

# Php Core

Funkcije (User Defined Functions)

***ITAcademy***

---

# Šta je funkcija?

- Enkapsulirana, apstraktna, ponovno iskoristiva programska celina
- Postoji u gotovo svim programskim jezicima u različitim formama
- Prihvata i vraća parametre
- Definiše se samo jednom

# Tipovi funkcija u php-u

- Built in (preko 700)
- User defined

## Array Functions

array\_diff (arr1, arr2 ...)  
 array\_filter (arr, function)  
 array\_flip (arr)  
 array\_intersect (arr1, arr2 ...)  
 array\_merge (arr1, arr2 ...)  
 array\_pop (arr)  
 array\_push (arr, var1, var2 ...)  
 array\_reverse (arr)  
 array\_search (needle, arr)  
 array\_walk (arr, function)  
 count (count)  
 in\_array (needle, haystack)

## String Functions

crypt (str, salt)  
 explode (sep, str)  
 implode (glue, arr)  
 nl2br (str)  
 sprintf (frmt, args)

## Regular Expression Functions

ereg (pattern, str)  
 split (pattern, str)  
 ereg\_replace (pattern, replace, str)  
 preg\_grep (pattern, arr)  
 preg\_match (pattern, str)  
 preg\_match\_all (pattern, str, arr)  
 preg\_replace (pattern, replace, str)  
 preg\_split (pattern, str)

## Regular Expressions Syntax

^	Start of string
\$	End of string
.	Any single character
(a b)	a or b
(...)	Group section
[abc]	Item in range (a, b or c)
[^abc]	Not in range (not a, b or c)
\s	White space
a?	Zero or one of a

## Date Formatting

Y	4 digit year (2008)
y	2 digit year (08)
F	Long month (January)
M	Short month (Jan)
m	Month ^ (01 to 12)
n	Month (1 to 12)
D	Short day name (Mon)
l	Long day name (Monday) (lowercase L)
d	Day ^ (01 to 31)
j	Day (1 to 31)
h	12 Hour ^ (01 to 12)
g	12 Hour (1 to 12)
H	24 Hour ^ (00 to 23)
G	24 Hour (0 to 23)
i	Minutes ^ (00 to 59)
s	Seconds ^ (00 to 59)
w	Day of week ^ (0 to 6)
z	Day of year (0 to 365)

# Built in funkcije

- Sastavni deo programskog jezika
- Imaju sličnu sintaksu, funkciju i nazive u većini jezika
  - npr. Broj članova niza
    - php=count()
    - C#=count() (važi za kolekcije)

# “User Defined” Funkcije

- Stvaraju se za posebne namene
- Kreira ih korisnik

# Zašto koristiti funkcije?

- Enkapsulacija
- Portabilnost
- Brzina
- Parametrizacija
- Reusability

# Kada koristiti funkcije?

- Kada je potrebno ponavljanje iste sekvence koda, sa ili bez različitih vrednosti
- Prilikom raslojavanja projekta/programa

# Način funkcionisanja

- PHP funkcija može primiti BILO KOJI tip podatka
- PHP funkcija može vratiti BILO KOJI tip podatka
- PHP funkcije mogu se gnjezditi
- PHP funkcije mogu ući u rekurziju
- Promenljive definisane u funkciji važe samo u scope-u funkcije



# Konstrukcija

- Konstruisati sa ključnom rečju ***function***
- Naziv mora poštovati pravila koja poštuju i sve ostale varijable, osim što ne počinje sa znakom \$ i nije osetljiva na velika i mala slova
- Nakon naziva obavezno je otvoriti i zatvoriti malu zagradu sa ili bez parametara u njoj
- Telo funkcije mora biti u vitičastim zagradama

# Konstrukcija

```
function mojaFunkcija()  
{  
    //telo funkcije  
}
```

Ovo ne valja:

```
function x10(&$number)  
$number *= 10;
```

Greska: **Parse error:** syntax error, unexpected T\_VARIABLE, expecting '{' in

# Pozivanje

- Funkcija se može pozvati direktno ili putem dodeljivanja vrednosti
- Ukoliko funkcija ne vraća parametar, poziva se direktno
- Ukoliko funkcija vraća parametar, može se dodeliti kao vrednost
- Funkcija neće biti aktivirana ukoliko nije eksplicitno pozvana iz koda

## Primer: clock widget (pcex11 clockwidget)

- Potrebno je napraviti funkciju koja će prikazivati widget sa tačnim vremenom prilikom poziva
- Funkcija treba da se poziva na sledeci nacin:

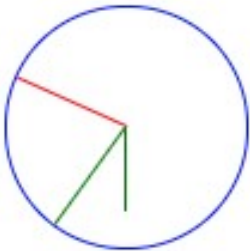
**showClockWidget()**

i da ima priblizno sledeci izlaz na strani:

17:59:37

## Izazov: analog clock widget (pcex11 anclockwidget)

- Pokušaj da prepraviš funkciju is prethodnog primera, tako da pokazuje analogni sat (poput sata prikazanog na slici ispod)



- Pomoć:**
- Ako ne znaš kako da postigneš rotaciju i kružni oblik kao na slici, pogledaj neke delove html-a pomoću kojih je napravljen sat sa slike:

```
<div style='position:relative;width:100px;height:100px;-webkit-border-radius:50px;-moz-border-radius:50px;border-radius:50px;border:1px solid blue;'>
```

```
  <div style='width:50px;height:1px;background-color:red;position:absolute;top:50px;left:50px;-ms-transform:rotate(204deg);-webkit-transform: rotate(204deg);transform: rotate(204deg);transform-origin: 0% 0%;'></div>
```

```
    <div style='width:50px;height:1px;background-color:green;position:absolute;top:50px;left:50px;-ms-transform:rotate(126deg);-webkit-transform: rotate(126deg);transform: rotate(126deg);transform-origin: 0% 0%;'></div>
```

```
      <div style='width:35px;height:1px;background-color:green;position:absolute;top:50px;left:50px;-ms-transform:rotate(50deg);-webkit-transform: rotate(450deg);transform: rotate(450deg);transform-origin: 0% 0%;'></div>
    </div>
```

# Zadatak: Login forma (pcex11 loginform)

- Potrebno je napraviti funkciju (proizvoljnog naziva) koja iscrtava login formu na strani
- Funkcija prikazuje dve kontrole za unos korisnickog imena i sifre i taster
- Forma ima pozadinsku sliku, koja je razlicita prilikom svakog otvaranja forme (bira se jedna od tri dostupne slike, po slucajnom izboru)



# Parametrizacija – ulazni parametri

```
function mojaFunkcija($a)  
{  
    echo $a;  
}
```

```
function mojaFunkcija($a,$b,$c)  
{  
    echo $a;  
    echo $b;  
    echo $c;  
}
```

## Parametrizacija – ulazni parametri

- Default vrednosti

```
function mojaFunkcija($a,$b=100)  
{  
echo $a;  
echo $b;  
}
```



# Parametrizacija – typehinting

- Typehinting je stroga tipizacija ulaznih parametara funkcije ili metode
- Ako želimo da nateramo korisnika da u funkciju ubaci baš parametar određenog tipa, možemo upotrebiti typehinting

```
<?php
function userInrole(array $roles, $userRole){
    return in_array($userRole,$roles);
}
$allowedRoles = [1,2,4,8];
$userValid = userInrole($allowedRoles,10);
echo "User is " . (!$userValid?"in":"" ) . "valid";
```

# Tipizacija rezultata funkcije

- U php funkcijama, od verzije 7, moguće je strogo tipizirati i rezultate funkcije

```
function add($a,$b):int {  
    return $a+$b;  
}  
echo add(2,3);
```

OK

```
function add($a,$b):int {  
    return "Hello World";  
}  
echo add(2,3);
```

**FATAL ERROR Uncaught  
TypeError: Return value of  
add() must be of the type  
integer, string returned ....**

# Tipizacija rezultata funkcije

- U php funkcijama, od verzije 7, moguće je strogo tipizirati i rezultate funkcije

```
function add($a,$b):int {  
    return $a+$b;  
}  
echo add(2,3);
```

OK

```
function add($a,$b):int {  
    return "Hello World";  
}  
echo add(2,3);
```

**FATAL ERROR Uncaught  
TypeError: Return value of  
add() must be of the type  
integer, string returned ....**

## Vežba: widget (pcex11 newswidget)

- Kreiraj funkciju koja prikazuje widget na zadatom mestu na strani
- Funkcija kao parametar prihvata niz sa vestima, ali prikazuje samo jednu vest po slučajnom izboru

### **La Liga: Ten-man Getafe stun Valencia**

Markel Bergara fired 10-man Getafe to a shock victory over second-placed Valencia to leave Barcelona five points clear at the top.

## Vežba: odds1 (pcex11 oddsparser)

- U aplikaciju stižu hendikep kvote u vidu asocijativnog niza
- Kvote treba prikazati na isti način kao na slici desno
- Kreiraj funkciju koja će prepakovati niz sa kvotama tako da se jednostavno može prikazati u traženom formatu

### Manchester United - Liverpool

HC	1	2
2.5	7.5	1.6
3.0	0.7	0.6
3.5	0.6	4.1
4.0	8.1	1.5

### New Castle - Leeds

HC	1	2
3.15	0.3	8
3.20	9.8	8.9
3.30	0.9	9.9
3.40	1.9	5.7

# Vežba: odds2 (pcex11 oddsparser1)

- U aplikaciju stižu winner kvote u vidu višedimenzionalnog indeksiranog niza
- Kvote treba prikazati na isti način kao na slici desno
- Treba kreirati funkciju koja će kvote iz izvornog niza, prepakovati u novi niz sa prihvatljivijom formom I drugu funkciju, koja će podatke iz prepakovanog niza, prikazati na traženi način
- Format izvornog niza  
`[["Tim1 - Tim2","Tim3 - Tim4"],...,  
[["2.85","3.30","2.40",true],  
[["1.65","4.00","4.60",true],...]`
- Format ciljnog niza  
`[["match"=>"FC Gillingham - Oxford United","odds"=>[2.85,3.30,2.40,1],...]`

## FC Gillingham - Oxford United

Winner	1	0	2
	2.85	3.30	2.40

## Hapoel R. Letzion - Maccabi Herzliya

Winner	1	0	2
	1.65	4.00	4.70

## Roter Stern - 1. FC Köln

Winner	1	0	2
	2.20	3.20	3.25

## KRC Genk - AS Eupen

Winner	1	0	2
	1.30	5.20	9.20

# Vežba: vote (pcex11 vote)

- U aplikaciji se nalazi niz sa pitanjima. Ovaj niz reprezentovan je na sledeci nacin:

```
$questions = [  
    "1"=>[  
        "question"=>"Who is best guitarist?",  
        "answers"=>["Eric Clapton","Jimi Hendirx","Mike Oldfield"]  
    ],  
    "5"=>[  
        .....  
    ]  
]
```

- Napravi funkciju **vote** koja ce na osnovu niza generisati pitanje za anketu, po slucajnom izboru, tako da krajnji izlaz bude sledeci:  

Who is best guitarist?

  - [Eric Clapton](#)
  - [Jimi Hendirx](#)
  - [Mike Oldfield](#)
- Pokusaj da, pomocu funkcije **save**, rezultate ankete snimas u fajl (ako znas da se podaci mogu snimati u fajl na sledeci nacin: **file\_put\_contents("answers.txt", "SOME CONTENT",FILE\_APPEND);**

# Automatsko preuzimanje parametara

Funkcija ***func\_get\_args()*** vraća niz svih parametara koji su prosleđeni u funkciju

```
<?php
```

```
function f() {
```

```
    print_r(func_get_args());
```

```
}
```

```
f("Belgrade","Paris","London","Juneau");
```

```
?>
```



# Automatsko preuzimanje parametara

Funkcije `func_num_args()` vraća broj prosleđenih parametara  
a funkcija `func_get_arg()` vraća pojedinačan parametar po  
zadatom indeksu

```
<?php
```

```
function f() {
```

```
    echo "U funkciju je uslo: " . func_num_args() . "  
    paramet(a)ra<br>";
```

```
    for($i=0;$i<func_num_args();$i++){
```

```
        echo $i+1 . ": " . func_get_arg($i) . "<br>";
```

```
    }
```

```
}
```

```
f("Belgrade","Paris","London","Juneau");
```

```
?>
```

## Prosleđivanje reference

```
<?php  
function byRef(&$apples)  
{ $apples++; }  
    $oranges = 5;  
    $apples = 5;  
    byRef($oranges);  
    echo "I have $apples apples  
    and $oranges oranges";  
?>
```

# Parametrizacija – izlazni parametri

Izlazni parametri (povratna vrednost) se prosleđuju naredbom `return`. Nakon izvršenja naredbe `return`, izvršenje funkcije se momentalno prekida

```
function mojaFunkcija()  
{
```

```
    return 1;
```

```
}
```

```
function mojaFunkcija()  
{
```

```
    $a=1;
```

```
    return $a;
```

```
}
```

# Parametrizacija - pozivanje

- Bez parametara

***mojaFunkcija();***

- Sa parametrom (ili parametrima)

***mojaFunkcija(5);***

***mojaFunkcija(4,5,6);***

# Parametrizacija – preuzimanje rezultata

- Ispisivanje

***echo mojaFunkcija();***

***echo mojaFunkcija(5);***

- Dodeljivanje

***\$a=mojaFunkcija();***

***\$a=mojaFunkcija(5);***

# Gnježđenje

```
function spoljnaFunkcija()  
{  
    function untrasnjaFunkcija()  
        {  
            echo "unutra";  
        }  
    echo "spolja";  
}
```

Obe funkcije se vide na globalnom nivou

# Opseg vidljivosti promenljivih

- Promenljive u deklarisanu u funkciji vide se samo u funkciji
- Promenljive deklarisanu van funkcije, vide se i u funkciji samo ako su unutar funkcije ponovo deklarisanu ključnom rečju `global`
  - `global $a;`

# Opseg vidljivosti promenljivih

Sledeći primer ne prikazuje ništa:

```
<?php
function f(){
    $x = 10;
}
f();
echo $x;
?>
```

Sledeći primer prikazuje broj 10:

```
<?php
function f(){
    global $x;
    $x = 10;
}
f();
echo $x;
?>
```

Sledeći primer ne prikazuje ništa:

```
<?php
$x = 10;
function f(){
    echo $x;
}
f();
?>
```

Sledeći primer prikazuje broj 10:

```
<?php
$x = 10;
function f(){
    global $x;
    echo $x;
}
f();
?>
```



# Rekurzija

Rekurzija je slučaj u kome jedna funkcija poziva samu sebe

```
function mojaFunkcija()  
{  
    mojaFunkcija();  
}
```

# Praćenje rekurzije

```
function mojaFunkcija($parametar)  
{  
    if($parametar<10)  
        {  
            $parametar++;  
            mojaFunkcija($parametar);  
        }  
    else  
        echo "kraj izvršavanja";  
}
```

# Dinamičko kreiranje funkcija

- Funkcije koje se kreiraju dinamički (in line)

```
$f = create_function('$x,$y', 'return $x + $y;');  
echo $f(2, 3);
```

```
function f() { echo "Hello!"; }  
$x = f;  
$x();
```

# Generisanje HTML-a u funkciji

```
function tb($id,$v)
```

```
{
```

```
echo "<input type=\"text\" value=\"\$v\"  
    id=\"\$id\" />";
```

```
}
```

```
tb(5,"moja vrednost");
```

# Funkcije u klasama

- Funkcije imaju posebnu ulogu u klasama
- Funkcije, sastavni delovi klase, nazivaju se **metodi ili metode**
- Funkcija (ili funkcije) koja učestvuje u kreaciji klase (ona koja se automatski poziva kada se kreira objekat klase) naziva se **konstruktor**

# Zadatak 1

- Napraviti funkciju koja vrši računsku operaciju (sabiranje, oduzimanje, deljenje ili množenje) na osnovu tri parametra, operanada i operatora

```
<?php
$op = "+";
$a = 2;
$b = 3;
function calc($op,$a,$b)
{
    switch($op)
    {
        case "+":
            return $a + $b;
            break;
        case "-":
            return $a - $b;
            break;
        case "*":
            return $a * $b;
            break;
        case "/":
            return $a / $b;
            break;
    }
}
echo calc("/",15,3);
?>
```

## Zadatak 2

- Kreirati funkciju koja iscrtava tag prema zadatim parametrima.
- Parametri su (respektivno): naziv taga, stil, sadržaj taga i broj ponavljanja. Ukoliko se ne unese broj ponavljanja, tag se mora iscrtati bar jednom.
- Primer:

*drawTags("div", "border: 1px solid red; background-*

Hello from me!

Hello from me!

Hello from me!



```
<?php
```

```
function drawTags($tname,$style,$content,$count=1)
{
    for($i=0;$i<$count;$i++)
        echo "<" . $tname . " style='" . $style . "' >" . $content
        . "</" . $tname . ">";
}
```

```
drawTags("div","border: 1px solid red; background-color:
yellow; color: red; margin: 4px; padding: 4px;","Hello
from me!",200);
```

```
?>
```

## Zadatak 3

- Kreirati funkciju koja iscrtava select kontrolu na osnovu sledećih parametara
  - Name
  - Id
  - Asocijativni niz stavki (za value i display vrednosti)
  - Odabrana stavka
  - Stil

# Rešenje 3

```
<?php
function drawSelect($name,$id,$arr,$sel,$style)
{
    echo "<select name='$name' id='$id' style='$style' >";
    foreach($arr as $k=>$v){
        echo "<option ";
        echo "value='$k' ";
        echo ($sel==$k)?"selected='selected' ":"";
        echo ">";
        echo $v;
        echo "</option>";
    }
    echo "</select>";
}
$arr = array("1"=>"Belgrade","2"=>"Paris","3"=>"London","4"=>"Juneau");
drawSelect("mySelect","mySelect",$arr,"3","padding: 2px; border: 1px solid #d3d3d3;");
?>
```