



Вежба (Passing by argument)

1. Напиши функција со име **aboutMe**.
2. Функцијата прима 2 параметри **\$name** и **\$age**.
3. Функцијата треба да печати **"Hello! My name is \$name, and I am \$age years old."**.
4. Откако ќе ја иницијализираш функцијата треба да ја повикаш за да видиш како работи.



Решение

```
function aboutMe ($name, $age)
{
    return print "Hello! My name is " . $name . " and I
am " . $age . " years old.";
}

aboutMe ('Dejan', '26');
```

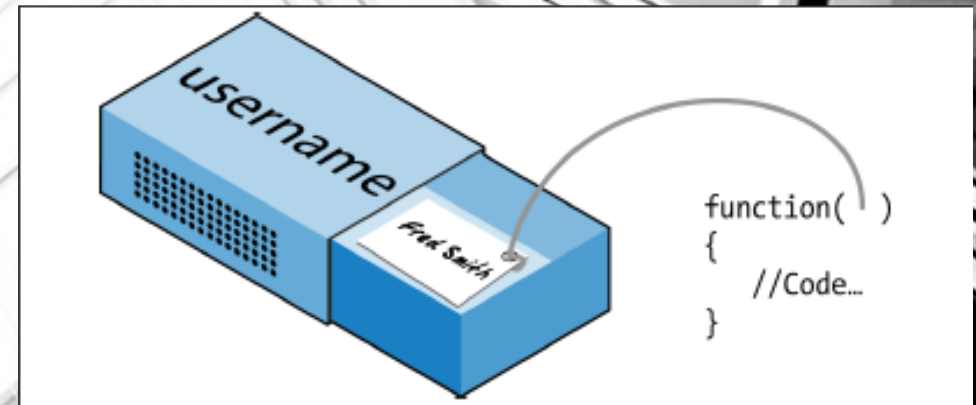


Passing by Reference

& симбол кој стои пред променливата покажува да се пренесе **reference** до вредноста на променливата, а не вредноста.

Со други зборови, наместо земање на лист од фиока - Копирајте го истиот на друг лист, додека оригиналот останува во фиоката.

Копијата се користи во функцијата





Пример

```
function fix_names(&$n1, &$n2, &$n3)
{
    $n1 = ucfirst(strtolower($n1)) ;
    $n2 = ucfirst(strtolower($n2)) ;
    $n3 = ucfirst(strtolower($n3)) ;
}
```

```
$a1 = "WILLIAM";
$a2 = "henry";
$a3 = "gatES";
```

```
echo $a1 . " " . $a2 . " " . $a3 . "<br>;
```

```
fix_names($a1, $a2, $a3);
```

```
echo $a1 . " " . $a2 . " " . $a3;
```




Passing By Reference

- ❑ Функцијата **fix_names** само треба да назначи нови вредности на променливите : **\$n1**, **\$n2**, и **\$n3**
- ❑ И да направи update на променливите **\$a1**, **\$a2**, и **\$a3**.

ИЗЛЕЗ

WILLIAM henry gatES
William Henry Gates

- ❑ Во кодот се забележува дека **echo** командите ги користат само променливите **\$a1**, **\$a2**, и **\$a3**.
- ❑ Биди внимателен со пренесување на променливите со референца.
- ❑ Ако ти требаат оригиналните вредности, задолжително прави копија од истите.



ФУНКЦИЈАТА **function_exists**

Ако има потреба да се провери дали определена функција е достапна за твојот код , тогаш се користи функцијата **function_exists()**, која ги проверува сите претходно дефинирани функции и оние креирани од корисникот.

```
if (function_exists("array_combine"))  
{  
    echo "Function exists";  
}  
else  
{  
    echo "Function does not exist -  
better write your own";  
}
```



Потсетување

PHP **Syntax**

PHP **Variables**

PHP **Echo / Print**

PHP **Data Types**

PHP **Arrays & Strings**

PHP **Constants**

PHP **Operators**

PHP **If...Elseif...Else**

PHP **Switch**

PHP **While Loops**

PHP **For Loops**

PHP **Functions**



PHP Arrays

- ❑ Претходно накратко се запознавме со низите од броеви, карактери итн. – доволно за да почувствуваме како овие елементи функционираат.
- ❑ Следува подетален опис и по темелно објаснување на употребата на PHP низи/arrays.
- ❑ Ако сте ги користеле во друг програмски јазик, на пример како C, ќе бидете пријатно изненадени од едноставноста и класичноста на PHP arrays.



PHP Arrays - Синтакса

```
$array[] = "Dejan"; или $array = array("Dejan");
```

- ❑ Низата е листа од елементи. Дозволува да зачувате повеќе од еден елемент во само една променлива.

```
$array = array("Egg", "Tomato", "Beans");
```

- ❑ `array()` – Показува дека варијаблата/променливата `$array` е всушност низа која може да содржи повеќе елементи
- ❑ Кога додаваме низа од карактери користиме наводници " "



PHP Arrays

- ❑ Кога додаваме броеви, ја пишуваме само вредноста
- ❑ Како и да е, секој член на низата мора да биде одделен со запирка.

Типови на низи

- **Indexed** Arrays
- **Associative** Arrays
- **Multidimensional** Arrays



Indexed Arrays

ПРИМЕР: Додавање членови на низа и печатење на низата

```
$paper[] = "Copier";  
$paper[] = "Inkjet";  
$paper[] = "Laser";  
$paper[] = "Photo";
```

```
for ($j = 0; $j <= 3; ++$j) {  
    echo "$j: $paper[$j]<br>";  
}
```

ИЗЛЕЗ

**0: Copier
1: Inkjet
2: Laser
3: Photo**



Пример

Пристап до низата преку нејзините индекси.

```
$myArray = array("do", "re", "mi");  
echo $myArray[0] ; // печати do  
echo $myArray[2] ; // печати mi
```




Associative Arrays

- Понекогаш е тешко да запомнеме кој е индексот на елементот кој ни е потребно да го повикаме. Пример во кој се користат **associative arrays**.

```
$paper['copier'] = "Copier & Multipurpose";  
$paper['inkjet'] = "Inkjet Printer";
```

- Имињата (**copier** и **inkjet**) се индекси (***indexes*** или ***keys***)
- Елементите кои ги посочуваат (како на примерот **Copier & Multipurpose** и **Inkjet Printer**) се наречени вредности (***values***).



Associative Arrays

Пример

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

или

```
$age['Peter'] = "35";
```

```
$age['Ben'] = "37";
```

```
$age['Joe'] = "43";
```

Печати

```
echo "Peter is " . $age['Peter'] . " years old.";
```



Loop низ **Associative** Arrays

Пример:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
  
foreach($age as $key => $value) {  
    echo "Key=" . $key . ", Value=" . $value;  
    echo "<br>";  
}
```



Multidimensional Arrays

Дво-димензионална низа е низа составена од повеќе други низи (три-димензионална низа е низа составена од низи кои се составени од повеќе низи).

Во селдните неколку слајдови ќе работиме со дво-димензионална низа која содржи 4 други низи (**arrays**).

Обрнете внимание на следниот слајд каде во табела ќе ги видиме прикажани елементите на понатамошните низи подредени во редови и колони.



Multidimensional Arrays

Name	Stock	Sold
Volvo	22	18
BMW	15	13
Saab	5	2
Land Rover	17	15

Од тука, податоците од табелата можеме да ги зачуваме во дво-димензионална низа на следниов начин:



Multidimensional Arrays

```
$cars = array  
(  
    array ("Volvo",      22,      18) ,  
    array ("BMW",        15,      13) ,  
    array ("Saab",        5,       2) ,  
    array ("Land Rover", 17,      15)  
) ;
```

Од примерот погоре може јасно да се види дека низата **\$cars** содржи 4 други низи. За да им пристапиме на елементите од низата **\$cars** , мора да го почитуваме нивниот редослед по **редови** и **колони**.



Multidimensional Arrays

Пример

```
echo $cars[0][0].": In stock: ".$cars[0][1].", sold:  
".$cars[0][2].".<br>;  
echo $cars[1][0].": In stock: ".$cars[1][1].", sold:  
".$cars[1][2].".<br>;  
echo $cars[2][0].": In stock: ".$cars[2][1].", sold:  
".$cars[2][2].".<br>;  
echo $cars[3][0].": In stock: ".$cars[3][1].", sold:  
".$cars[3][2].".<br>;
```



Multidimensional Arrays

Исто така може да се стави **for loop** во друг **for loop** за да ги пристапиме елементите од низата **\$cars** (притоа сеуште мора да се придржуваме на нејзините индикатори од претходно)

```
for ($row = 0; $row < 4; $row++) {  
    echo "<p><b>Row number $row</b></p>";  
    echo "<ul>";  
    for ($col = 0; $col < 3; $col++) {  
        echo "<li>".$cars[$row][$col]."</li>";  
    }  
    echo "</ul>";  
}
```