

# APLIKACJE INTERNETOWE PHP



**TEMAT 3-06:** Zaawansowane tworzenie własnych funkcji.

Autor dokumentu: Wojciech Galiński

środa, 7 września 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>,  
<http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

## Zagadnienia obowiązkowe

1. **Przekazywanie wartości do funkcji przez referencję** – to przekazanie do funkcji oryginalnej wartości, a nie jej kopii.

PRZYKŁAD:

```
function zwieksz_o_1(&$liczba) { $liczba += 1; echo $liczba; }  
$k=1; echo $k; zwieksz_o_1($k); echo ' '. $k;
```

Bardzo zbliżoną funkcjonalność ma poniższy przypadek. Wymaga on jednak więcej pamięci, ponieważ do funkcji przekazywana jest kopia zmiennej.

PRZYKŁAD:

```
function zwieksz_o_1($liczba) { return $liczba + 1; }  
$k=1; echo $k; $k = zwieksz_o_1($k); echo ' '. $k;
```

2. **Funkcje zwracające referencję** – stosuje się je przede wszystkim w programowaniu obiektowym, ale praktyczne zastosowanie widać także w poniższych przykładach:

```
function &druga_nazwa_zmiennej($nazwa) {  
    global $$nazwa;  
    // Od teraz globalna zmienna  
    // dostępna jest wewnątrz funkcji  
    return $$nazwa;  
}  
  
$a = 3;  
$b = &druga_nazwa_zmiennej('a');  
echo $a. ' '. $b. '<br />';  
++$a; echo $a. ' '. $b. '<br />';  
++$b; echo $a. ' '. $b. '<br />';  
  
function &zwieksz_zmienna_statyczna()  
{  
    static $licznik = 0;  
    $licznik += rand(1,9);  
    return $licznik;  
}  
  
// zwraca referencję do zmiennej  
$podglad = &zwieksz_zmienna_statyczna();  
for ($i=0; $i<3; ++$i)  
{  
    echo $podglad. ' ';  
    zwieksz_zmienna_statyczna();  
}
```

Pierwszy przykład pokazuje, jak odwoływać się do tej samej komórki pamięci za pomocą dwóch nazw.

Drugi przykład pokazuje, jak używać zmiennych statycznych, które mogą być tworzone lokalnie, ale ich wartości nie są usuwane po zakończeniu funkcji. Dzięki temu we wszystkich wywołaniach funkcji w skrypcie odwołujemy się do tej samej zmiennej statycznej.

3. **Funkcje zmiennych** – zmienna z nawiasami to jak funkcja o nazwie zapisanej w tej zmiennej.

PRZYKŁAD:

```
function reklama_1($powitanie)  
{return $powitanie. 'Kupuj w czerwonym sklepie!<style>div{background: #fbb;</style>';}  
function reklama_2()  
{ return 'Kupuj w zielonym sklepie!<style>div{background: #bfb;}</style>'; }  
function reklama_3($pozdrawienie)  
{ return 'Kupuj w niebieskim sklepie!<style>div{background: #bbf;}</style>'.  
    $pozdrawienie; }  
$f = 'reklama_'. rand(1, 3);  
echo '<div style="padding:5px; text-align:center;">'.$f(' Hej! '). '</div>';
```

4. **Funkcje rekurencyjne** – to funkcje, które wywołują same siebie. Może to prowadzić do bardzo ciekawych wyników. Najpraktyczniejszym przykładem jest chyba wyświetlenie struktury katalogów i plików na dysku.

PRZYKŁAD:

```
function ls($sciezka, $wyswietlaj_tylko_katalogi = true)
{ if (!is_readable($sciezka)) { echo ' (brak dostępu)'; return; }
  $katalog = opendir($sciezka);  echo '<ol>';

  if ($katalog)
    while ($plik = readdir($katalog))
    { if ($plik=='.' || $plik=='..') ;
      elseif (is_dir($sciezka. '/' . $plik))
      { echo '<li><b>'. $plik. '</b>';
        ls($sciezka. '/' . $plik, $wyswietlaj_tylko_katalogi);  echo '</li>';
      }
      else if (!$wyswietlaj_tylko_katalogi) echo '<li>'. $plik. '</li>';
    }
    echo '</ol>';
    closedir($katalog);
}
ls('/boot', true);
```

5. (\*) **Funkcje o zmiennej ilości argumentów** – do obsługi tego typu funkcji wykorzystujemy następujące funkcje:

- ➔ **func\_num\_args()** – pobiera liczbę przekazanych argumentów;
- ➔ **func\_get\_arg(numer\_argumentu)** – pobiera wartość wybranego argumentu;
- ➔ **func\_get\_args()** – pobiera wartości wszystkich argumentów i zapisuje je w tablicy.

PRZYKŁAD:

```
function test_argumentow()
{
  $liczba_argumentow = func_num_args();
  echo "Liczba argumentów funkcji to $liczba_argumentow";
  if ($liczba_argumentow>=1)
    echo ', a pierwszym argumentem jest '. func_get_arg(0);
  echo '<br />';
  if ($liczba_argumentow>=2)
  {
    echo 'Następne argumenty to:<br />';
    $lista_argumentow = func_get_args();
    for ($i=1; $i<$liczba_argumentow; ++$i)
      echo "Argument nr $i = ". $lista_argumentow[$i]. '<br />';
  }
}
test_argumentow(1, 2, 3);
```

## Zadania

1. Zapoznaj się z przedstawionymi przykładami. Uruchom je we własnych skryptach i przeanalizuj.
2. Do funkcji „ls” dodaj wyświetlanie nazw plików (kontroluj wyświetlanie plików za pomocą drugiego parametru funkcji).
3. Do funkcji „ls” dodaj trzeci parametr informujący o maksymalnej liczbie wywołań rekurencyjnych, np. 5 oznacza, że funkcja będzie pokazywać maksymalnie do 5 poziomów w drzewie katalogów.
4. Napisz funkcję, która będzie liczyć medianę wszystkich argumentów. Jeśli któryś z argumentów nie będzie liczbą, należy zgłosić błąd.