

APLIKACJE INTERNETOWE PHP



TEMAT 1-05: Tablice.

Autor dokumentu: Wojciech Galiński

piątek, 16 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. Rodzaje tablic – wyróżniamy:

➔ **tablica sekwencyjna** – to tradycyjna tablica z indeksami w postaci liczb całkowitych (indeksy muszą należeć do ciągłego przedziału: [a, b] w języku Pascal, [0, N-1] w C/C++). Taką tablicę można bezpiecznie przeglądać za pomocą pętli „for” oraz „foreach”;

➔ **tablica asocjacyjna** – to tablica z indeksami w postaci tekstu (w szczególnym przypadku może pracować jako tablica tradycyjna).

Taką tablicę można bezpiecznie przeglądać tylko za pomocą pętli „foreach”.

W języku PHP wszystkie tablice są asocjacyjne, ale niektóre działają, jak tablice sekwencyjne.

2. Tablica w języku PHP – to obiekt umożliwiający przechowywać różnego typu pary (indeksy i wartości) – wszystko to pod jedną nazwą. Oto schematyczna budowa tablicy:

tablica	Element 1	Element 2	Element 3	Element 4
Indeksy:	0	'1.5'	'imie'	'nazwisko'
Wartości:	5.5	-123	'Jan'	'Nowak'

Do elementów tablicy (pojedynczych wartości w tablicy) odwołujemy się za pomocą operatora „[]”, np. \$tablica[1], \$tablica['1.5'], \$tablica['imie']. Jeśli odwołujemy się do indeksu, który nie istnieje, zostanie wygenerowany tekst:

„Notice: Undefined index: {nazwa_indeksu} in {nazwa_skryptu} on line {numer_linii}”

3. Tworzenie tablicy – oto przykłady:

```
// Pusta tablica
$nazwa_zmiennej = array(); // pusta tablica
// Tablica tradycyjna w stylu języka C++ (indeksy od 0 w zwyż)
$nazwa_zmiennej = array(wartosc_1, ..., wartosc_N);
// Tablica asocjacyjna (indeksy dowolne)
$nazwa_zmiennej = array(indeks_1=>wartosc_1, ..., indeks_N=>wartosc_N);
```

Należy pamiętać, aby indeksy oraz wartości w postaci tekstu były umieszczone pomiędzy apostrofami albo pomiędzy nawiasami.

PRZYKŁADY:

```
$pusta_tablica = array(); $cyfry_parzyste = array(0, 2, 4, 6, 8);
$miasta = array('Nysa', 'Opole'); $kolory = array("red", "blue");
$dane_osoby = array('imie'=>'Jan', 'pol'=>true, 'wiek'=>20, 'wzr'=>.9);
$ilosc_osob_wg_wieku = array(13=>15, 16=>290, 17=>165, 18=>78, 20=>3);
```

4. Tablice zagnieżdżone – to tablice, których elementami są tablice. W PHP tablice mogą być wielokrotnie zagnieżdżone

PRZYKŁADY:

```
$uczniowie = array( array('Ala', 12), array('Tomek', 15) );
$uczniowie_2 = array($uczniowie, $uczniowie, $uczniowie, $uczniowie);
$stab3D = array( array( array(0,0), array(0,0) ),
                 array( array(0,0), array(0,0) ) );
```

5. **Odczyt i zapis zawartości tablicy** – wymaga podania nazwy tablicy oraz indeksu elementu:

```
// Odczyt zmiennej z tablicy
echo $nazwa_tablicy[indeks]; $nazwa_zmiennej = $nazwa_tablicy[indeks];
// Zapis zmiennej do wybranego indeksu w tablicy
$nazwa_tablicy[indeks] = wartosc_zmiennej_lub_wyrazenie;
// Zapis zmiennej na końcu tablicy
$nazwa_tablicy[] = wartosc_zmiennej_lub_wyrazenie;
// Rozmiar tablicy
echo count($nazwa_tablicy);
```

PRZYKŁADY:

```
$liczby = array(2,3,5,7,11); $miasta = array('Nysa', 'Ełk', 'Łódź');
echo $liczby[1]. ' '. $miasta[4]. '<br />'. 'Rozmiar: '. count($miasta);
$auto = array( 'bus'=>array('Opel', 2005), 'tir'=>array('VW', 2012) );
$tekst = $auto['tir'][0]. ' ('. $auto['tir'][1]. ')'; echo $tekst;
```

6. **Podglądanie tablic (debugowanie)** – służą do tego funkcje: „var_dump” oraz „print_r”.

PRZYKŁADY:

```
for ($i=0; $i<3; ++$i) $stab[$i][$i][$i] = $i*$i*$i;
echo '<p>PRINT_R:</p><pre>'; print_r($stab); echo '</pre>';
echo '<p>VAR_DUMP:</p><pre>'; var_dump($stab); echo '</pre>';
```

7. **Usuwanie wartości z elementu tablicy** – wykonujemy to za pomocą następującej instrukcji:

```
$nazwa_tablicy[indeks] = null;
```

PRZYKŁAD: \$stab[0][0] = null; echo '''. \$stab[0][0]. ''';

8. **Usuwanie elementu tablicy** – wykonujemy to za pomocą następującej instrukcji:

```
unset($nazwa_tablicy[indeks]);
```

PRZYKŁAD: unset(\$stab[0][0]); echo \$stab[0][0];

9. **Usuwanie całej tablicy** – wykonujemy to za pomocą następującej instrukcji:

```
unset($nazwa_tablicy);
```

Powyższa instrukcja usuwa całą tablicę z pamięci. Od tej chwili odwoływanie się do niej generować będzie błąd.

PRZYKŁAD: unset(\$stab); echo \$stab[1][1][1];

10. **Pętla „foreach”** – służy do przeglądania wszystkich indeksów tablicy. Oto wzorce tej instrukcji:

```
foreach ($stab as $wartosc)      foreach ($stab as $indeks=>$wartosc)
{                                {
    // instrukcje                // instrukcje
}
```

W każdej iteracji zmienna „\$wartosc” (wartość z bieżącej komórki) przyjmuje kolejną wartość z tablicy „\$stab”, a zmienna „\$indeks” przechowuje kolejną nazwę indeksu tablicy „\$stab”.

Pętla „foreach” powstała specjalnie w celu przeglądania całych tablic asocjacyjnych, ponieważ jej indeksami są teksty, a nie kolejne liczby.

PRZYKŁADY:

```
$dane_osoby = array('imie'=>'Jan', 'pol'=>true, 'wiek'=>20, 'wzr'=>.9);
foreach ($dane_osoby as $k=>$w) echo $k. '=>'. $w. ' ';
$auto = array( 'bus'=>array('Opel', 2005), 'tir'=>array('VW', 2012) );
foreach ($auto as $t) foreach ($t as $k=>$w) echo $k. '=>'. $w. ' ';
```

11. **Eksport i import tablicy formatu „JSON”** – eksport tablicy do tekstu w formacie „JSON” umożliwia nam funkcja: „json_encode”. Natomiast tekst w formacie „JSON” możemy zaimportować do tablicy PHP za pomocą funkcji: „json_decode”.

PRZYKŁAD:

```
$stab = array(imie=>'Jan', 'wiek'=>18, 'wzrost'=>1.78, 'polak'=>true);
$json = json_encode($stab); echo '<p>'. $json. '</p>';
$stab = json_decode($json, true); echo '<pre>'; print_r($stab); echo '</pre>';
```

Zadania

1. Utwórz tablicę obecnych osób z Twojej grupy wraz z ich wzrostem, np. 'Jan'=>1.82.
2. Utwórz tablicę przechowującą elementy ciągu $a_n=2n+1$, $n=1, \dots, 1000$.
3. Zapisz do pliku dane 3 osób wzorując się przykładem z punktu 11. Następnie odczytaj te dane i wyświetl.