

APLIKACJE INTERNETOWE PHP



TEMAT 3-04: Funkcje obsługi tablic.

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. Operator „+” dla tablic – służy do łączenia tablic, np.

```
$t1 = array(1=>'a'); $t2 = array(1=>'tekst'); $wynik = $t1 + $t2;
```

Jest on podobny do funkcji „array_merge”, ale są różnice w ich działaniu, np.

```
$t1 = array(1=>'a'); $t2 = array(1=>'tekst'); $wynik = array_merge($t1, $t2);
```

2. Sortowanie i mieszanie tablic – więcej: <http://php.net/manual/pl/array.sorting.php>.

W tej kategorii wyróżniamy następujące funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
sort, rsort; asort, arsort	Sortowanie tablicy bez zachowania / z zachowaniem kluczy z porządkowaniem rosnącym / malejącym .	<pre>\$t=array('c',2,'a',8); sort(\$t); foreach (\$t as \$k=>\$w) echo 't['.\$k.']='. \$w.'
';</pre>	array_reverse	Odwraca kolejność elementów w tablicy.	<pre>\$t=array('c'=>1,'a'=>2, 'd'=>0,'b'=>4); \$t = array_reverse(\$t); print_r(\$t); \$t=array_reverse(\$t,true); print_r(\$t);</pre>
ksort, krsort	Sortowanie tablicy według kluczy.	<pre>\$t=array('c'=>1,'a'=>2, 'd'=>0,'b'=>4);</pre>	shuffle	Miesza elementy tablicy.	<pre>\$t = array(1,2,3,4,5,6,7); shuffle(\$t); print_r(\$t);</pre>
usort, uasort; uksort	Sortuje tablicę przy pomocy zdefiniowanej funkcji porównującej (bez / z zachowaniem kluczy albo sortowanie według kluczy).	<pre>function cmp(\$a,\$b) { if (\$a==>\$b) return 0; return (\$a<\$b)? -1: 1; } \$t = array(3,2,5,6,1); usort(\$t, 'cmp'); var_dump(\$t);</pre>	array_rand	Wybiera jeden lub więcej losowych elementów z tablicy (zwraca je uporządkowane).	<pre>\$t = array(1,2,3,4,5,6,7); \$los = array_rand(\$t); print_r(\$los); \$los = array_rand(\$t, 3); print_r(\$los);</pre>
			array_multisort	Sortuje wiele tablic lub tablice wielowymiarowe.	<pre>\$t_in = array(2,3,1,0,1); \$t = array(\$t_in, \$t_in); \$t = array_multisort(\$t); print_r(\$t);</pre>

3. Funkcje tablic – więcej: <http://php.net/manual/pl/ref.array.php>. Podzieliłem je następująco:

➔ **podstawowe funkcje operujące na tablicach** – funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
array	Tworzy nową tablicę.	<pre>\$tab=array(1, 2, 'abc', 'imie'=>'Ewa');</pre>	count	Zlicza ilość elementów w tablicy lub pól obiektu	<pre>echo count(\$t);</pre>
range	Tworzy tablicę zawierającą przedział elementów.	<pre>print_r(range(1,10)); print_r(range(1,9,2));</pre>	array_sum, array_product	Oblicza sumę/iloczyn wartości w tablicy.	<pre>\$t=array(1,1,2,3,4,5,5); echo array_sum(\$t). ' '. array_product(\$t);</pre>
in_array, array_key_exists Sprawdza, czy wartość / klucz (indeks) istnieje w tablicy.		<pre>\$t = array(1, 2, 'a'=>'b'); \$b = in_array(\$t,1); var_dump(\$b); \$b=array_key_exists(\$t,1); var_dump(\$b);</pre>	array_keys, array_values	Zwraca wszystkie klucze/wartości z tablicy.	<pre>\$t=array(1,2,3,'a'=>4); \$t = array_values(\$t); print_r(\$t); \$t = array_keys(\$t); print_r(\$t);</pre>
			array_unique	Usuwa duplikaty wartości z tablicy	<pre>\$t=array(2,1,1,4,'2',3); \$t = array_unique(\$t); print_r(\$t);</pre>

➔ **funkcje wyszukujące oraz zamieniające dane w tablicy** – funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
array_search Przeszukuje tablicę pod kątem podanej wartości i w przypadku sukcesu zwraca klucz do jej pierwszego wystąpienia		<pre>\$t=array(1,2,3,4); echo array_search(1,\$t); echo array_search(3,\$t);</pre>	array_splice	Wycina z tablicy jej fragment i wstawia w to miejsce coś innego.	<pre>\$t=array(1,2,3,4,5,6); \$t=array_splice(\$t,-2); print_r(\$t); \$t=array_splice(\$t,1,3); print_r(\$t); \$t=array_splice(\$t,1,3,0); print_r(\$t);</pre>
array_replace, array_replace_recursive Podmienia wybrane elementy jednej tablicy. docelowej na elementy tablic źródłowych		<pre>\$t=array('a','b','c'); \$str1=array(0=>'d',2=>'e'); \$str2=array(2=>'f'); \$t=array_replace(\$t, \$str1,\$str2); print_r(\$t);</pre>	array_slice	Zwraca wycięty kawałek tablicy.	<pre>\$t=array(1,2,3,4,5,6); \$t2=array_slice(\$t,2,-1); \$t3=array_slice(\$t,2,-1,true); print_r(\$t2); print_r(\$t3);</pre>

➔ funkcje pracujące na wskaźniku bieżącego elementu tablicy – funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
reset, end; prev, next	Przesuwa wewnętrzny wskaźnik tablicy na pierwszy/ostatni/poprzedni/następny element.	<code>\$t=array(0,2,4,6,8);</code> <code>echo next(\$t). next(\$t);</code> <code>echo prev(\$t);</code> <code>echo end(\$t);</code> <code>echo reset(\$t);</code>	key; current, pos	Zwraca klucz/wartość bieżącego elementu tablicy.	<code>\$t=array('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4);</code> <code>echo current(\$t). ' '.</code> <code>key(\$t);</code>

➔ symulacja kolejki i stosu za pomocą tablicy – funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
array_push, array_pop	Wstawia/Zdejmuje element na końcu tablicy (symulacja stosu).	<code>array_push(\$stos,1,2,3,4,5);</code> <code>print_r(\$stos);</code> <code>echo array_pop(\$stos);</code> <code>print_r(\$stos);</code>	array_unshift, array_shift	Wstawia/Zdejmuje element na początku tablicy (symulacja kolejki).	<code>array_unshift(\$t_k,1,2,3,4);</code> <code>print_r(\$t_k);</code> <code>echo array_shift(\$t_k);</code> <code>print_r(\$t_k);</code>

➔ funkcje operujące na tablicach wykorzystujące funkcje zwrotne – funkcje:

Funkcje	Opis	Przykłady
array_filter	Filtruje zawartość tablicy przy użyciu funkcji zwrotnej	<code>\$t=array(false, 0, null, -1, 'a'); print_r(array_filter(\$t));</code> <code>function npar(\$w) {return (\$w&1);} print_r(array_filter(\$t,'npar'));</code>
array_reduce, array_reduce_recursive	Iteracyjnie redukuje tablicę do pojedynczej wartości używając funkcji zwrotnej.	<code>\$t=array(1,2,3,4,5); \$tx=array();</code> <code>\$t1 = array_reduce(\$t, 'rsum'); \$t2 = array_reduce(\$t, 'rmul',10);</code> <code>\$t3 = array_reduce(\$tx, 'rsum', 'tx_pusta'); echo \$t1.' '.\$t2.' '.\$t3;</code>
array_map	Wykonuje funkcję zwrótną na wszystkich elementach tablicy.	<code>\$t=array(1,2,3,4,5);</code> <code>\$t=array_map('my_sqr',\$t); print_r(\$t);</code>
array_walk, array_walk_recursive	Wykonuje funkcję użytkownika na wszystkich elementach tablicy (ostatnia funkcja obsługuje tablice wielowymiarowe).	<code>function drukuj(&\$item, \$key) { echo \$key.'='.\$item.'
';}</code> <code>function my_sqr(&\$item, \$key, \$tekst) { \$item=\$item*\$item. \$tekst; }</code> <code>\$t=array('d'=>1, 'a'=>2, 'b'=>3, 'c'=>4);</code> <code>array_walk(\$t, 'drukuj'); array_walk(\$t, 'my_sqr', 'j<sup>2</sup>');</code> <code>echo '
'; array_walk(\$t, 'drukuj');</code>

➔ funkcje konwertujące – wyróżniamy:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
explode; implode, join	Zamiana tekstu z separatorami na tablicę i odwrotnie.	<code>\$s = 'a,b,c,1,2,3';</code> <code>\$t = explode(',', \$s);</code> <code>var_dump(\$t); echo "\n";</code> <code>echo implode(' ', \$t);</code>	str_getcsv	Konwertuje tekst w formacie CSV do tablicy.	<code>\$s="1;2;3;4;5;6;7";</code> <code>\$t=str_getcsv(\$s, ';');</code> <code>print_r(\$t);</code>

➔ pozostałe funkcje operujące na tablicach – funkcje:

Funkcje	Opis	Przykłady	Funkcje	Opis	Przykłady
array_diff, array_intersect	Zwraca różnice / przecięcie pomiędzy tablicami.	<code>\$t=array(0,1,2,3);</code> <code>\$t2=array(2,3,4,5);</code> <code>\$t=array_diff(\$t,\$t2);</code> <code>var_dump(\$t);</code> <code>\$t=array_intersect(\$t,\$t2);</code> <code>var_dump(\$t);</code>	array_change_key_case	Ustawia wielkość liter we wszystkich kluczach tablicy.	<code>\$t=array('Abc', 'XY', 'xy');</code> <code>\$t=array_change_key_case(\$t);</code> <code>print_r(\$t);</code> <code>\$t=array_change_key_case(\$t, CASE_LOWER);</code> <code>print_r(\$t);</code>
array_fill, array_pad	Wypełnia całą tablicę / Dopełnia tablicę do podanej długości podanymi wartościami.	<code>\$t=array_fill(5,3,'hej');</code> <code>print_r(\$t);</code> <code>\$t=array_pad(\$t, 5, 0);</code> <code>print_r(\$t);</code> <code>\$t=array_pad(\$t, -7, '*');</code> <code>print_r(\$t);</code>	array_chunk	Dzieli tablicę na równe kawałki (ostatni kawałek może mieć inny rozmiar).	<code>\$t=array(1,2,3,4,5,6,7,8);</code> <code>\$t=array_chunk(\$t,3);</code> <code>print_r(\$t);</code> <code>\$t=array_chunk(\$t,3,true);</code> <code>print_r(\$t);</code>
compact, extract	Konwersja pomiędzy listą zmiennych, a tablicą zmiennych i odwrotnie.	<code>\$miasto = 'Nysa';</code> <code>\$kraj = 'PL';</code> <code>\$imie = 'Ewa';</code> <code>\$lista = array('kraj1','kraj2');</code> <code>print_r(compact('imie',\$lista));</code>	array_count_values	Zlicza liczbę wystąpień każdej wartości w tablicy.	<code>\$t=array(4,1,1,1,3,2,2,1);</code> <code>\$t=array_count_values(\$t);</code> <code>print_r(\$t);</code>
each list	Zwraca bieżącą parę klucza i wartości z tablicy i przesuwając kursor tablicy. Przypisuje zmienne tak jakby były tablicą.	<code>\$t=array('a'=>'x', 'b'=>'y');</code> <code>\$t_each = each(\$t);</code> <code>var_dump(\$t, \$t_each);</code> <code>while (list(\$k,\$w)=each(\$t))</code> <code>echo "\$k=>\$w
";</code>	array_merge, array_merge_recursive	Łączy jedną lub więcej tablic. Ostatnia funkcja zachowuje wielokrotne zagnieżdżenia elementów.	<code>\$t1=array(1,2,'a'=>'b');</code> <code>\$t2=array('a'=>'c',1,2,3);</code> <code>\$t=array_merge(\$t1,\$t2);</code> <code>print_r(\$t);</code> <code>\$t3=array('a'=>\$t1,5);</code> <code>\$t4=array(10,'a'=>\$t2);</code> <code>\$t=array_merge_recursive(\$t3,\$t4); print_r(\$t);</code>
array_fill_keys	Podmienia klucze w tablicy według wartości z innej tablicy.	<code>\$tk = array('x',5,8,'Y');</code> <code>\$t = array_fill_keys(\$tk, 't');</code> <code>print_r(\$t);</code>	array_combine	Tworzy tablicę biorąc z jednej tablicy klucze, a z drugiej – wartości.	<code>\$t1 = array('a','b',1,2);</code> <code>\$t2 = array('x',8,9,'y');</code> <code>\$t=array_combine(\$t1,\$t2);</code> <code>print_r(\$t);</code>
			array_flip	Zamiana: klucze stają się wartościami, a wartości – kluczami.	<code>\$t = array('x','y','z');</code> <code>print_r(array_flip(\$t));</code>

Zadania

1. Zapoznaj się z opisanymi wyżej funkcjami. Wypróbuj i przeanalizuj działanie podanych przykładów.
2. Utwórz 2 tablicę: zawierającą owoce i warzywa. Połącz je. Posortuj połączoną tablicę najpierw rosnąco, a później malejąco. Dokumentuj kolejne kroki wyświetlając zawartość tablicy.
3. Wygeneruj tablicę z kolejnymi liczbami od 1 do 1000 i pomieszaj te wartości w tablicy.
4. Przygotuj test składający się z 5 pytań (do każdego pytania 3 odpowiedzi). Po napisaniu testu użytkownik ma zostać poinformowany o liczbie punktów wyniku testu (zaliczony, nie zaliczony). Pytania i odpowiedzi trzymaj w tablicy.