



PHP

Ponovimo

- ◇ Varijable nije nužno deklarirati na početku skripte već se mogu deklarirati u trenutku kada su potrebne

Primjer:

```
$k = 'kolicina';  
$$k=100;  
echo 'kolicina'; //ispisuje 100
```

DEFINIRANJE REFERENCE NA VARIJABLU

```
$a=10;  
$b=&$a;  
echo $b;
```

- ◇ & - operator dohvaćanja adrese

```
$a=10;  
$b=&$a;  
$b=200;  
echo $a;
```

UKLANJANJE VARIJABLI IZ MEMORIJE

- ◆ PHP nudi i mehanizme uklanjanja varijabli iz memorije tijekom izvođenja skripte pomoću funkcije `unset()`

```
$a=10;  
$b=15;  
unset($b);
```

Ispis varijable

```
$a = 10;  
$b = 15;  
echo $a;  
echo $b;
```

```
$a = 10;  
$b = 15;  
echo '<b>Vrijednost varijable a: </b>';  
echo $a;  
echo '<br/>';  
echo '<b>Vrijednost varijable b: </b>';  
echo $b;
```

ili pomoću operatora spajanja nizova znakova (.) možemo ugraditi varijablu unutar HTML koda:

```
echo '<b>Vrijednost varijable a: </b>'.$a;
```

KONSTANTE

- ◊ Konstante se definiraju samo jednom i nakon toga im se više ne može mijenjati vrijednost
- ◊ Koristi se funkcija `define()`. Prvi parametar je naziv konstante, a drugi vrijednost.
- ◊ Za konstante se koristi samo naziv bez znaka `$`, zato ih je dobro pisati velikim tiskanim slovima i tako lakše uočavati u kodu.

```
define("PI", 3.14);  
const pi="PI";  
echo "Konstanta".pi."iznosi:".PI."<br>";
```

IZRAZI I OPERATORI

- ◇ Izrazi su dijelovi koda koji predstavljaju određenu cjelinu koja se može vrednovati i daje neki rezultat (vrijednost).
- ◇ Operatori uzimaju neke vrijednosti i s njima izvode određene operacije da bi dobili neki rezultat (međurezultat).
- ◇ Potrebno je poštivati prioritete operatora.
- ◇ Upotrebom zagrada možemo promijeniti prioritete operatora.

OPERATOR DODJELJIVANJA VRIJEDNOSTI

- ◊ Za dodjeljivanje vrijednosti varijabli (operandu) koja se nalazi lijevo od operatora.
- ◊ Desni operand može biti broj, niz znakova, izraz čiju vrijednost želimo pohraniti u varijablu ili ispisati...

```
$a=10;  
$b='PHP';  
$c=1+2*3;
```

OPERATORI PRETVORBE

- ◇ U PHP-u varijable se deklariraju prema potrebi, a PHP im odredi tip podataka prema vrijednosti koja im se dodjeljuje
- ◇ Operatori dodjeljivanja omogućuju definiranje točnog tipa podataka neke vrijednosti zbog:
- ◇ Kontrole nad dodjeljivanjem tipa podataka
- ◇ Prevođenja vrijednosti neke varijable iz jednog tipa podataka u drugi (npr. iz niza znakova u broj ili obrnuto)
- ◇ (int), (float), (string), (bool), (array), (object)



Programske strukture u PHP-u

IF, IF-ELSE

- ◆ Provjerava da li je izraz u zagradi istinit, pa izvršava odgovarajući dio koda
- ◆ Pomoću ključne riječi else definira se alternativni blok naredbi koje će se izvršiti ako izraz u zagradama nije istinit
- ◆ Dozvoljeno je i ugnježđivanje if-else naredbi

```
$x=5;  
if ($x==5)  
{echo 'Vrijednost varijable je 5 '};  
else  
{echo ' Vrijednost varijable različita je od 5 '};
```

```
$x=5;  
if ($x==5):  
Echo ' Vrijednost varijable je 5 ';  
else:  
echo 'Vrijednost varijable različita je od 5';  
endif:
```

IF-ELSEIF-ELSE

- ◆ Može se dodati i više alternativnih programskih blokova pomoću ključne riječi elseif

```
$a = 3;  
$b = 1;  
if ($a > $b) {  
    echo "a je veći od b";  
}  
elseif ($a == $b) {  
    echo "a jednak b";  
}  
else {  
    echo "a je manji od b";  
}
```

SWITCH-CASE

- ❖ Ispituje se opća vrijednost izraza (cjelobrojna ili niz znakova) i prema njoj izvršava određeni blok programskog koda
- ❖ blok nakon riječi default izvodi se ako ne postoji case sa vrijednošću varijable

```
$a=1;
switch ($a)
{
    case 1:
        echo 'Pozdrav';
        break;
    case 2:
        echo 'Zbogom';
        break;
    default:
        echo 'Vidimo se';
}
```

```
$a=1;
switch ($a):
    case 1:
        echo 'Pozdrav';
        break;
    case 2:
        echo 'Zbogom';
        break;
    default:
        echo 'Vidimo se';
endswitch;
```

Primjer

```
$i = 5;  
switch ($i) {  
    case 0:  
        echo "i je jednak 0";  
        break;  
    case 1:  
        echo "i je jednak 1";  
        break;  
    case 2:  
        echo "i je jednak 2";  
        break;  
    default:  
        echo "i nije ni 0, ni 1, ni 2";  
}
```

KLJUČNA RIJEČ BREAK I PROPADANJE (FALL-THROUGH)

- ◊ Oznaka da završava blok koda (case dio) koji je trebalo izvršiti
- ◊ Izlazi se iz switch-case naredbe
- ◊ Ako ne zatvorimo case dio pomoću ključne riječi break dolazi do propadanja, odnosno omogućujemo da se dio koda izvršava za dvije ili više vrijednosti varijable
- ◊ Ako propadanje radimo namjerno, poželjno je to komentirati

PETLJA WHILE

- ◆ Za ponavljanje nekog dijela programskog koda (npr. kod stvaranja HTML tablice s podacima u petlji iscrtavamo redove tablice dinamički)
- ◆ u svakom se koraku provjerava izraz u zagradama i dokle god je istinit izvršava se (ponavlja) odgovarajući dio programa
- ◆ Uz upotrebu ključne riječi `break` možemo prekinuti izvršavanje petlje i ranije.

```
$i=1;
while($i<=100)
{
    $zbroj+=$i;
    $i++;
}
echo $zbroj;
```

```
$i=1;
while($i<=100):
    $zbroj+=$i;
    $i++;
endwhile;
echo $zbroj;
```

```
$i=1;
while($i<=100)
{
    $zbroj+=$i;
    $i++;
    if ($zbroj>=50)
        break;
}
echo $zbroj;
```

PETLJA DO-WHILE

- ◆ Uvjet provjerava na kraju i ako je istinit vraća se na početak petlje; ključna riječ break može ranije prekinuti izvođenje
- ◆ Omogućava da se programski kod unutar petlje izvrši barem jednom

```
$a=1;  
do  
{  
    echo 'a='.$a;  
    echo '<br />';  
    $a++;  
} while ($a<=5);
```

FOR PETLJA

- ◆ Koristi se kad je unaprijed poznat broj ponavljanje.
- ◆ Deklarira se brojač i dodjeljuje mu se početnu vrijednost, zatim određuje uvjet koji određuje do kada će se petlja izvoditi, a treći je korak uvećanja.

```
for($i=1;$i<=5;$i++)  
{  
  echo 'i='.$i.'<br />';  
}
```

Primjer usporedbe for i while petlje

Količina	Cijena
10	50
20	100
30	150
40	200
50	250
60	300
70	350
80	400
90	450
100	500

```
<?php
$cijena = 5;
$b = 10;

echo "<table>";
echo '<table border="1" cellpadding="5">';
echo "<tr><th>Količina</th>";
echo "<th>Cijena</th></tr>";
  for ( $b= 10; $b <= 100; $b += 10){
    echo "<tr><td>";
    echo $b;
    echo "</td><td>";
    echo $cijena * $b;
    echo "</td></tr>";
  }
echo "</table>";
?>
```

```
<?php
$cijena = 5;
$b = 10;

echo "<table>";
echo '<table border="1" cellpadding="5">';
echo "<tr><th>Količina</th>";
echo "<th>Cijena</th></tr>";
  while ( $b <= 100 ) {
    echo "<tr><td>";
    echo $b;
    echo "</td><td>";
    echo $cijena * $b;
    echo "</td></tr>";
    $b = $b + 10;
  }
echo "</table>";
?>
```

Zadaci

- ◆ Korištenjem for petlje ispišite niz brojeva od 1 do 20 i njihove kvadrate. Svaki par ispisati u novi red.
- ◆ Napišite program koja će izračunavati prosječnu ocjenu razreda od 24 učenika na pismenom ispitu. Za simulaciju dobivenih ocjena koristite funkciju rand(). Koristiti do ili do-while petlju.

\$ocjena = rand(1,5); // varijabli \$ocjena dodijeli slučajnu vrijednost između 1 i 5.

```
$polje=array('Ivan', 'Luka', 'Hrvoje');  
foreach($polje as $ime)  
{  
echo $ime.'<br />';  
}
```

PETLJA FOREACH()

Ima specifičnu namjenu,
koristimo ju kod ispisa
elemenata polja.

Primjer

```
<?php
    $starost;
    $starost["Ivan"] = "28";
    $starost["Luka"] = "16";
    $starost["Josip"] = "35";
    $starost["Matej"] = "46";
    $starost["Hrvoje"] = "34";

    foreach( $starost as $indeks => $vrijednost){
        echo "Ime: $indeks, starost: $vrijednost <br />";
    }
?>
```

```
Ime: Ivan, starost: 28
Ime: Luka, starost: 16
Ime: Josip, starost: 35
Ime: Matej, starost: 46
Ime: Hrvoje, starost: 34
```



Polja u PHP-u

Polje

- ◆ Polje je poseban tip podataka, više podataka posloženih u skup parova tipa ključ-vrijednost (ključ služi za pristup podacima)
- ◆ **Primjer:**

```
$a=array(123, 15, 140);
```

Ključevi (indeksi) su: 0, 1 i 2

Podjela polja

- ◇ Polja razlikujemo prema tipu ključa:
 - ◇ **indeksna** (imaju brojčane ključeve) i
 - ◇ **asocijativna** (ključevi su tekstualni)

INDEKSNA POLJA

- ◊ u zagradi su vrijednosti članova, brojčani ključevi ovdje se dodjeljuju automatski, 0 1 2

```
$polje=array('Ivan', 'Luka', 'Hrvoje');
```

- ◊ Moguće je ključevima definirati vrijednosti:

```
$polje=array();  
$polje[10]='Ivan';  
$polje[11]='Luka';  
$polje[12]='Hrvoje';
```

- ◊ Ili

```
$polje=array(10=>'Ivan', 11=>'Luka', 12=>'Hrvoje');
```

- ◊ Novu vrijednost (na ključ za jedan veći od trenutno najvećeg) možemo dodati:

```
$polje[]='Jelena';
```

FUNKCIJA ARRAY_PUSH()

- ◆ Koristimo ju za dodavanje vrijednosti u polje:

```
$polje=array('Ivan','Luka','Hrvoje');  
array_push ($polje, 'Mark', 'Jelena');
```

ASOCIJATIVNA POLJA

```
$polje=array();  
$polje['ime1']='Ivan';  
$polje['ime2']='Luka';  
$polje['ime3']='Hrvoje';
```

```
$polje=array(ime1=>'Ivan',ime2=>'Luka',ime3=>'Hrvoje');
```

KOMBINIRANJE INDEKSNIH I ASOCIJATIVNIH KLJUČEVA

- ◆ PHP „ne zna” automatski uvećavati asocijativne ključeve, nego novo dodanom elementu dodjeljuje prvi slobodni indeksni ključ tako da neki elementi istog polja mogu imati asocijativne, a neki brojčane ključeve.

DOHVAĆANJE I ISPIS ELEMENATA POLJA

- ◆ Elemente polja možemo dohvatiti preko vrijednosti ključa i pomoću petlji

```
echo $polje1[1];  
echo $polje2['ime2'];  
  
for($i=0;$i<=2;$i++)  
    echo $polje1[$i];
```

- ◆ PHP funkcija count() za predano polje vraća koliko u njemu ima elemenata i tako omogućuje ispravno funkcioniranje ispisa u programu i kod dodavanja novih vrijednosti u polje

```
for($i=0;$i<count($polje1);$i++)  
    echo $polje1[$i];
```

ISPIS ASOCIJATIVNIH POLJA

- ◆ Polje koje ima asocijativne ključeve ispisujemo pomoću foreach petlje

```
foreach($polje2 as $key=>$val)  
    echo $val;
```

```
foreach($polj2 as $val)  
    echo $val;
```

```
<?php  
    $starost;  
    $starost["Ivan"] = "28";  
    $starost["Luka"] = "16";  
    $starost["Josip"] = "35";  
    $starost["Matej"] = "46";  
    $starost["Hrvoje"] = "34";  
  
    foreach( $starost as $indeks => $vrijednost){  
        echo "Ime: $indeks, starost: $vrijednost <br />";  
    }  
?>
```

VIŠEDIMENZIONALNA POLJA

- ◆ U jednodimenzionalnim poljima vrijednosti elemenata su brođane ili tekstualne.
- ◆ Kod višedimenzionalnih polja vrijednost elementa polja je novo polje.

```
$a=array();  
$a[]=array(10, 15, 20);  
$a[]=array(1, 5);  
$a[]=array(100, 200);  
echo $a[2][1]
```

0			1		2	
0	1	2	0	1	0	1
10	15	20	1	5	100	200

Ispis 2D polja

- ◆ Za ispis koristimo ugniježdenu foreach petlju

```
$p=array();  
$p[1][2][]=100;  
$p[1][2][]=200;  
$p[1][2][]=300;  
  
foreach($p as $element)  
{  
    foreach($element as $val)  
    {  
        echo $val.'<br />';  
    }  
}
```

INDEKSNA I ASOCIJATIVNA VIŠEDIMENZIONALNA POLJA

```
$p=array();  
$p[1]['ime']='Ivan';  
$p[1]['prezime']='Ivic';  
$p[2]['ime']='Luka';  
$p[2]['prezime']='Lukic';  
$p[3]['ime']='Petra';  
$p[3]['prezime']='Petric';
```

Glavno polje je indeksno i ima ključeve od 1 do 3. Svako polje unutar glavnih elemenata ima asocijativne ključeve koji se kod svakog novog polja ponavljaju.

```
for($i=1;$i<=3;$i++)
{
echo 'Ucenik br. '.$i.<br />';
echo 'Ime: '.$p[$i]['ime'].'<br />';
echo 'Prezime: '.$p[$i]['prezime'].'<br />';
}
```

```
foreach($p as $key=>$ucenik)
{
echo 'Ucenik br. '.$key.<br />';
echo 'Ime: '.$ucenik['ime'].'<br />';
echo 'Prezime: '.$ucenik['prezime'].'<br />';
}
```

NAČINI ISPISA POLJA

Elemente polja možemo ispisati pomoću for i foreach petlje.

Primjer

```
<?php
    $osebe = array(
        'Ivan'=>array('starost'=>32,'visina'=>172),
        'Luka'=>array('starost'=>18,'visina'=>183)
    );
    echo 'Luka ima ' . $osebe['Luka']['starost'] . ' godina.';
?>
```

Primjer

```
<table border="1" cellpadding="5">
<tr>
  <td>1</td>
  <td>Ivan</td>
</tr>
<tr>
  <td>2</td>
  <td>Luka</td>
</tr>
<tr>
  <td>3</td>
  <td>Jelena</td>
</tr>
</table>
```

```
<?php
$polje = array(1=>'Ivan', 2=>'Luka',3=>'Jelena');
echo '<table border="1" cellpadding="5">';
foreach($polje as $key=>$val)
{
echo '<tr>
<td>' . $key . '</td>
<td>' . $val . '</td>
</tr>';
}
echo '</table>'
?>
```

1	Ivan
2	Luka
3	Jelena

Primjer

Direktan pristup članovima preko indeksa

ruže koštaju 1.25 i možete uzeti 15
tulipani koštaju 0.75 i možete uzeti 25
orhideje koštaju 1.15 i možete uzeti 7

Pristup članovima pomoću petlje

1. Redak s indeksom 0

- o ruže
- o 1.25
- o 15

2. Redak s indeksom 1

- o tulipani
- o 0.75
- o 25

3. Redak s indeksom 2

- o orhideje
- o 1.15
- o 7

```
<?php
$shop = array( array("ruže", 1.25 , 15),
array("tulipani", 0.75 , 25),
array("orhideje", 1.15 , 7)
);
?>
```

```
<?php
echo "<h1>Direktan pristup članovima preko indeksa</h1>";
echo $shop[0][0]." koštaju ".$shop[0][1]." i možete uzeti ".$shop[0][2]."<br />";
echo $shop[1][0]." koštaju ".$shop[1][1]." i možete uzeti ".$shop[1][2]."<br />";
echo $shop[2][0]." koštaju ".$shop[2][1]." i možete uzeti ".$shop[2][2]."<br />";
```

```
echo "<h1>Pristup članovima pomoću petlje</h1>";
echo "<ol>";
for ($redak = 0; $redak < 3; $redak++)
{
echo "<li><b>Redak s indeksom $redak</b>";
echo "<ul>";
for ($stupac = 0; $stupac < 3; $stupac++)
{
echo "<li>".$shop[$redak][$stupac]."</li>";
}
echo "</ul>";
echo "</li>";
}
echo "</ol>";
?>
```

Zadatak

- ◇ Zadana je varijabla \$grad koja sadržava polje s članovima
godina => broj stanovnika
- ◇ Izračunajte:
 - a) Prosječan broj stanovnika kroz sve godine?
 - b) Koje godine je bilo najviše stanovnika?
 - c) Koliko godina se provodilo mjerenje?
- ◇ Primjer varijable **\$grad = array(2000 => 234551)**

Rješenje

```
<?php
$grad=array( 2000 => 234551,
2005 => 421522,
2010 => 520701,
2015 => 424214,
2020 => 65003);

$leng=count($grad);
$sum=0;
$maxx=0;
$maxi=0;

foreach($grad as $key=>$value){
$sum+=$value;
if ($value>$maxi){
$maxi=$value;
$maxx=$key;
}
}
echo "Prosjek stanovnika kroz sve godine je ".round($sum/$leng)."<br>";
echo "Najviše stanovnika bilo je ".$maxx."<br>";
echo "Mjerenje se provodilo kroz ".$leng." godina";
?>
```

Prosjek stanovnika kroz sve godine je 333198
Najviše stanovnika bilo je 2010
Mjerenje se provodilo kroz 5 godina

Funkcija count(\$polje) – vraća broj elemenata u polju.

FUNKCIJE

- ◆ Funkcija je izdvojeni blok programskog koda koji za zadatak ima izvesti neki posebni zadatak i upotrebljava se kad pri pisanju skripte neki blok koda koristimo više puta ili se pojavljuje na više mjesta.
- ◆ Funkcije se obično definiraju na početku skripte i pozivaju prema potrebi.
- ◆ PHP ima dvije vrste funkcija: ugrađene (`count()`, `rand()`, `array_push()`,...) i one koje definiramo sami.
- ◆ Da bi koristili ugrađene funkcije potrebno je poznavati njihov poziv i koji će rezultat vratiti. Nije potrebno poznavati način rada. Načinom rada upravljamo kod funkcija koje definiramo sami.
- ◆ Funkcije pozivamo tako da im napišemo ime i u zagradama prosljedimo potrebne parametre.

Ugrađene funkcije

- ◇ **time()** – vraća broj sekundi koji je prošao od ponoći 1. siječnja 1970. (UNIX timestamp):
`echo time();`
- ◇ **strlen()** – prebrojava znakove u nizu znakova
`echo strlen($rijec);`
- ◇ **is_array()** – vraća logičku vrijednost, provjerava je li predani parametar polje
`if (is_array($p))`
- ◇ **date()** – ovisno o parametru može ispisati neki podatak vezan uz trenutni datum i vrijeme
`echo date('d.m.Y');`
- ◇ **rand()** – vraća slučajno odabrani cijeli broj, za nasumični prikaz nekog broja
`echo rand();`
- ◇ ako je potrebno odrediti interval iz kojeg će se generirati nasumični broj
`echo rand(1, 10);`

Korisničke funkcije

- ◆ Funkcije definiramo pomoću ključne riječi **function** nakon koje slijedi ime funkcije.
- ◆ Nakon imena funkcije pišu se zagrade unutar kojih se definiraju varijable koje će primiti vrijednosti prosljeđenih parametara.
- ◆ Programski kod koji funkcija treba izvršiti piše se unutar vitičastih zagrada. Ako funkcija mora vratiti neku vrijednost, to se označava pomoću ključne riječi **return**.

```
<?php
function zbroji($x,$y) {
    $zbroj = $x + $y;
    return $zbroj; //povratna vrijednost
}

//poziv funkcije
echo '3 + 12 = ' . zbroji(3,12);
?>
```

3 + 12 = 15



Funkcije za rad s poljima

FUNKCIJE SORT I RSORT

- ◆ Obje funkcije sortiraju polje
- ◆ **sort()** sortira od najmanjeg ka najvećem ili od A do Z
- ◆ **rsort()** sortira od najvećeg prema najmanjem ili od Z do A.

```
$polje=array('A' => 5, 'B' => 25, 'C' => 32, 'D' => 14, 'E' => 15,);  
sort($polje);  
foreach ($polje as $value) {  
    echo "<br />";  
    echo $value ;  
}
```

```
5  
14  
15  
25  
32
```

```
$polje=array('A' => 5, 'B' => 25, 'C' => 32, 'D' => 14, 'E' => 15,);  
rsort($polje);  
foreach ($polje as $value) {  
    echo "<br />";  
    echo $value ;  
}
```

```
32  
25  
15  
14  
5
```

Funkcija print_r()

```
$boje = array('plava', 'zelena', 'crvena', 'bijela', 'crna');  
// vraca niz sortiran abecednim redom  
sort($boje);  
print_r($boje);  
print "<br />";  
// vraca niz sortiran obrnutim poretom  
rsort($boje);  
print_r($boje);
```

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

Funkcija **print_r()** prolazi kroz niz i ispisuje sve njegove ključeve i pridružene vrijednosti.

Funkcije `array_push()`/`array_pop()`

- ◆ Funkcija `array_push` dodaje jedan ili više elemenata na kraj niza
- ◆ Funkcija `array_pop()` uzima element s kraja niza

```
$hrana = array('sendvič', 'pizza', 'odrezak');  
// dodaje dva elementa nizu $hrana  
array_push($hrana, 'salata', 'kruh');  
print_r($hrana);
```

```
Array ( [0] => sendvič [1] => pizza [2] => odrezak [3] => salata [4] => kruh )
```

```
$hrana = array('sendvič', 'pizza', 'odrezak', 'salata', 'kruh');  
echo array_pop($hrana) . '<br/>';  
print_r($hrana);
```

```
kruh
```

```
Array ( [0] => sendvič [1] => pizza [2] => odrezak [3] => salata )
```

Zadaci

- ◆ Unijeti imena 5 učenika u polje pa ispisati sve učenike pomoću petlje na slijedeći način: Ime 1. učenika je Ivan..

Zadatak riješiti bez upotrebe nove varijable-brojača.

- ◆ Unijeti u polje podatke za udaljenosti Osijeka od najbliža susjedna četiri grada. Ispisati ime najbližeg grada, te koliko vremena je potrebno da bi se tamo stiglo ako je prosječna brzina 60 km/h.
- ◆ Unijeti u asocijativno polje imena i visine za 5 učenika pa ispisati imena najnižeg i najvišeg učenika.