do 12
H	24- satni format datuma (dvoznamenkasti zapis)	00 do 23

Funkcija DATE i TIME

Format	Opis	Primjer
i	minute	00 do 59
s	sekunde	00 do 59
u	mikrosekunde PHP 5.2.2	654321
e	Timezone PHP 5.1.0	UTC, GMT, Atlantic/Azores

Funkcija DATE i TIME

- Sintaksa

```
<html>
<body>
<?php
    $today = date("F j, Y, g:i a"); // March 10, 2013, 5:16 pm
    $today = date("m.d.y");        // 03.10.13
    $today = date("j, n, Y");      // 10, 3, 2013
?>
</body>
</html>
```

- Više informacija o funkcijama, formatima datuma i vremenu možete pogledati na:
<http://php.net/manual/en/function.date.php>

PHP If...else naredbe

- If, elseif i else naredbe u PHP-u se koriste za izvođenje različitih akcija baziranih na različitim uvjetima.
- if ...else naredbu koristimo kada želimo izvršiti određenu akciju – skup naredbi ako je postavljeni uvjet istinit a drugu akciju ako postavljeni uvjet nije istinit

If ...else naredba

- Sintaksa

```
if (uvjet) {  
    kod koji će se izvršiti ako je uvjet istinit;  
}  
else {  
    kod koji će se izvršiti ako uvjet nije istinit;  
}
```

If ...else naredba

- Slijedeći primjer će proizvesti izlaz Ugodan vikend ako je tekući dan petak a u suprotnom će izlaz biti Ugodan dan

```
<html>
<body>
<?php
    $d=date("D");
    if ($d=="Fri") { echo "Ugodan vikend!"; }
    else { echo "Ugodan dan!"; }
?>
</body>
</html>
```


If ...else naredba

- Primjer

```
<html>
<body>
<?php
    $d=date("D");
    if ($d=="Fri") {
        echo "Hello!<br />";
        echo "Ugodan vikend!";
        echo "Vidimo se u ponedjeljak!";
    }
?>
</body>
</html>
```

else if naredba

- Ako želimo izvršiti neki kod u slučaju da je jedan od nekoliko uvjeta istinit koristimo else if naredbu:

```
if (uvjet) { kod koji će se izvršiti ako je uvjet istinit; }
```

```
elseif (uvjet) { kod koji će se izvršiti ako je uvjet istinit; }
```

```
else { kod koji će se izvršiti ako uvjet nije istinit; }
```

else if naredba

- Slijedeći primjer će proizvesti izlaz “Ugodan vikend” ako je tekući dan petak; „Ugodna nedjelja”, ako je tekući dan nedjelja; a „Ugodan dan” u svim drugim slučajevima.

```
<html>
<body>
  <?php
    $d=date("D");
    if ($d=="Fri") { echo "Ugodan vikend!"; }
    else if ($d=="Sun") { echo "Ugodna nedjelja!"; }
    else { echo "Ugodan dan!"; }
  ?>
</body>
</html>
```

PHP Switch naredba

- Switch naredba u PHP-u se koristi za izvršavanje jedne od nekoliko različitih akcija baziranih na jednom ili više različitih uvjeta.
- Ako želimo selektirati jedan ili više blokova koda koji će biti izvršeni koristimo Switch naredbu.
- Switch naredba se koristi za izbjegavanje dugih blokova if...else if...else naredbe.

PHP Switch naredba

- Sintaksa

```
switch (izraz)
{
case slucaj1:
    kod koji će biti izvršen ako je izraz = slucaj1;
    break;
case slucaj2:
    kod koji će biti izvršen ako je izraz = slucaj2;
    break;
default:
    kod koji će biti izvršen ako je izraz različit od oba "slucaja";
}
```

Što je array (polja podataka)?

- Kad koristimo više sličnih varijabli.
- Umjesto više sličnih varijabli možemo podatke spremiti kao elemente u matricu – array.
- Svaki element u matrici ima svoj vlastiti ID i na taj način mu se može jednostavno pristupiti.

Što je array (polja podataka)?

- Postoje tri vrste matrica:
 - **numerička matrica** – matrica sa numeričkim ID ključem
 - **pridružena matrica** – ID ključevima je pridružena vrijednost
 - **multidimenzionalna matrica** – matrica koja sadrži jednu ili više matrica

Arrays

- Numerički array sprema svaki element sa numeričkim ID ključem
- Postoji nekoliko načina za kreiranje numeričkih matrica

Arrays

- Primjer 1:
 - U slijedećem primjeru ID će biti automatski pridružen

```
<html>
<body>
  <?php
    $imena = array("Zvonko","Petar","Štef");
  ?>
</body>
</html>
```

Arrays

- Primjer 2:
 - U slijedećem primjeru pridružujemo ID ključ ručno:

```
<html>
<body>
  <?php
    $imena[0] = "Zvonko";
    $imena[1] = "Petar";
    $imena[2] = "Štef";
  ?>
</body>
</html>
```

Arrays

- ID ključevi se mogu koristiti u skripti

```
<?php
    $imena[0] = "Zvonko";
    $imena[1] = "Petar";
    $imena[2] = "Štef";
    echo $imena[1] . " i " . $imena[2] . " su ". $imena[0] . "ovi susjedi";
?>
```

Izlaz: Petar i Štef su Zvonkovi susjedi

Arrays

```
<html>
<body>
  <?php
    $postBrojevi = array(10000, 51000, 21000, 31000);
    $gradovi = array(„Zagreb“, „Rijeka“, „Split“, „Osijek“);
  ?>
</body>
</html>
```

```
<?php
  echo $postBrojevi[2];
  echo $gradovi[2];
?>
```

Izlaz: 21000Split

Arrays

```
<?php  
    $novoPolje[3] = 20000;  
?>
```

- U navedenom primjeru varijabla \$novoPolje prije nije bila korištena.
- PHP varijablu \$novoPolje automatski pretvara u tip podataka polje i na četvrto mjesto upisuje vrijednost 20000.

Arrays

```
<?php  
    $novoPolje[] = 30000;  
?>
```

- Ako je podatak potrebno samo dodati u polje, a nije važno na koje mjesto tada je potrebno upisati samo uglate zagrade bez vrijednost polja u koje se upisuje.
- PHP će sam odrediti koja je zadnja upisana vrijednost u polje i dodat će novi podatak na kraj polja

PITANJA?

