

# ПРОГРАМИРАЊЕ ВЕБ АПЛИКАЦИЈА

## **Nastavna jedinica: Predavanje 4**

- Višedimenzionalni nizovi
- Funkcije za rad sa nizovima
- Include, require
- Rad sa datotekama
- HTML forme – čitanje podataka



# VIŠEDIMENZIONNI NIZOVI

- U slučaju da niz sadrži druge nizove kao elemente, postaje dvodimenzioni, trodimenzioni, itd...

```
<?php
// Dvodimenzioni niz:
$cars = array
(
    array("Volvo", 100, 96),
    array("BMW", 60, 59),
    array("Toyota", 110, 100)
);
?>
```

# VIŠEDIMENZIONNI NIZOVI

- Svaki element u podnizu, rekurzivno, takođe može biti niz:

```
<?php
// Dvodimenzioni asocijativni niz:
$amilies = array
(
    "Griffin"=>array("Peter", "Lois", "Megan"),
    "Quagmire"=>array("Glenn"),
    "Brown"=>array("Cleveland", "Loretta", "Junior")
);
?>
```

```
$proizvodi=array
(
    array("ID"=>"12",
        "Opis"=>"Akcija",
        "Cena"=>"2345"
    ),
    array("ID"=>"13",
        "Opis"=>"Triler",
        "Cena"=>"3345"
    ),
    array("ID"=>"14",
        "Opis"=>"Istorija",
        "Cena"=>"5345"
    )
);
print_r ($proizvodi);
```

# ISPIS POJEDINAČNOG ELEMENTA

- ▶ Ako ispisujemo pojedinačni element moramo ispratiti sve ključeve koje element ima

```
<?php
// Dvodimenzioni asocijativni niz:
$families = array
(
    "Griffin"=>array ( "Peter", "Lois", "Megan" ),
    "Quagmire"=>array("Glenn"),
    "Brown"=>array("Cleveland", "Loretta", "Junior")
);
echo "Da li je " . $families['Griffin'][1] . " deo Griffin familije?";
    //Izlaz: Da li je Lois deo Griffin familije?
?>
```

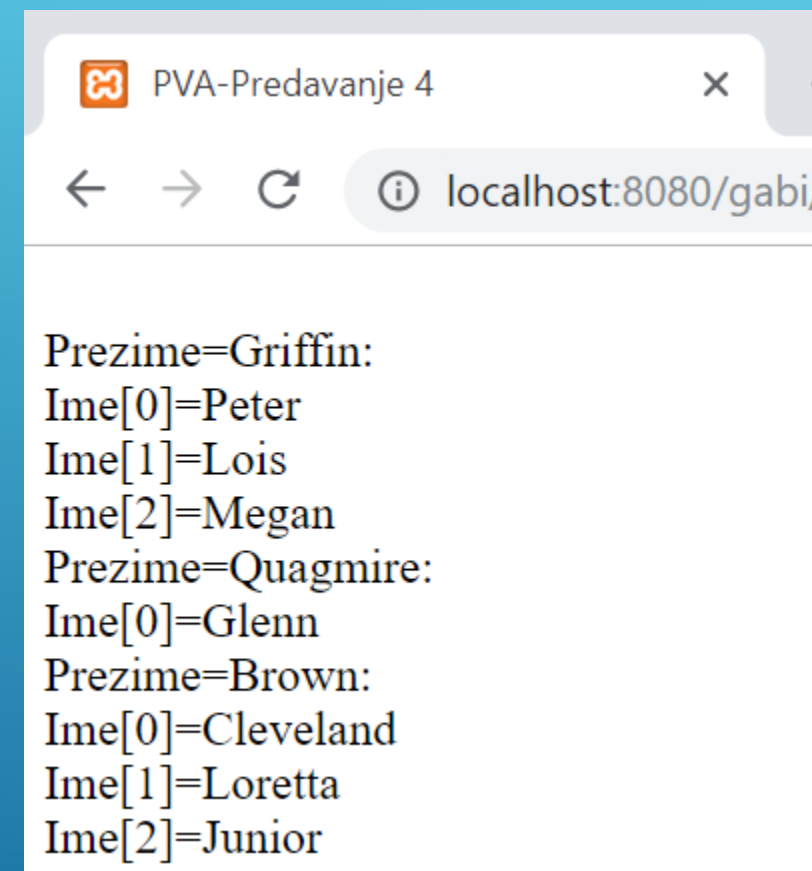
# ISPIS VIŠEDIMENZIONIH NIZOVA

- ▶ Za ispis ovakvih nizova možemo koristiti kombinaciju FOR i FOREACH petlji
- ▶ Ukoliko višedimenzioni niz ima numerički deo, koristimo FOR
- ▶ Za asocijativni deo niza moramo FOREACH
- ▶ Najjednostavnije je koristiti FOREACH za sve



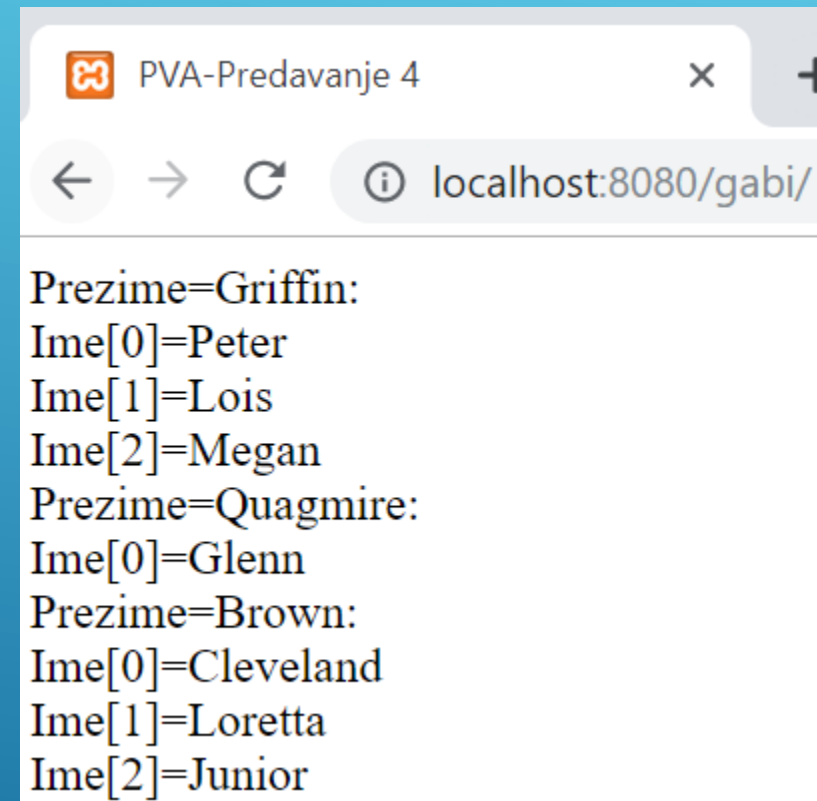
# ISPIS VIŠEDIMENZIONIH NIZOVA – FOR I FOREACH

```
<?php
// Dvodimenzioni asocijativni niz:
$families = array
(
    "Griffin"=>array ( "Peter", "Lois", "Megan"),
    "Quagmire"=>array ("Glenn"),
    "Brown"=>array ("Cleveland", "Loretta", "Junior")
);
foreach($families as $kljuc=>$vrednost)
{
    echo "Prezime=" . $kljuc . ":" ;
    echo "<br>";
    for($i=0;$i<count($vrednost);$i++)
    {
        echo "Ime[$i]=" . $vrednost[$i];
        echo "<br>";
    }
}
?>
```



# ISPIS VIŠEDIMENZIONIH NIZOVA – SAMO FOREACH

```
<?php
// Dvodimenzioni asocijativni niz:
$families = array
(
    "Griffin"=>array ( "Peter", "Lois", "Megan" ),
    "Quagmire"=>array ( "Glenn" ),
    "Brown"=>array ( "Cleveland", "Loretta", "Junior" )
);
foreach($families as $kljuc=>$vrednost)
{
    echo „Prezime=" . $kljuc;
    echo "<br>";
    foreach($vrednost as $y=>$y_value)
    {
        echo „Ime[$y]=".$y_value;
        echo "<br>";
    }
}
?>
```



Prezime=Griffin:  
Ime[0]=Peter  
Ime[1]=Lois  
Ime[2]=Megan  
Prezime=Quagmire:  
Ime[0]=Glenn  
Prezime=Brown:  
Ime[0]=Cleveland  
Ime[1]=Loretta  
Ime[2]=Junior

# FUNKCIJE ZA RAD SA NIZOVIMA

| Ime                                       | opis                                                                |
|-------------------------------------------|---------------------------------------------------------------------|
| <u><a href="#">array_diff()</a></u>       | Poredi vrednosti dva niza i vraća različite                         |
| <u><a href="#">array_fill()</a></u>       | Popunjava niz sa vrednostima.                                       |
| <u><a href="#">array_fill_keys()</a></u>  | Popunjava niz sa vrednostima specificirajući ključeve               |
| <u><a href="#">array_intersect()</a></u>  | Poredi nizove i vraća niz sa vrednostima koje se podudaraju.        |
| <u><a href="#">array_key_exists()</a></u> | Provjerava da li postoji navedeni ključ u nizu                      |
| <u><a href="#">array_merge()</a></u>      | Spaja jedan ili više nizova u novi rezultujući niz                  |
| <u><a href="#">array_pop()</a></u>        | Briše zadnji element niza                                           |
| <u><a href="#">array_push()</a></u>       | Dodaje jedan ili više elemenata na kraj niza                        |
| <u><a href="#">array_replace()</a></u>    | Menja vrednosti prvog niza sa vrednostima iz navedenih nizova       |
| <u><a href="#">array_search()</a></u>     | Pretražuje niz za zadatu vrednost i vraća ključ pronađene vrednosti |
| <u><a href="#">array_slice()</a></u>      | Vraća selektovani deo niza                                          |

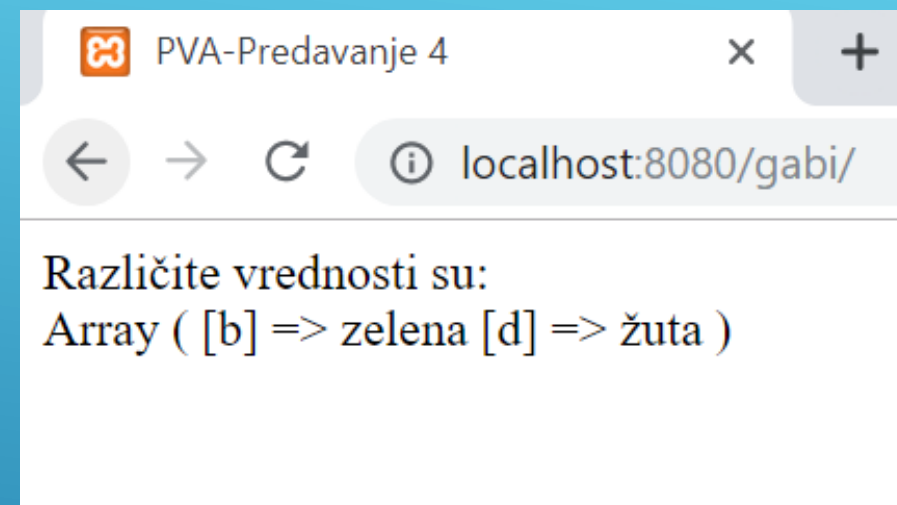
# ARRAY\_DIFF()

Poredi vrednosti dva niza i vraća različite

```
<?php
$a1=array("a"=>"red","b"=>"zelena","c"=>"blue","d"=>"žuta");
$a2=array("e"=>"red","f"=>"green","g"=>"blue");

$rezultatPoredjenja=array_diff($a1,$a2);

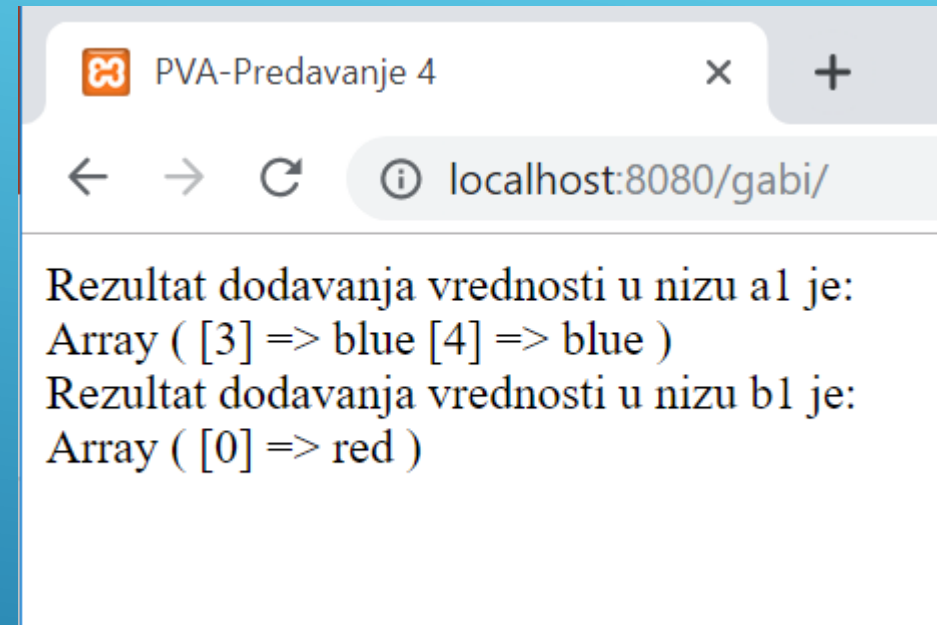
echo "Različite vrednosti su:<br>";
print_r($rezultatPoredjenja);
?>
```



# ARRAY\_FILL()

Popunjava niz sa vrednostima.

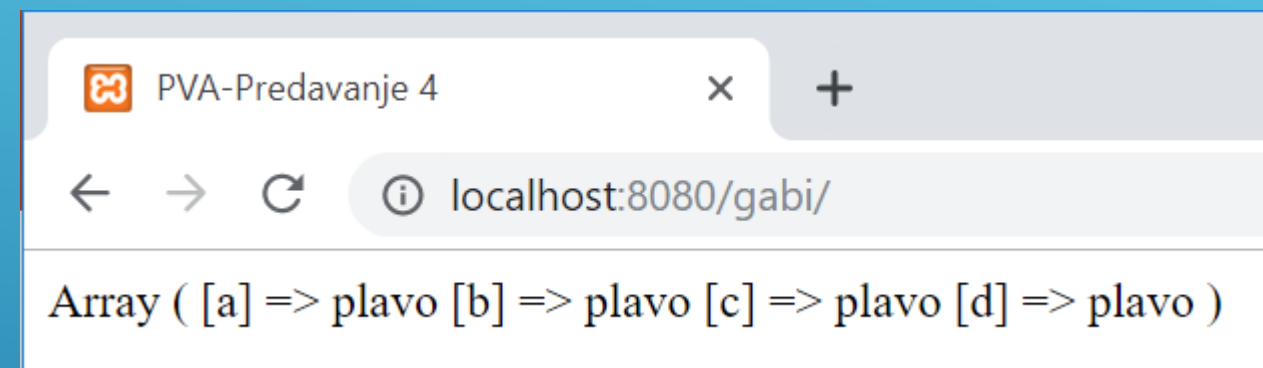
```
<?php
$a1=array_fill(3,2,"blue");
$b1=array_fill(0,1,"red");
echo "Rezultat dodavanja vrednosti u nizu a1 je:<br>";
print_r($a1);
echo "<br>";
echo "Rezultat dodavanja vrednosti u nizu b1 je:<br>";
print_r($b1);
?>
```



# ARRAY\_FILL\_KEYS()

Popunjava niz sa vrednostima specificirajući ključeve

```
<?php
$kljuc=array("a","b","c","d");
$a1=array_fill_keys($kljuc,"plavo");
print_r($a1);
?>
```

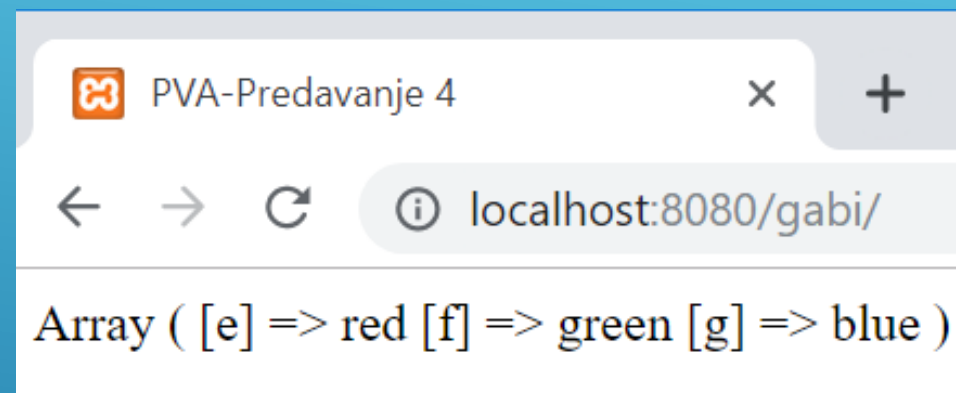


# ARRAY\_INTERSECT()

Poredi nizove i vraća niz sa vrednostima koje se podudaraju.

```
<?php
$a1=array("a"=>"red","b"=>"green","c"=>"blue","d"=>"yellow");
$a2=array("e"=>"red","f"=>"green","g"=>"blue");

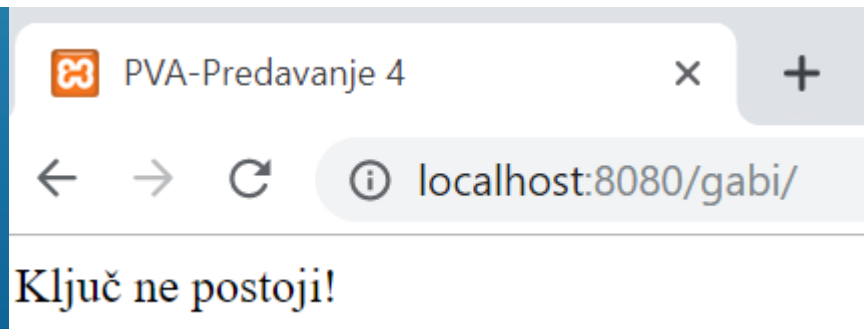
$result=array_intersect($a2,$a1);
print_r($result);
?>
```



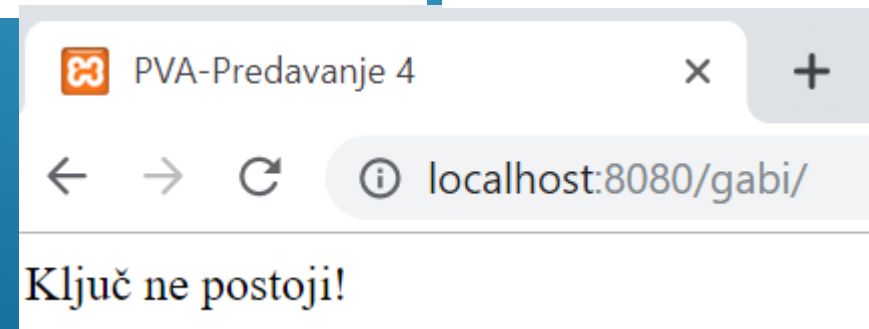
# ARRAY\_KEY\_EXISTS()

Proverava da li postoji navedeni ključ u nizu

```
<?php
$a=array("Volvo"=>"XC90", "BMW"=>"X5");
if (array_key_exists("Toyota",$a))
{
    echo "Ključ postoji!";
}
else
{
    echo "Ključ ne postoji!";
}
?>
```



```
<?php
$a=array("Volvo", "BMW");
if (array_key_exists(3,$a))
{
    echo "Ključ postoji!";
}
else
{
    echo "Ključ ne postoji!";
}
?>
```



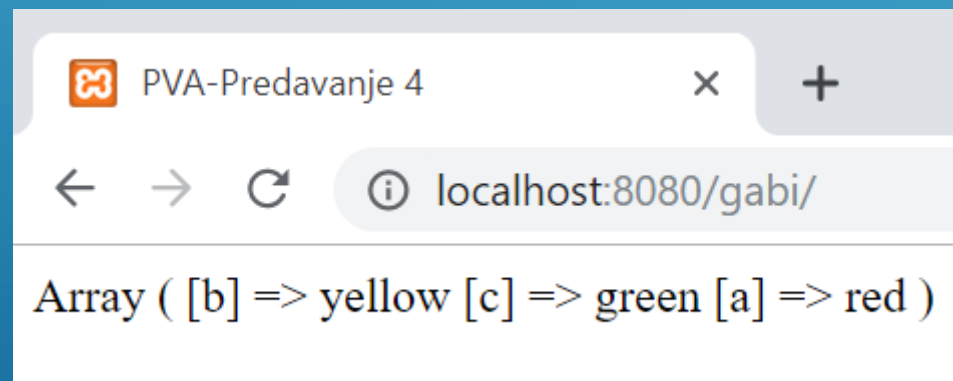
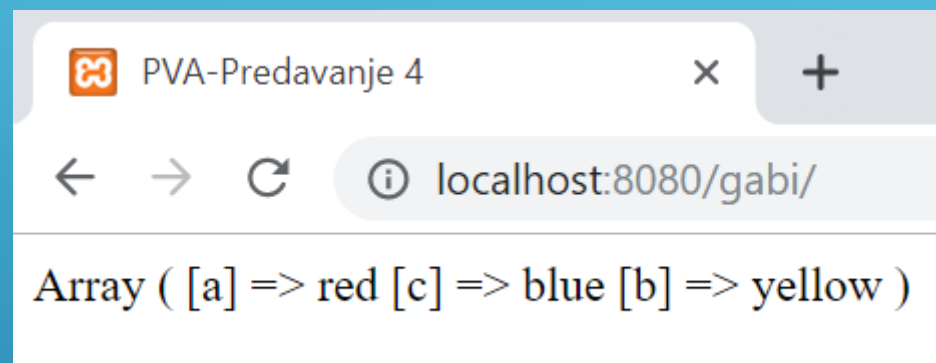
```
<?php
$b[1]="banana";
$b[2]="kivi";
if (array_key_exists(0,$b))
{
    echo "Ključ postoji!";
}
else
{
    echo "Ključ ne postoji!";
}
?>
```

# ARRAY\_MERGE()

Spaja jedan ili više nizova u novi rezultujući niz

```
<?php
$a1=array("a"=>"red","c"=>"green");
$a2=array("b"=>"yellow","c"=>"blue");
print_r(array_merge($a1,$a2));
?>
```

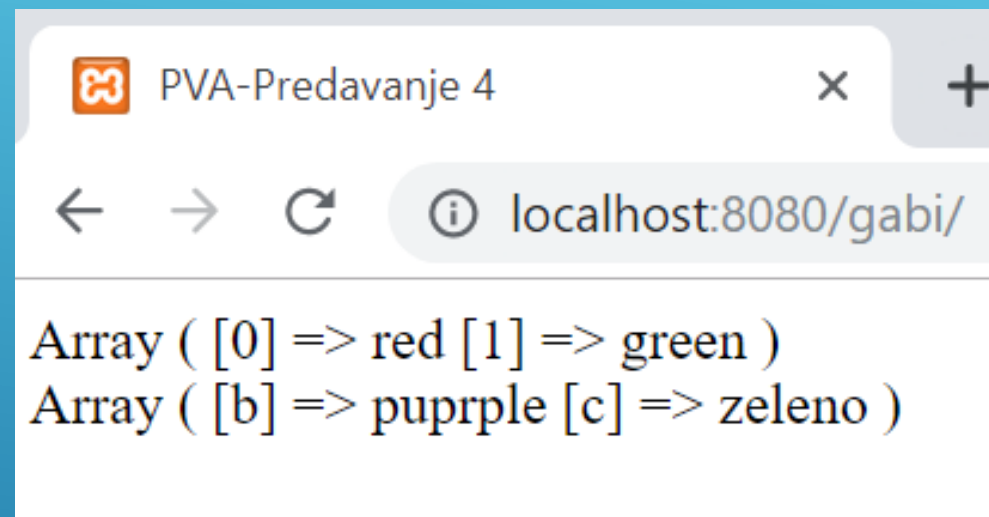
```
<?php
$a1=array("a"=>"red","c"=>"green");
$a2=array("b"=>"yellow","c"=>"blue");
print_r(array_merge($a2,$a1));
?>
```



# ARRAY\_POP()

Briše zadnji elemenat niza

```
<?php
$a=array("red","green","blue");
array_pop($a);
print_r($a);
echo "<br>";
$b=array("b"=>"puprple","c"=>"zeleno","d"=>"sivo");
array_pop($b);
print_r($b);
?>
```



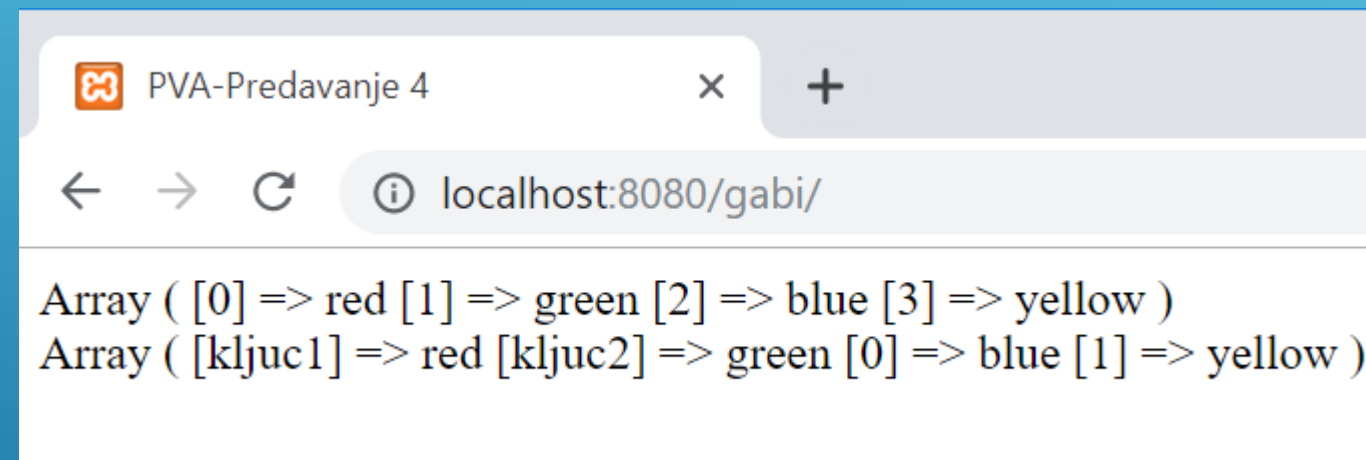
# ARRAY\_PUSH()

Dodaje jedan ili više elemenata na kraj niza

```
<?php
$a=array("red","green");
array_push($a,"blue","yellow");
print_r($a);

echo"<br>";

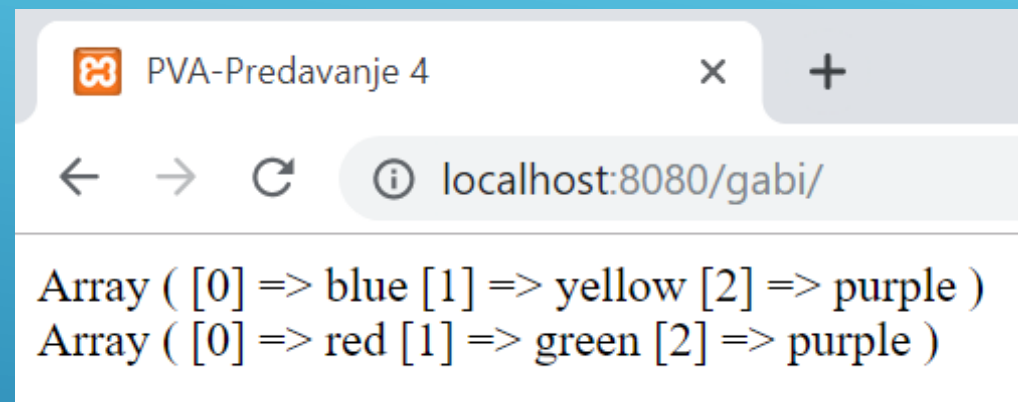
$b=array("kljuc1"=>"red","kljuc2"=>"green");
array_push($b,"blue","yellow");
print_r($b);
?>
```



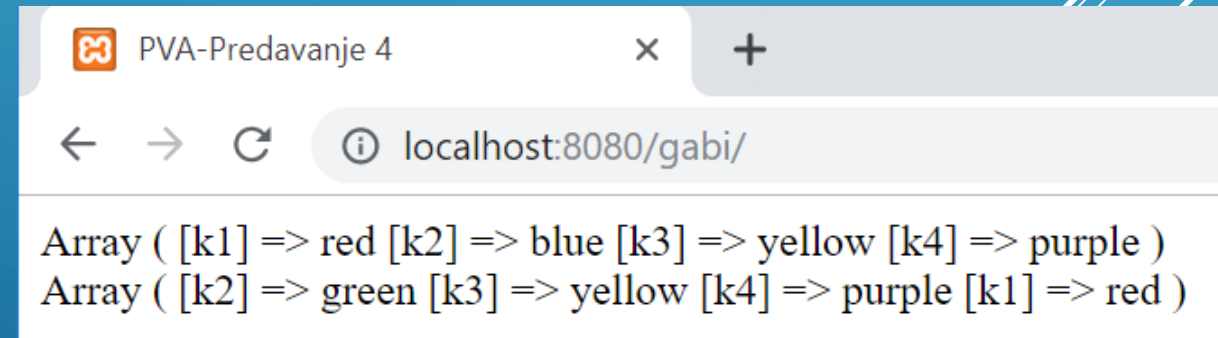
# ARRAY\_REPLACE()

Menja vrednosti prvog niza po indeksima sa vrednostima iz navedenih nizova

```
<?php
$a1=array("red","green");
$a2=array("blue","yellow","purple");
print_r(array_replace($a1,$a2));
echo "<br>";
print_r(array_replace($a2,$a1));
?>
```



```
<?php
$a1=array("k1"=>"red","k2"=>"green");
$a2=array("k2"=>"blue","k3"=>"yellow","k4"=>"purple");
print_r(array_replace($a1,$a2));
echo "<br>";
print_r(array_replace($a2,$a1));
?>
```

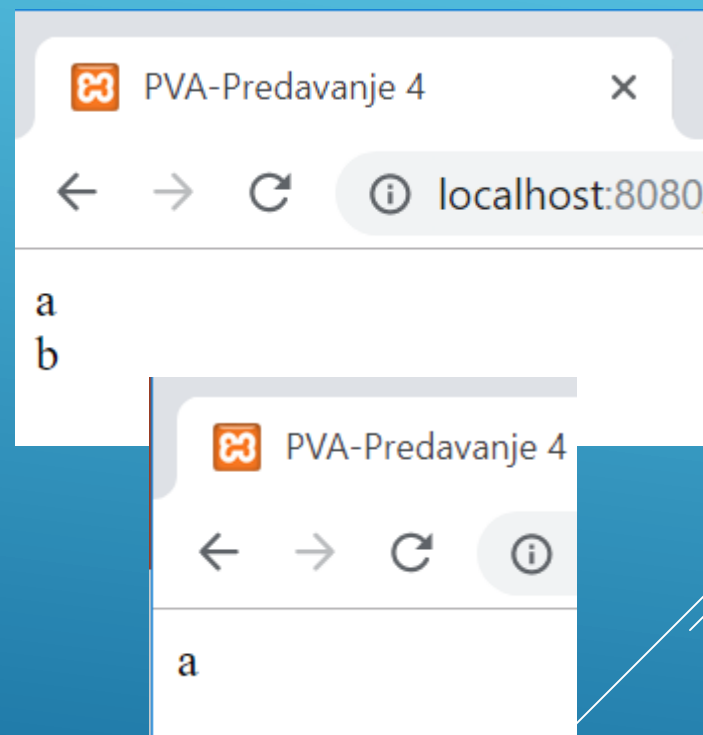


# ARRAY\_SEARCH()

Pretražuje niz za zadatu vrednost i vraća ključ pronađene vrednosti

```
<?php
$niz1=array("a"=>"red","b"=>"green","c"=>"blue");
echo array_search("red",$niz1);
echo "<br>";
$niz2=array("a"=>"5","b"=>5,"c"=>"5");
echo array_search(5,$niz2,true);
?>
```

```
$niz2=array("a"=>"5","b"=>5,"c"=>"5");
echo array_search(5,$niz2);
```



# ARRAY\_SLICE()

Vraća selektovani deo niza

array\_slice(varijabla niz, pozicija elementa od kog se vrši isecanje, broj elemenata rezultujućeg niza, sačuvati ili resetovati ključeve)

```
<?php  
$a=array("red","green","blue","yellow","brown");  
print_r(array_slice($a,2));  
?>
```

Array ( [0] => blue [1] => yellow [2] => brown )

```
<?php  
$a=array("red","green","blue","yellow","brown");  
print_r(array_slice($a,2,3));  
?>
```

Array ( [0] => blue [1] => yellow [2] => brown )

```
<?php  
$a=array("red","green","blue","yellow","brown");  
print_r(array_slice($a,-2,2));  
?>
```

Array ( [0] => yellow [1] => brown )



# ARRAY\_SLICE()

```
<?php
// Čuvanje ključeva postavljeno na TRUE:
$a=array("red","green","blue","yellow","brown");
echo "Rezultujući niz kada su sačuvani ključevi:<br>";
print_r(array_slice($a,1,2,true));
echo "<br>";
// Čuvanje ključeva postavljeno na FALSE(default vrednost):
$a=array("red","green","blue","yellow","brown");
echo "Rezultujući niz kada su resetovani ključevi:<br>";
print_r(array_slice($a,1,2,false));
```

Rezultujući niz kada su sačuvani ključevi:  
Array ( [1] => green [2] => blue )  
Rezultujući niz kada su resetovani ključevi:  
Array ( [0] => green [1] => blue )



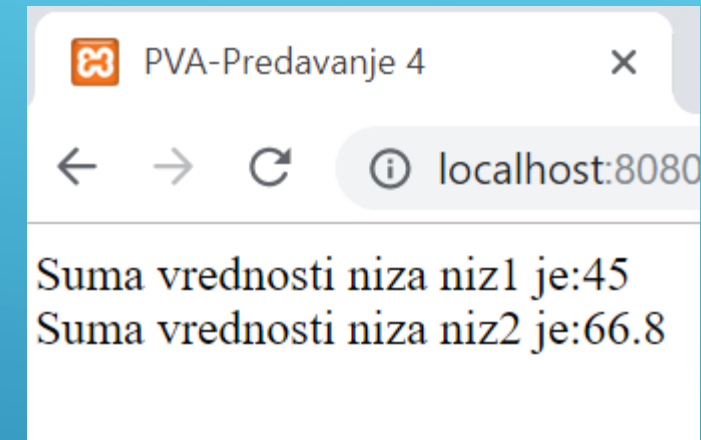
# FUNKCIJA ZA RAD SA NIZOVIMA

| Ime                   | opis                                                                                                                                   |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <u>array_sum()</u>    | Vraća sumu vrednosti u nizu                                                                                                            |
| <u>array_unique()</u> | Uklanja duplicirane vrednosti vrijednosti iz niza                                                                                      |
| <u>array_values()</u> | Vraća sve vrednosti niza                                                                                                               |
| <u>compact()</u>      | Kreira niz koji sadrži promenljive i njihove vrednosti                                                                                 |
| <u>count()</u>        | Vraća broj elemenata u nizu                                                                                                            |
| <u>extract()</u>      | Uvozi varijable u lokalnu tabelu simbola iz niza. Ova funkcija koristi ključeve niza kao imena promenljivih, a vrednosti kao vrednosti |
| <u>in_array()</u>     | Proverava da li postoji određena vrednost u nizu                                                                                       |

# ARRAY\_SUM()

Vraća sumu vrednosti u nizu

```
$niz1=array(5,15,25);  
echo "Suma vrednosti niza niz1 je:".array_sum($niz1);  
echo "<br>";  
$niz2=array("a"=>52.2,"b"=>13.7,"c"=>0.9);  
echo "Suma vrednosti niza niz2 je:".array_sum($niz2);
```



# ARRAY\_UNIQUE()

Uklanja duplicirane vrednosti vrednosti iz niza

```
<?php  
$a=array("a"=>"red","b"=>"green","c"=>"red","d"=>"red");  
print_r(array_unique($a));  
?>
```

```
Array ( [a] => red [b] => green )
```

# COMPACT()

Kreira niz koji sadrži promenljive i njihove vrednosti

```
<?php
$ime = "Peter";
$prezime = "Petrović";
$godine = "41";

$rezultat = compact("prezime", "ime", "godine");

print_r($rezultat);
?>
```



localhost:8080/gabi/

Array ( [prezime] => Petrović [ime] => Peter [godine] => 41 )



# EXTRACT()

```
<?php
$niz1=array("Prezime"=>"Petrović", "Ime"=>"Petar", "Godine"=>"45");
extract($niz1);
echo "\$a = $Prezime; \$b = $Ime; \$c = $Godine";
?>
```

`$a = Petrović; $b = Petar; $c = 45`

# IN\_ARRAY()

```
<?php
$niz2 = array("Petar", "Marko", "Aca", "Beograd", 23);

    if (in_array("23", $niz2, TRUE))
    {
        echo "Pronađena vrednost u nizu<br>";
    }
else
    {
        echo "Nije pronađena vrednost u nizu<br>";
    }
if (in_array("Marko",$niz2, TRUE))
    {
        echo "Pronađena vrednost u nizu<br>";
    }
else
    {
        echo "Nije pronađena vrednost u nizu<br>";
    }
}
```

```
if (in_array(23,$niz2,TRUE))
{
    echo "Pronađena vrednost u nizu.<br>";
}
else
{
    echo "Nije pronađena vrednost u nizu.<br>";
}

if (in_array("23",$niz2))
{
    echo "Pronađena vrednost u nizu.<br>";
}
else
{
    echo "Nije pronađena vrednost u nizu.<br>";
}
?>
```

Nije pronađena vrednost u nizu  
Pronađena vrednost u nizu  
Pronađena vrednost u nizu  
Pronađena vrednost u nizu



# FUNKCIJA DATE()

- ▶ Koristi se za formatiranje vremena i datuma.
- ▶ Format naredbe: **date (format, timestamp)**
- ▶ Vremenski pečat (**timestamp**) je niz karaktera koji označava datum i vreme dešavanja određenog događaja
- ▶ **date()** funkcija formatira vremenski pečat u čitljiv oblik
- ▶ Format – obavezno; određuje format vremenskog pečata
- ▶ Timestamp – opciono; specificira vremenski pečat – podrazumevana vrednost je trenutno vreme i datum



# FUNKCIJA DATE()

- ▶ Format je niz karaktera koji opisuje prikaz vremena na karakterističan način
  - ▶ d – dan (01-31)
  - ▶ m – mesec (01-12)
  - ▶ Y – godina (reprezentacija godine sa 4 cifre)
  - ▶ H – sati (00-23)
  - ▶ h – sati (01-12)
  - ▶ i – minuti (00-59)
  - ▶ s – sekunde (00-59)
  - ▶ l - dan u nedelji (malo slovo L)



# FUNKCIJA DATE()

```
<!DOCTYPE html>
<html>
<body>

© 2010-<?php echo date("Y");?>

</body>
</html>
```

© 2010-2019

```
<?php
echo "Danas je " . date("Y/m/d") . "<br>";
echo "Danas je " . date("Y.m.d") . "<br>";
echo "Danas je " . date("Y-m-d") . "<br>";
echo "Danas je " . date("d.m.Y") . "<br>";
echo "Danas je " . date("l");
?>
```

Danas je 2019/10/28  
Danas je 2019.10.28  
Danas je 2019-10-28  
Danas je 28.10.2019  
Danas je Monday



# FUNKCIJA DATE()

```
<?php
// Prikazuje dan u nedelji
echo date("l") . "<br>";
// Prikazuje numeričku reprezentaciju dana u nedelji
echo date("w") . "<br>";
// Prikazuje dan, datum,mesec, vreme, AM ili PM
echo date("l jS \of F Y h:i:s A") . "<br>";
//Prikazuje datum.mesec.godinu sati:minuti:sekunde
echo date("d.m.Y H:i:s");
echo "<br>";
//Prikazuje datum-mesec-godinu sati:minuti
echo date("d-m-Y H:i");
echo "<br>";
//Prikazuje datum/mesec/godinu
echo date("d/m/Y");
?>
```

```
Sunday
0
Sunday 27th of October 2019 08:53:13 PM
27.10.2019 20:53:13
27-10-2019 20:53
27/10/2019
```

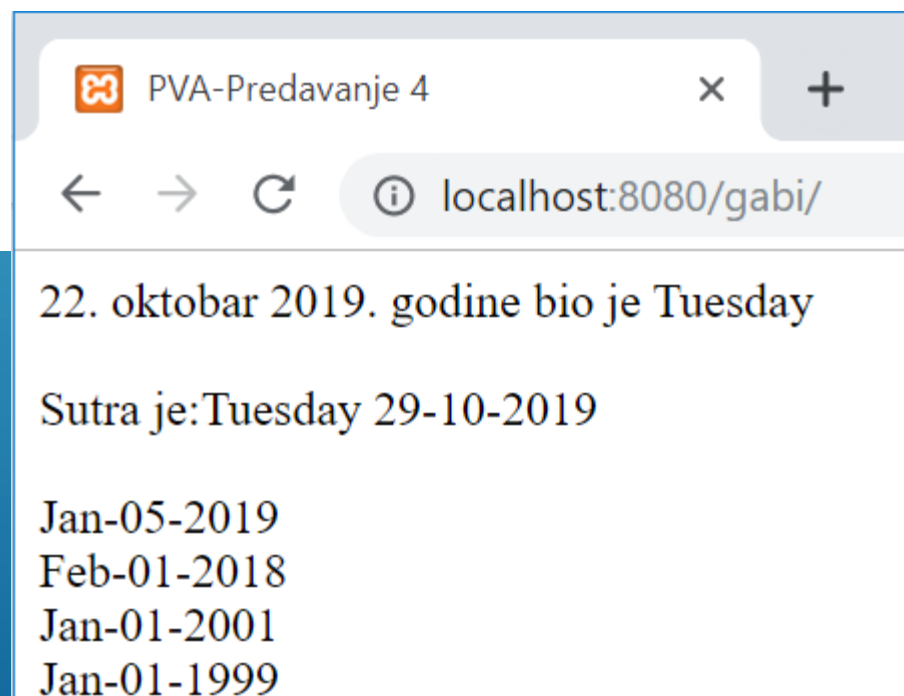
# FUNKCIJA MKTIME()

- ▶ **mktime(hour,minute,second,month,day,year,is\_dst)**
- ▶ Ukoliko ne specificiramo vreme, koristi se vreme izvršavanja skripta na serveru
- ▶ Ukoliko želimo da specificiramo neki datum, moramo ga pretvoriti u vreme proteklo od 01.01.1970 funkcijom **mktime()**
- ▶ Parametar is\_dst ima vrednost 1 u slučaju letnjeg računanja vremena, a 0 u slučaju zimskog. Podrazumevano je zimsko

```
<?php
    $sutra=mktime(0,0,0,date("m"), date("d")+1, date("Y"));
    echo "Cyтpa je: ".$sutra;//Izlaz je; Cyтpa je: 1571788800
?>
```

# FUNKCIJA MKTIME()

```
<?php
// Prikazuje da je 22.oktobar 2019.godine utorak
echo " 22. oktobar 2019. godine bio je". " ".date("l", mktime(0,0,0,10,22,2019))."<br><br>";
echo "Sutra je:". date("l d-m-Y", mktime(0,0,0,date("m"),date("d")+1,date("Y")))."<br><br>";
//Korisna za aritmetičke operacije i validaciju.
//Automatski izračunava validnu vrednost datuma ako su navedene vrednosti van opsega
echo date("M-d-Y",mktime(0,0,0,12,36,2018)) . "<br>";
echo date("M-d-Y",mktime(0,0,0,14,1,2017)) . "<br>";
echo date("M-d-Y",mktime(0,0,0,1,1,2001)) . "<br>";
echo date("M-d-Y",mktime(0,0,0,1,1,99)) . "<br>";
?>
```



# FUNKCIJA STRTOTIME()

- ▶ Ova PHP funkcija će pokušati da pretvori u vreme string koji se navede kao parametar.
- ▶ Parametar mora biti u formatu Engleskog datuma, ako se navodi datum

**Napomena:** Ova funkcija se mora koristiti oprezno jer ne znate da li ste dobili dobro vreme bez naknadne provere

```
<?php
echo strtotime("now"), "<br>";
echo strtotime("10 September 2019"), "<br>";
echo strtotime("+1 day"), "<br>";
echo strtotime("+1 week"), "<br>";
echo strtotime("+1 week 2 days 4 hours 2 seconds"), "<br>";
echo strtotime("next Thursday"), "<br>";
echo strtotime("last Monday"), "<br>";
?>
```

# INCLUDE() I REQUIRE() DIREKTIVE

- ▶ Retkost je da se ceo kod napisan u PHP-u nalazi u jednoj datoteci. Ako je kod napisan u više datoteka i treba da ih spojimo u jednu celinu možemo koristiti **include(imeDatoteke)** i **require(imeDatoteke)** direktive.
- ▶ Razlika je sledeća:
  - ▶ **include()** – ako *imeDatoteke* ne postoji PHP interpreter izbacuje upozorenje (warning), ali nastavlja da radi dalje
  - ▶ **require()** – ako *imeDatoteke* ne postoji PHP interpreter izbacuje grešku i ne nastavlja sa radom

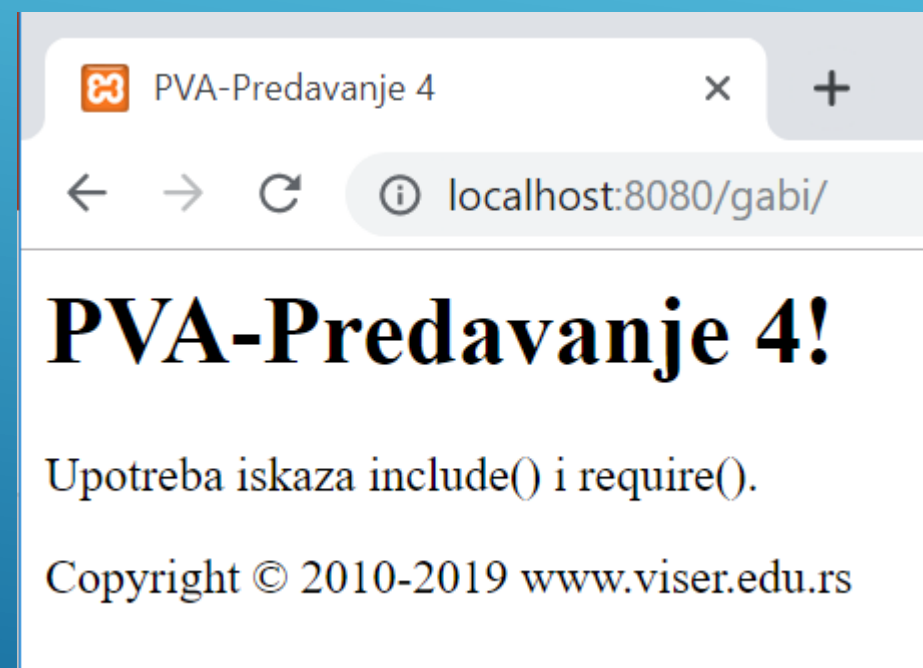
```
<?php
    include ("include.php") ;
    require ("require.php") ;
?>
```



# INCLUDE() I REQUIRE() DIREKTIVE

```
<?php  
echo "<p>Copyright &copy; 2010-" . date("Y") . " www.viser.edu.rs</p>";  
?>
```

```
<html>  
<head>  
<meta charset="utf-8">  
<title>PVA-Predavanje 4</title>  
</head>  
<body>  
<h1>PVA-Predavanje 4!</h1>  
<p>Upotreba iskaza include() i require().</p>  
<?php  
include ("footer.php");  
?>  
</body>
```



# INCLUDE\_ONCE() I REQUIRE\_ONCE() DIREKTIVE

- ▶ Obe prethodno navedene direktive imaju po jednu varijaciju. Ovim se obezbeđuje da datoteka koja se dodaje bude dodata samo jednom.
  - ▶ **include\_once(imeDatoteke)**
  - ▶ **require\_once(imeDatoteke)**
- ▶ Ovim se obezbeđuje da *imeDatoteke* bude dodata samo jednom, bez obzira koliko puta se pojavljuju direktive

```
<?php
    include_once ("include.php") ;
    require_once ("require.php") ;
?>
```

# DATOTEKE

- ▶ PHP može kreirati, čitati, menjati i brisati datoteke na serveru.
- ▶ Upisivanje podataka u datoteku:
  - ▶ Otvaranje datoteke
  - ▶ Upisivanje podataka u datoteku
  - ▶ Zatvaranje datoteke
- ▶ Čitanje podataka iz datoteke:
  - ▶ Otvaranje datoteke
  - ▶ Čitanje podataka iz datoteke
  - ▶ Zatvaranje datoteke
- ▶ Za otvaranje datoteke koristi se funkcija **fopen(imeDatoteke, nacinRada)**
  - ▶ *imeDatoteke* je putanja do datoteke (apsolutna ili relativna)
  - ▶ *nacinRada* je šta hoćemo da radimo sa datotekom



# NAČIN RADA SA DATOTEKAMA

| Režim | Ime            | Opis                                                                                                                                                                                                      |
|-------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 'r'   | Read           | Otvora datoteku za čitanje. Postavlja pokazivač na početak datoteke.                                                                                                                                      |
| 'r+'  | Read           | Otvora datoteku za čitanje i pisanje. Postavlja pokazivač na početak datoteke.                                                                                                                            |
| 'w'   | Write          | Otvora datoteku za upisivanje. Postavlja pokazivač na početak datoteke. Ukoliko datoteka postoji, postojeći sadržaj se briše. Ako datoteka ne postoji, sistem pokušava da je kreira.                      |
| 'w+'  | Write          | Otvora datoteku za upisivanje i čitanje. Postavlja pokazivač na početak datoteke. Ukoliko datoteka postoji, postojeći sadržaj se briše ako ga ima. Ako datoteka ne postoji, sistem pokušava da je kreira. |
| 'a'   | Append         | Otvora datoteku samo za dodavanje (upisivanje) od kraja postojećeg sadržaja. Ako datoteka ne postoji sistem pokušava da je napravi.                                                                       |
| 'a+'  | Append         | Otvora datoteku samo za dodavanje (upisivanje) i čitanje od kraja postojećeg sadržaja. Ako datoteka ne postoji sistem pokušava da je napravi.                                                             |
| 'x'   | Cautious write | Kreira i otvara datoteku za upisivanje, od početka. Kada datoteka postoji, sistem je ne otvara, funkcija <i>fopen()</i> vraća vrednost FALSE, a PHP generiše upozorenje.                                  |
| 'x+'  | Cautious write | Kreira i otvara datoteku za upisivanje i čitanje, od početka. Kada datoteka postoji, sistem je ne otvara, funkcija <i>fopen()</i> vraća vrednost FALSE, a PHP generiše upozorenje.                        |

# NAČIN RADA SA DATOTEKAMA

```
<?php
```

```
    $handle = fopen("/home/rasmus/file.txt", "r");
```

```
    $handle = fopen("/home/rasmus/file.gif", "w");
```

```
    $handle = fopen("http://www.example.com/", "r");
```

```
    $handle = fopen("ftp://user:password@example.com/somefile.txt", "w");
```

```
?>
```

# OTVARANJA DATOTEKE

- ▶ Ukoliko pokušavate da otvorite datoteku u „w“ načinu rada
  - ▶ Ako datoteka ne postoji biće kreirana
  - ▶ Ako datoteka postoji sav tekst u njoj će biti izbrisan i upisan novi
- ▶ Ukoliko pokušavate da otvorite datoteku u „r“ načinu rada
  - ▶ Ako datoteka ne postoji, PHP interpreter će izbaciti grešku
  - ▶ Ukoliko datoteka postoji, biće otvorena za čitanje
- ▶ Ukoliko pokušavate da otvorite datoteku u „a“ načinu rada
  - ▶ Ako datoteka ne postoji biće kreirana
  - ▶ Ako datoteka postoji, biće otvorena za upis, ali će novi sadržaj biti dodat na kraj starog, ukoliko stari postoji

# FILE\_EXISTS() I DIE() DIREKTIVE

► Da bi skript radio ispravno možemo

1. Proveriti da li datoteka postoji pre čitanja (**file\_exists()**) ili
2. Obezbediti da ako ne postoji obavestimo korisnika o tome (**die()**)

```
<?php
    if(file_exists("primer.txt"))
        $handle=fopen("primer.txt", "r");
    else
        echo "Datoteka 'primer.txt' ne postoji";
?>
```

```
<?php
    $handle=fopen("primer.txt", "r") or die("Datoteka 'primer.txt' ne postoji");
?>
```



# ZATVARANJE DATOTEKE - FCLOSE()

- ▶ Zatvaranje datoteke se vrši direktivom **fclose()**
- ▶ Parametar je referenca na otvorenu datoteku

```
<?php
    $file=fopen("primer.txt", "r");
    /* rad sa datotekom */
    fclose($file);
?>
```

# PROVERA KRAJA DATOTEKE – FEOF()

- ▶ Funkcija feof() proverava da li se došlo do kraja datoteke (end-of-file (EOF))
- ▶ EOF simbol se razlikuje u zavisnosti od operativnog sistema
- ▶ Nije moguće čitanje (read) iz datoteke koja je otvorena u w, a i x načinu rada tako da ovu proveru možemo vršiti samo sa datotekama koje su otvorene u r načinu rada

```
<?php
    $file=fopen("primer.txt", "r");
    /* rad sa datotekom */
    if (feof($file)) echo "Kraj datoteke";
    fclose($file);
?>
```



# UPISIVANJE U DATOTEKU

- ▶ Ukoliko upisujemo string u datoteku, možemo to uraditi na nekoliko načina. Dva najčešća su:
  - ▶ **fwrite(file, string)** – mora da postoji referenca na datoteku
  - ▶ **file\_put\_contents(file\_name, string)** – ne mora da postoji referenca na datoteku
- ▶ Obe direktive daju identičan rezultat

```
<?php
    $file=fopen("primer.txt", "w");
    fwrite($file, "Ovo je tekst za upis u datoteku");
    fclose($file);
?>
```

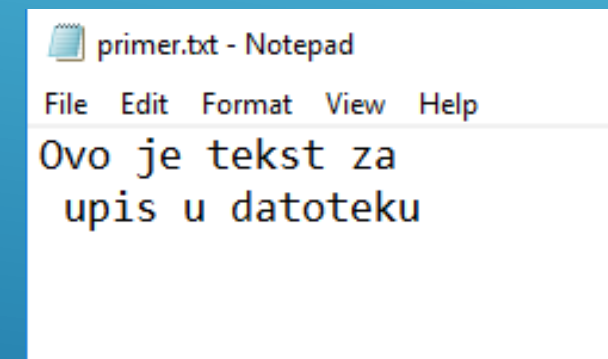
```
<?php
    file_put_contents("primer.txt", "Ovo je tekst za upis u datoteku");
?>
```



# UPISIVANJE U DATOTEKU

- ▶ Ukoliko želimo da u datoteku dodamo novi red moramo koristiti oznake za novi red ta .txt datoteke
- ▶ Pod Windows operativnim sistemom to je kombinacija simbola „\r\n“ – carriage return i line feed

```
<?php
    $file=fopen("primer.txt", "w");
    fwrite($file, "Ovo je tekst za \r\n upis u datoteku");
    fclose($file);
?>
```



```
<?php
    file_put_contents("primer.txt", "Ovo je tekst za \r\n upis u datoteku");
?>
```

# ČITANJE IZ DATOTEKE

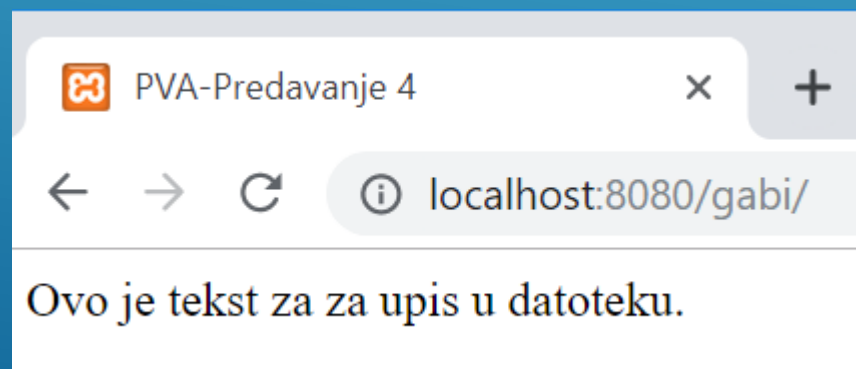
- ▶ Postoji nekoliko načina za čitanje datoteke
  - ▶ **file\_get\_contents(imeDatoteke)** - čitanje cele datoteke
  - ▶ **fgets(file)** – čitanje red po red
  - ▶ **fgetc(file)** – čitanje karakter po karakter
- ▶ Prilikom čitanja iz datoteke moramo voditi računa o novom redu.
- ▶ U datoteci to je `\r\n`, a u HTML-u to je **<br>**



# FILE\_GET\_CONTENTS()

- Direktiva ***file\_get\_contents(imeDatoteke)*** čita celu datoteku *imeDatoteke* i smešta u string

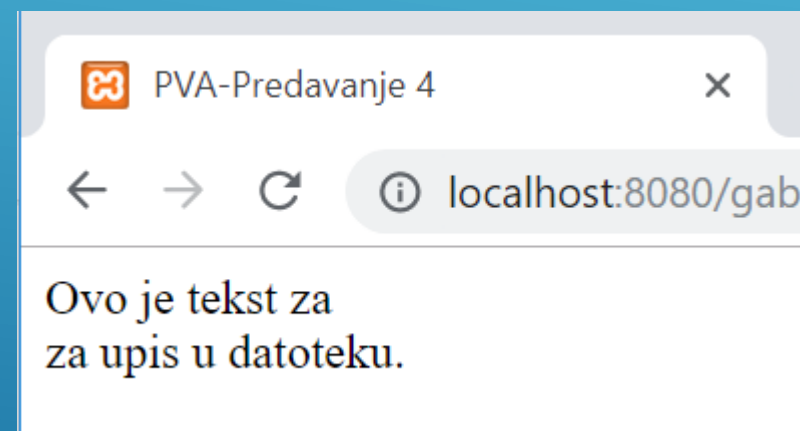
```
<?php
    $tekst=file_get_contents("primer.txt");
    echo $tekst; //Izlaz je: Ovo je tekst za upis u datoteku
?>
```



# FILE\_GET\_CONTENTS()

- ▶ Obzirom da HTML ne prepoznaje novi red u tekstualnoj datoteci, ceo tekst će biti prikazan u jednom redu
- ▶ Da bismo rešili ovaj problem koristićemo funkciju za rad sa stringovima **str\_replace()**

```
<?php
    $tekst=file_get_contents("primer.txt");
    $tekst=str_replace("\r\n", "<br>", $tekst);
    echo $tekst;
?>
```



# ČITANJE RED PO RED – FGETS()

```
<?php
$file=fopen("primer.txt", "r") or die("Nema datoteke!!!!");
//čitanje do kraja datoteke
while(!feof($file))
{
    $string= fgets($file);
    echo $string."<br>";
}
fclose($file);
?>
```

# ČITANJE KARAKTER PO KARAKTER – FGETC()

```
<?php
$file=fopen("primer.txt", "r") or die("Nema datoteke!!!!");
//čitanje do kraja datoteke
while(!feof($file))
{
    $karakter = fgetc($file);
    if($karakter=="\n") $karakter="<br>";
    echo $karakter;
}
fclose($file);
?>
```

# BRISANJE DATOTEKE

```
<?php  
    unlink("primer.txt");  
?>
```

# NEKE OD FUNKCIJA ZA RAD SA STRINGOVIMA

- ▶ addslashes — Bitno zbog prikaza u HTML-u. Ispred svakog znaka navoda stavi kosu crtu jer treba da prikaže te znake.
- ▶ stripslashes — Iz povratne informacije koju pročita iz baze skloni znake navoda
- ▶ explode — Split a string by a string
- ▶ htmlentities — Konvertuje sve dostupne karaktere u HTML entitete
- ▶ htmlspecialchars — Convert special characters to HTML entities
- ▶ implode — Join array elements with a string
- ▶ join — Alias of implode
- ▶ crypt — Metode za hešovanje
- ▶ md5 — Calculate the md5 hash of a string
- ▶ str\_repeat — Repeat a string
- ▶ str\_replace — Replace all occurrences of the search string with the replacement string



# NEKE OD FUNKCIJA ZA RAD SA STRINGOVIMA

- ▶ strcmp — Binary safe string comparison. Poredi string proveravajući velika i mala slova.
- ▶ strrev — Reverse a string
- ▶ strtolower — Make a string lowercase
- ▶ strtoupper — Make a string uppercase
- ▶ strtr — Translate characters or replace substrings
- ▶ substr\_compare — Binary safe comparison of two strings from an offset, up to length characters
- ▶ substr\_count — Count the number of substring occurrences
- ▶ substr\_replace — Replace text within a portion of a string
- ▶ substr — Return part of a string (Navodi se pozicija i broj karaktera koji prikazuje)
- ▶ trim — Strip whitespace (or other characters) from the beginning and end of a string



# PREUZIMANJE PODATAKA IZ FORMULARA

- ▶ Korisnički podaci iz forme do servera se mogu proslediti na dva načina:
  - ▶ Direktno submitovanjem forme iz HTML-a
  - ▶ Slanjem podataka iz AJAX-a
- ▶ Za razliku od GET, podaci koji se šalju POST metodom su nevidljivi za korisnika i ne dodaju se URL adresi. Oni se prenose u telu HTTP zahteva.
- ▶ POST metod nema ograničenje u smislu veličine, dok GET ima, zbog ograničenja dužine URL adrese
- ▶ GET i POST su dva metoda HTTP-a, koja se mogu koristiti za komunikaciju browser-a i servera
- ▶ Postoji mnogo više metoda HTTP-a, ali ova dva su ograničenje ako se podaci šalje iz web forme
- ▶ GET treba koristiti kada se serveru obraćamo za dohvaćanje sadržaja, kojim se ne menjaju podaci u bazi podataka (select)
- ▶ POST treba koristiti kada se serveru šalju podaci sa kojima se menjaju podaci u bazi podataka (insert, update, delete)



# VEZA HTML I PHP STRANICE

1. Napraviti HTML stranicu sa tagom *form*, atributima *action* i *method*
2. U formi napraviti *input* elemente koji imaju definisane *name*, *id* attribute i *value*.
3. Napraviti taster *Submit* za slanje podataka na php stranu, koja je definisana u atributu *action*
4. Napraviti pomenutu PHP stranicu u kojoj je potrebno prihvatiti podatke iz web forme
5. Koristeći superglobalne promenljive, `$_GET`, **`$_POST`** ili `$_REQUEST`, prihvatiti podatke
6. NPR. **`$x= $_POST[' atribut_name ']`** podatak iz input elementa HTML forme, koji ima `name="atribut_name"` je smešten u PHP stranicu u promenljivu `$x`.



# ELEMENTI ZA UNOS PODATAKA - INPUT

- ▶ Elementi za interakciju korisnika sa veb aplikacijom se zovu input elementi i imaju prost tag `<input />`
- ▶ Tipovi input elemenata su
  - ▶ text – tekstualno polje jedan red
    - ▶ number
    - ▶ date
    - ▶ time
    - ▶ mail
    - ▶ password
  - ▶ checkbox
  - ▶ radio
  - ▶ submit
  - ▶ button – od (HTML5 ima svoj složeni tag `<button>`)
  - ▶ file



# ELEMENTI ZA UNOS PODATAKA

- ▶ select – padajuća lista
  - ▶ option - podelementi
- ▶ textarea – tekstualno polje više redova



# ELEMENTI ZA UNOS PODATAK

```
<input type="text" name="tekst" id="tekst" /> Tekstualno polje<br><br>
<input type="radio" value="1" name="radio" id="radio" />Radio button<br><br>
<input type="checkbox" value="1" name="check" id="check" />Checkbox<br><br>
<input type="file" name="file" id="file" /><br><br>
<select name="select" id="select">
  <option value="0">Izaberite opciju</option>
  <option value="1">Opcija 1</option>
  <option value="2">Opcija 2</option>
</select><br><br>
<textarea name="textarea" id="textarea" cols="30" rows="10"></textarea><br><br>
<input type="button" value="Button"><br><br>
<input type="submit" value="Submit"><br><br>
```

PVA - Predavanje 4

Tekstualno polje

☐ Radio button

☐ Checkbox

No file selected.

Izaberite opciju ▼

# ČITANJE PODATAKA IZ HTML FORME

- ▶ Da bi se ovi podaci poslali putem HTTP protokola moraju biti smešteni u HTML element `<form>`
- ▶ Forma mora imati **submit** dugme koje služi za slanje podataka ili **button** ako je u pozadini JavaScript

# PRENOS IZ TEKSTUALNOG POLJA

- ▶ Prenos sadržaja tekstualnog polja u PHP **se vrši po atributu name**, a **prenosi se vrednost atributa value**
- ▶ Iz tog razloga treba voditi računa da ne bude preklapanja atributa po imenu (name)
- ▶ Sadržaj polja se prenosi u originalu, bez obzira na broj reči i razmake
- ▶ Preuzimanje u PHP kodu je pomoću:  
`$_POST['atribut_name']` ILI `$_GET['atribut_name']`

# PRENOS IZ TEKSTUALNOG POLJA

```
<html>
<head> </head>
<body>
<form action="prenos1.php" method="post">
  Unesite ime
  <input type="text" name="ime">
  <br/>
  <input type="submit" value="Unos">
</form>
</body>
</html>
```

localhost:8080/gabi/primer1.html x +

← → ↻ ⓘ localhost:8080/gabi/primer1.html

Unesite ime

```
<html>
<head> </head>
<body>
  Uneto ime je:
  <?php
    $ime = $_POST['ime'];
    echo ($ime);
  ?>
</body>
</html>
```

localhost:8080/gabi/prenos1.php x +

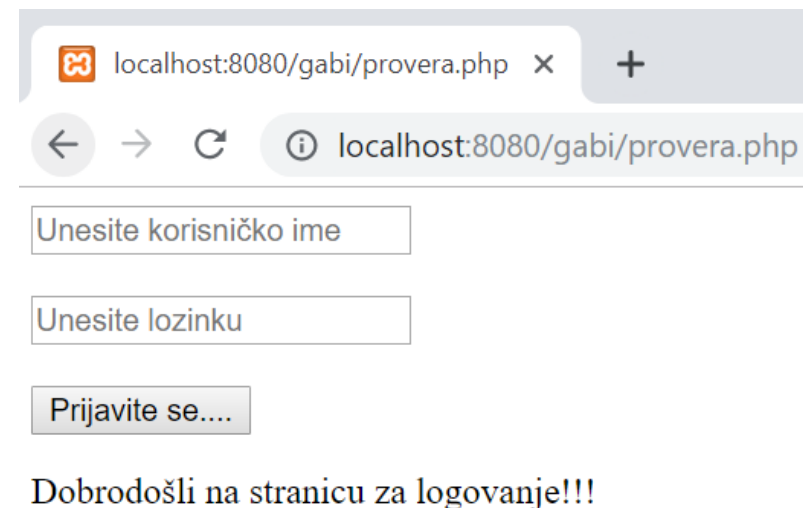
← → ↻ ⓘ localhost:8080/gabi/prenos1.php

Uneto ime je: Lazar

# KORISNIČKO IME I LOZINKA – TEXT, PASSWORD

Da bi se zaštitio prenos podataka van forme, koristi se funkcija `isset()`

```
<body>
<form action="index.php" method="post">
  <input type="text" name="username" id="username" placeholder="Unesite korisničko ime"/><br><br>
  <input type="password" name="password" id="password" placeholder="Unesite lozinku" /><br><br>
  <input type="submit" value="Prijavite se...." />
</form>
<?php
if(isset($_POST['username']) and isset($_POST['password']))
{
    $korime=$_POST['username'];
    $lozinka=$_POST['password'];
    if(!empty($korime) and !empty($lozinka))
    {
        echo "Dobrodošao, $korime<br>Vaša lozinka je: $lozinka<br>";
    }
    else
        echo "Niste uneli sve podatke!!!";
}
else
    echo "Dobrodošli na stranicu za logovanje!!!";
?>
</body>
```



# KORISNIČKO IME I LOZINKA - OPREZ

- ▶ Prijavljivanje korisnika na sajt može biti najranjivije mesto u veb aplikaciji.
- ▶ Jedino mesto gde korisnik šalje podatke a ne zna se ko je.
- ▶ Provera unetih podataka od strane korisnika prilikom logovanja je OBAVEZNA
- ▶ Može postojati i provera sa strane korisnika (JavaScript) i to će zaustaviti 99% slučajnih grešaka prilikom prijavljivanja
- ▶ Treba da se brinete za onih 1%
- ▶ Provera sa PHP strane je neophodna

# KORISNIČKO IME I LOZINKA – SQL INJECTION

- ▶ Korisnik umesto imena ili lozinke može ubaciti deo sql upita kojim bi pristupio bazi podataka bez da je ukucao validno ime i lozinku
- ▶ Osnovna zaštita bi trebala da bude provera da li korisničko ime i lozinka sadrže nedozvoljene karaktere koji se mogu naći u sql upitu („“, „=“, znake navoda....)

```
<?php
function validanString($string)
{
    if(strpos($string, "=") !== false) return false;
    if(strpos($string, "'" ) !== false) return false;
    if(strpos($string, " " ) !== false) return false;
    if(strpos($string, '"' ) !== false) return false;
    return true;
}
$lozinka=$_POST['password'];
$korime=$_POST['username'];
if(!validanString($korime) or !validanString($lozinka))
    echo "Korisničko ime ili lozinka sadrže nedozvoljene karaktere!!!";
else
    echo "Korisničko ime ili lozinka su ispravni";
?>
```



# PODACI IZ SELECT, CHECKBOX, RADIO, TEXTAREA

```
<form action="index.php" method="post">
  <select name="zemlja" id="zemlja">
    <option value="0" selected>--Izaberite zemlju--</option>
    <option value="Srbija">Srbija</option>
    <option value="Bosna i Hercegovina">BiH</option>
    <option value="Severna Makedonija">Severna Makedonija</option>
  </select><br><br>
  <input type="radio" name="pol" value="muški" checked />Muški<br>
  <input type="radio" name="pol" value="ženski" />Ženski<br><br>
  <textarea name="podaci" id="podaci" cols="30" rows="10">Napišite nešto o sebi....</textarea><br><br>
  <input type="checkbox" name="check" id="check" value="1">Da li se slažete sa uslovima
korišćenja?<br><br>
  <input type="submit" value="Snimate podatke" /><br><br>
</form>
```



# PODACI IZ RADIO TASTERA

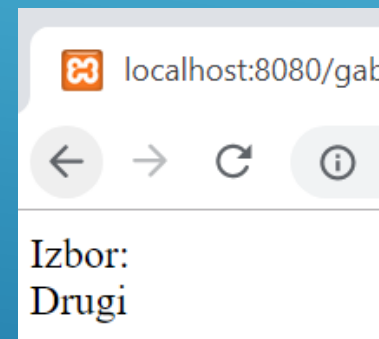
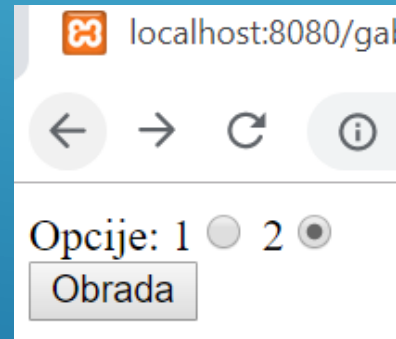
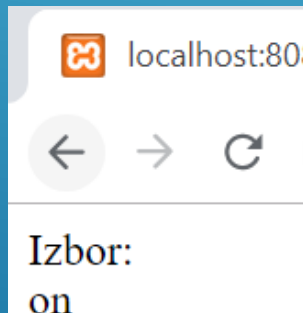
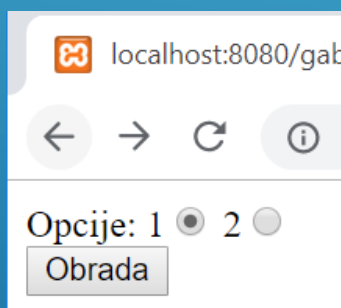
- Prenos sadržaja radio tastera u PHP se vrši po atributu name.
- Radio buttoni koji imaju istu vrednost u atributu name čine jednu grupu.
- Ako u tagu `<input />` nije definisan atribut value, tada je rezultat čekiranog polja u PHP-u *on*.
- Ako je atribut value definisan, prenosi se value vrednost



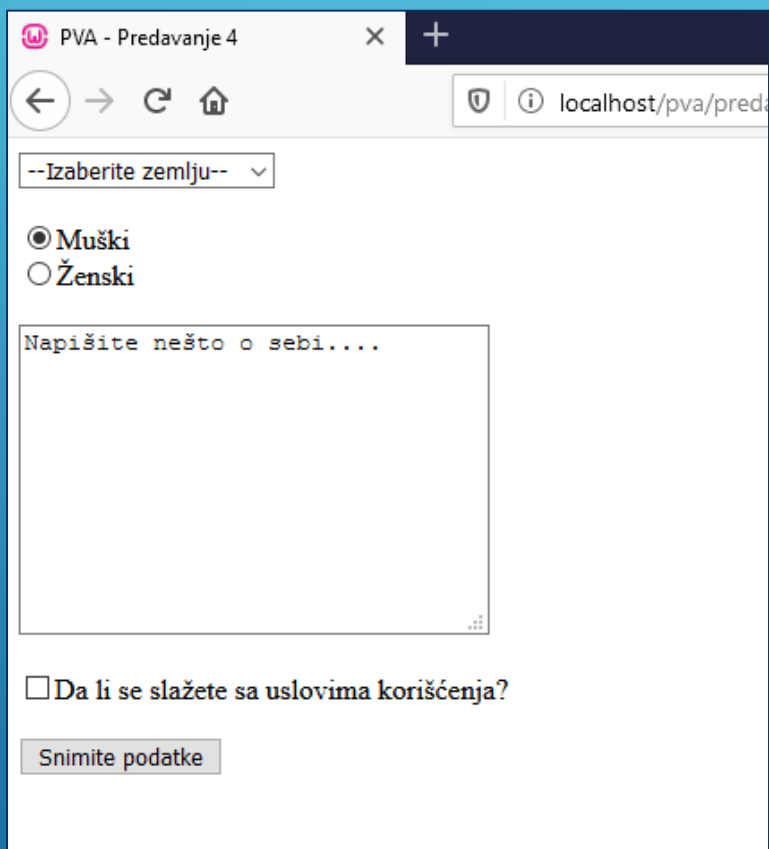
# PODACI IZ RADIO TASTERA

```
<html>
<head> </head>
<body>
<form action="prenos2.php" method="post">
Opcije:
1<input type="radio" name="1" >
2<input type="radio" name="1" value="Drugi" >
<br>
<input type="submit" value="Obrada">
</form>
</body>
</html>
```

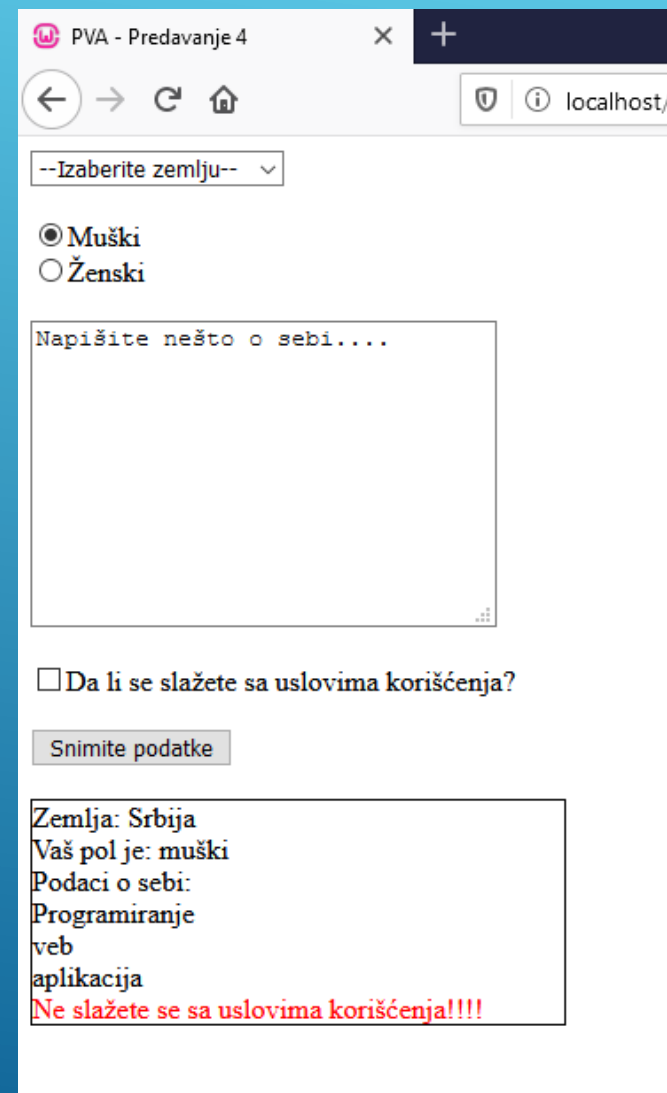
```
<html>
<body>
Izbor: <br>
<?php
echo $_POST['1'];
?>
</body>
</html>
```



# NEKI PODACI – SELECT, CHECKBOX, RADIO, TEXTAREA



A screenshot of a web browser window titled "PVA - Predavanje 4" showing a form at the URL "localhost/pva/preda". The form contains a dropdown menu labeled "--Izaberite zemlju--", two radio buttons for "Muški" (selected) and "Ženski", a large text area with the placeholder "Napišite nešto o sebi....", a checkbox for "Da li se slažete sa uslovima korišćenja?", and a "Snimate podatke" button.



A screenshot of the same web browser window after the form has been submitted. The dropdown menu now shows "Srbija", the radio button for "Muški" remains selected, and the text area now contains the text "Napišite nešto o sebi....". The checkbox for "Da li se slažete sa uslovima korišćenja?" is still unchecked. The "Snimate podatke" button is now disabled. A new box at the bottom of the form displays the submitted data: "Zemlja: Srbija", "Vaš pol je: muški", "Podaci o sebi: Programiranje veb aplikacija", and a red error message "Ne slažete se sa uslovima korišćenja!!!!".

# SELECT, CHECKBOX, RADIO, TEXTAREA

```
<?php
if(isset($_POST['zemlja']))
{
    echo '<div style="border:1px solid black; width: 300px">';
    //select
    $zemlja=$_POST['zemlja'];
    if($zemlja=="0")
        echo "Zemlja: Niste izabrali zemlju<br>";
    else
        echo "Zemlja: ".$zemlja."<br>";
    //pol
    $pol=$_POST['pol'];
    echo "Vaš pol je: ".$pol."<br>";
    //textarea
    $podaci=$_POST['podaci'];
    echo "Podaci o sebi:<br>";
    echo nl2br($podaci)."<br>";
    //checkbox
    if(isset($_POST['check']))
        echo "<font color='green'>Slažete se sa uslovima korišćenja!!!!</font>>";
    else
        echo "<font color='red'>Ne slažete se sa uslovima korišćenja!!!!</font>>";
    echo "</div>";
}
?>
```



# PREBACIVANJE DATOTEKA NA SERVER

- ▶ Za prebacivanje binarnih datoteka na server potrebno je:
  - ▶ HTML form objekat na HTML stranici u kome se nalazi input element sa atributom **type=„file“** pomoću koga ćemo izabrati datoteku za slanje i **enctype=„multipart/form-data“**
  - ▶ PHP kod koji će zaista prebaciti datoteku na server
- ▶ Sve može biti u jednoj .php stranici



# HTML DEO

- ▶ Podaci se moraju slati POST metodom i forma mora imati jedan dodatan parametar, a to je **enctype=„multipart/form-data“**
- ▶ Mora postojati dugme tipa **submit** ili tipa **button** ako u pozadini radi JavaScript

```
<form action="index.php" enctype="multipart/form-data" method="post">  
  <input type="file" name="datoteka" id="datoteka"><br><br>  
  <input type="submit" value="Klikni me....">  
</form>
```



# PHP DEO

- ▶ PHP će pročitati informacije iz file elementa
- ▶ Prebaciti u privremeni direktorijum na serveru sa privremenim imenom
- ▶ Dok se datoteka nalazi u privremenom direktorijumu možemo izvršiti sve potrebne provere
- ▶ Nakon provera datoteka se prebacuje u direktorijum gde treba da se nalazi koristeći **move\_uploaded\_file()** direktivu
- ▶ Prebacivanje datoteka (upload) je problematično mesto u veb aplikaciji. Mora se voditi računa šta korisnik prebacuje i da li je to ono što očekujemo. Provere treba da se vrše po tipu datoteke, veličini datoteke i da li postoji greška prilikom uploada

# PHP DEO – \$\_FILES

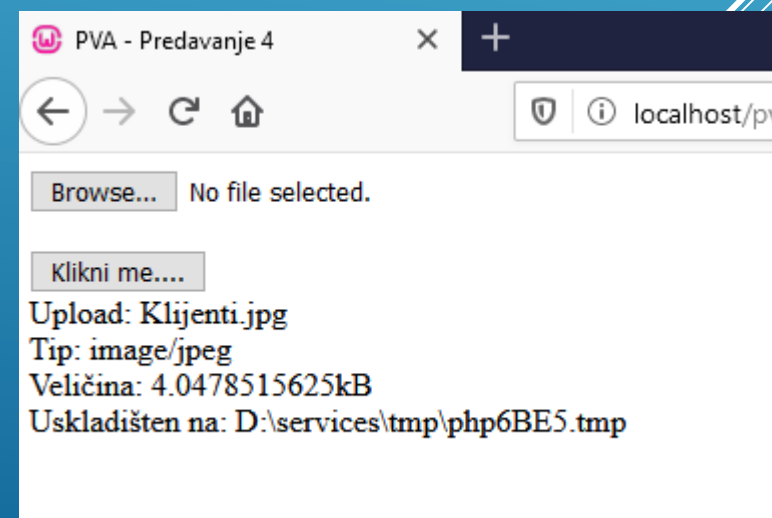
- ▶ Input element tipa „file“ nudi nekoliko parametara
  - ▶ name
  - ▶ type
  - ▶ tmp\_name
  - ▶ error
  - ▶ size
- ▶ Parametre error, size i type možemo koristiti za naše provere
- ▶ Prenosi se u globalnoj promenljivoj \$\_FILES



# PHP DEO – PRIKAZ PODATAKA

```
<?php
//Provera da li je upload prošao bez greške
if($_FILES['datoteka']['error']>0)
{
    echo "Greška prilikom prebacivanja datoteke!!!";
    exit();
}
//Ispis informacija o datoteci (samo upload u privremeni direktorijum)
echo "Upload: " . $_FILES["datoteka"]["name"] . "<br>";
echo "Tip: " . $_FILES["datoteka"]["type"] . "<br>";
echo "Veličina: " . ($_FILES["datoteka"]["size"] / 1024) . "kB<br>";
echo "Uskladišten na: " . $_FILES["datoteka"]["tmp_name"];

?>
```



# PHP DEO – UPLOAD BEZ PROVERE

- Ovo je najjednostavniji primer uploada, bez ikakve provere

```
<?php
    move_uploaded_file($_FILES['datoteka']['tmp_name'], $_FILES['datoteka']['name']);
?>
```



# PHP DEO –PROVERA

- Ovo je primer gde se proverava da li je datoteka slika, i da li je veličina slike manja od 100kB

```
<?php
//Provera da li je upload prošao bez greške
if($_FILES['datoteka']['error']>0)
{
    echo "Greška prilikom prebacivanja datoteke!!!";
    exit();
}
//Provera koji je tip datoteke
if ((($_FILES["datoteka"]["type"] == "image/gif") || ($_FILES["datoteka"]["type"] == "image/jpeg") ||
($_FILES["datoteka"]["type"] == "image/jpg") || ($_FILES["datoteka"]["type"] == "image/pjpeg") || ($_FILES["datoteka"]["type"]
== "image/x-png") || ($_FILES["datoteka"]["type"] == "image/png"))
{
    if($_FILES["datoteka"]["size"] < 100000)
    {
        if(!move_uploaded_file($_FILES['datoteka']['tmp_name'], $_FILES['datoteka']['name']))
            echo "Neuspelo prebacivanje datoteke na server";
        else
            echo "Uspešno prebacivanje datoteke na server";
    }
    else
        echo "Datoteka je prevelika";
}
else
    echo "Nije dobar tip datoteke";
?>
```



# UPLOAD - UPOZORENJE

- ▶ Ако postoji datoteka na serveru sa istim imenom, biće prebrisana bez upozorenja

