**Tablice w PHP cz.III**

W lekcji tej przedstawione zostanie wykorzystanie niektórych funkcji PHP dotyczących tablic. Pełną listę takich funkcji znajdziesz na: <http://www.w3schools.com/php/php_ref_array.asp>

Sortowanie jednowymiarowej tablicy indeksowanej numerycznie za pomocą funkcji sort()

**sort(array, sortingtype);**

* array – nazwa tablicy
* sortingtype – typ sortowania (opcjonalny), np.
* SORT\_REGULAR – domyślny - porównuje elementy normalnie (nie zmienia typów)
* SORT\_NUMERIC - porównuje elementy jako liczby
* SORT\_STRING - porównuje elementy jako ciągi tekstowe

**UWAGA!** Funkcja sortuje tablice jednowymiarowe – rosnąco. Ostrożnie z polskimi znakami w tekstach.

Aby sortowanie uwzględniało polskie znaki dodaj: ***setlocale(LC\_ALL, 'pl\_PL', 'pl', 'polish');***

i sortuj tak: ***sort($t1, SORT\_LOCALE\_STRING);***

*Ćwiczenie 1* Posortuj i wypisz poniższe tablice.

$t1=array('jabłka','cytryny','gruszki','porzeczki'); $t2=array('k1'=>'jabłka','k2'=>'cytryny','k3'=>'gruszki','k4'=>'porzeczki');

$t3=array(2,12,9,8);

Sortowanie jednowymiarowej tablicy asocjacyjnej za pomocą funkcji asort()

W przypadku tablic asocjacyjnych, zmiana kluczy jest niedopuszczalna, dlatego do sortowania tych tablic nie używa się sort().

*Ćwiczenie 2* Posortuj tablice z poprzedniego ćwiczenia funkcją asort().

Sortowanie jednowymiarowej tablicy asocjacyjnej za pomocą funkcji ksort()

*Ćwiczenie 3* Posortuj tablice z poprzedniego ćwiczenia funkcją ksort().

UWAGA! Tablice zostały posortowane według kluczy w kolejności 'k1', 'k2', 'k3', 'k4'. Elementy tablic indeksowanych numerycznie, nie zmieniły kolejności.

Sortowanie jednowymiarowych tablic malejąco

Wszystkie zaprezentowane powyżej funkcje sortują rosnąco. Każda z nich ma swój odpowiednik sortujący malejąco.

* sort() - rosnąco, rsort() - malejąco
* asort() - rosnąco, arsort() - malejąco
* ksort() - rosnąco, krsort() - malejąco

*Ćwiczenie 4* Przetestuj funkcje sortujące malejąco.

Przetasowanie tablicy za pomocą funkcji shuffle()

Podczas "mieszania" elementów, tworzone są nowe klucze, więc funkcja ta nie nadaje się to tablic asocjacyjnych. W wyniku zastosowania funkcji shuffle(), kolejność elementów tablic zostanie zmieniona w sposób losowy. Utworzone zostaną nowe indeksy.

*Ćwiczenie 5* Przetasuj elementy tablic i wypisz je:

$t1=array(3,1,67,34,8);

$t2=array('k1'=>3,'k2'=>1,'k3'=>67,'k4'=>34,'k5'=>8);

Odwrócenie kolejności elementów tablicy za pomocą funkcji array\_reverse()

Funkcja ma postać

**array\_reverse(array, keystate);**

* array – nazwa tablicy
* keystate – czy zachować klucze (opcjonalny):
* true
* false

Jak widzimy, opcjonalnie możemy również zachować klucze. Elementy tablicy asocjacyjnej zawsze zachowują klucze.

*Ćwiczenie 6* Odwróć i wypisz tablice bez zachowania kolejności indeksów i z zachowaniem.

$t1=array(3,1,67,34,8);

$t2=array('k1'=>3,'k2'=>1,'k3'=>67,'k4'=>34,'k5'=>8);

Tworzenie tablicy ze stringa za pomocą funkcji explode()

Tę funkcję możemy zastosować, jeżeli mamy ciąg znaków w którym poszczególne jego części oddzielone są określonym znakiem (np. znakiem przecinka, tabulacji, spacji, itd.) lub kilkoma znakami (np. '^^^'). Cały string ulega rozbiciu, powstaje tablica, której elementami są fragmenty znajdujące się pomiędzy znakami rozdzielającymi.

*Przykład*

$tekst='8^^12^^3^^45';

$tablica=explode('^^',$tekst);

Przekształcenie tablicy w ciąg znaków za pomocą funkcji implode()

Za pomocą tej funkcji wykonujemy operację odwrotną do pokazanej w poprzednim przykładzie.

*Przykład*

$tablica=array(8,12,3,45);

$tekst=implode('^^',$tablica);

echo $tekst;