**Tablice w PHP cz.I**

Tablice wymyślono po to, aby w jednej zmiennej **można było przechowywać wiele różnych danych**. Co ciekawe, o ile np. w C++ czy Pascalu muszą to być dane jednego typu, o tyle w PHP nie ma takich ograniczeń. Bardzo ważny jest szybki dostęp do danych przechowywanych w tablicy. Pliki są czytane sekwencyjnie - jeden element po drugim, aby "dostać się" do potrzebnego elementu musimy wczytać wszystkie elementy znajdujące się przed nim. Do **elementu tablicy** dostajemy się natychmiast, poprzez podanie **indeksu** lub **klucza** danego elementu. Na przykład $t[12] jest elementem tablicy $t o indeksie równym 12, natomiast $t['nazwisko'] jest elementem tablicy $t o kluczu 'nazwisko'. W zależności od tego tablice dzielimy na [*indeksowane numerycznie*](http://www.teacher.webd.pl/kurs_php/k_2_4_0.php) oraz [*tablice asocjacyjne*](http://www.teacher.webd.pl/kurs_php/k_2_5_0.php)*.*

Jeżeli elementy tablicy są także tablicami, to mamy [**tablicę wielowymiarową**](http://www.teacher.webd.pl/kurs_php/k_2_3_0.php). Jeżeli chcemy na przykład zapisać w tablicy, dane znajdujące się w tabeli, to musimy zastosować tablicę dwuwymiarową, w której każdy wiersz tabeli (rekord) będzie się składał z kolumn (pól) tabeli. Elementy takiej tablicy $tabela[$wiersz][$kolumna] będą określane przez dwa indeksy - numer wiersza i numer kolumny. Wyróżniamy w ten sposób podział na tablice jednowymiarowe, dwuwymiarowe, itd.

**UWAGA! Indeksy w tablicach rozpoczynają się od zera!**

***Inicjowanie tablic*** – czyli jak wypełniać tablicę

1. Inicjowanie tablicy poprzez zastosowanie funkcji array(), bez określania indeksów

**$liczby = array(1, 2, 5, 1.25);**

*Przykład*

<?php

$liczby = array(1, 2, 5, 1.25);

//wypisanie każdego elementu tablicy oddzielną instrukcją echo

echo '$liczby[0] przechowuje wartość '.$liczby[0].'<br />';

echo '$liczby[1] przechowuje wartość '.$liczby[1].'<br />';

echo '$liczby[2] przechowuje wartość '.$liczby[2].'<br />';

echo '$liczby[3] przechowuje wartość '.$liczby[3].'<br />';

echo '<br /><br />';

//a teraz przy pomocy pętli for

for($i=0; $i<4; $i++)

echo '$liczby['.$i.'] przechowuje wartość '.$liczby[$i].'<br />';

//a teraz przy pomocy specjalnej funkcji print\_r()

echo'<pre>'; //wykomentuj

print\_r($liczby);

echo'</pre>';//wykomentuj

?>

1. Inicjowanie tablicy poprzez zastosowanie funkcji array(), z określeniem indeksów

**$liczby = array(1, 20 => 2, 10 => 5, 1.25);**

*Przykład*

<?php

$liczby=array(1, 20 => 2, 10 => 5, 1.25);

echo '$liczby[0] przechowuje wartość '.$liczby[0].'<br />';

echo '$liczby[20] przechowuje wartość '.$liczby[20].'<br />';

echo '$liczby[10] przechowuje wartość '.$liczby[10].'<br />';

echo '$liczby[21] przechowuje wartość '.$liczby[21].'<br />';

echo'<pre>';

print\_r($liczby);

echo'</pre>';

echo '$liczby[5] przechowuje wartość '.$liczby[5];

?>

**W praktyce, najczęściej indeksy numeryczne są kolejnymi liczbami począwszy od zera**.

1. Inicjowanie tablicy asocjacyjnej poprzez zastosowanie funkcji array()

**$liczby=array('pierwsza'=>1,'dwudziesta'=>2,'dziesiata'=>5,'dwudziesta\_pierwsza'=>1.25);**

*Przykład*

<?php $liczby=array('pierwsza'=>1,'dwudziesta'=>2,'dziesiata'=>5,'dwudziesta\_pierwsza'=>1.25);

echo'<pre>';

print\_r($liczby);

echo'</pre>';

?>

1. Inicjowanie tablic poprzez zastosowanie nawiasów kwadratowych bez indeksów

**$liczby[]=1;**

**$liczby[]=2;**

**$liczby[]=5;**

**$liczby[]=1.25;**

*Przykład*

<?php

$liczby[]=1; $liczby[]=2; $liczby[]=5; $liczby[]=1.25;

echo '$liczby[0] przechowuje wartość '.$liczby[0].'<br />';

echo '$liczby[1] przechowuje wartość '.$liczby[1].'<br />';

echo '$liczby[2] przechowuje wartość '.$liczby[2].'<br />';

echo '$liczby[3] przechowuje wartość '.$liczby[3].'<br />';

echo'<pre>';

print\_r($liczby);

echo'</pre>';

?>

1. Inicjowanie tablicy poprzez zastosowanie nawiasów kwadratowych z określeniem indeksów

**$liczby[]=1;**

**$liczby[20]=2;**

**$liczby[10]=5;**

**$liczby[]=1.25;**

*Przykład*

<?php

$liczby[]=1; $liczby[20]=2; $liczby[10]=5; $liczby[]=1.25;

echo '$liczby[0] przechowuje wartość '.$liczby[0].'<br />';

echo '$liczby[20] przechowuje wartość '.$liczby[20].'<br />';

echo '$liczby[10] przechowuje wartość '.$liczby[10].'<br />';

echo '$liczby[21] przechowuje wartość '.$liczby[21].'<br />';

echo'<pre>';

print\_r($liczby);

echo'</pre>';

?>

1. Inicjowanie tablicy asocjacyjnej poprzez zastosowanie nawiasów kwadratowych

**$liczby['pierwsza']=1;**

**$liczby['dwudziesta']=2;**

**$liczby['dziesiata']=5;**

**$liczby['dwudziesta\_pierwsza']=1.25;**

*Przykład*

<?php

$liczby['pierwsza']=1; $liczby['dwudziesta']=2; $liczby['dziesiata']=5;

$liczby['dwudziesta\_pierwsza']=1.25;

echo'<pre>';

print\_r($liczby);

echo'</pre>';

?>

**Film 22, 23, 24, 25, 26**

***Tablice $\_GET i $\_POST***

Już na początku nauki PHP, [nauczyliśmy się odbierać dane wysyłane formularzem](http://www.teacher.webd.pl/kurs_php/k_1_4_0.php). Wykonaliśmy do tej pory wiele ćwiczeń, nie zdając sobie nawet sprawy z tego, że korzystamy z [tablic asocjacyjnych](http://www.teacher.webd.pl/kurs_php/k_2_0_0.php). Kluczami tablic $\_GET i $\_POST są nazwy (atrybut name) [pól formularzy](http://www.teacher.webd.pl/kurs_html5/k_2_6_2.php).

*Ćwiczenie1* Zastosuj funkcję print\_r() do podglądu zawartości tablic $\_GET i $\_POST stosowanych we wcześniejszych skryptach.

***Dostęp do elementów o znanych indeksach lub kluczach***

Jeżeli chcemy uzyskać dostęp do elementu tablicy o znanym indeksie lub kluczu, to mamy prostą sytuację, np.

$element=$tablica[indeks]; lub $element=$tablica['jakis\_klucz'];

Jeżeli nie mamy pewności, czy dany element istnieje możemy dodatkowo zastosować funkcje isset() lub empty().

***Dostęp do elementów poprzez zastosowanie pętli for oraz funkcji count()***

Tworząc tablicę indeksowaną numerycznie, najczęściej dbamy o to, aby indeksy stanowiły kolejne liczby całkowite począwszy od zera. W takiej sytuacji możemy przeglądać dane stosując pętlę for oraz funkcję count(), określającą liczbę elementów danej tablicy.

*Ćwiczenie2* Napisz skrypt, który wypełni n-elementową tablicę, liczbami losowymi, branymi z zakresu od 30 do 300, a następnie wyświetli elementy tej tablicy. Liczbę n wylosuj z liczb od 2 do 10. Zastosuj jeden plik ze skryptem PHP.

<?php

$rozmiar\_tablicy=rand(2,10);

for ($i=0; $i<$rozmiar\_tablicy; $i++)

$tablica[$i]=rand(30,300);

//a jeżeli byśmy nie znali rozmiaru tablicy

for ($i=0; $i<count($tablica); $i++)

echo $tablica[$i].', ';

?>

***Dostęp do elementów poprzez zastosowanie pętli foreach***

Pętla foreach jest idealna do przeglądania elementów tablicy. Nie musimy znać liczby elementów tablicy, ani jej indeksów czy kluczy. Po prostu w każdym obiegu pętli dostajemy kolejny element tablicy.

**Film 27**

*Ćwiczenie3* Zmodyfikuj skrypt z poprzedniego ćwiczenia tak, aby zastosować petlę foreach.

<?php

$rozmiar\_tablicy=rand(2,10);

for ($i=0; $i<$rozmiar\_tablicy; $i++)

$tablica[$i]=rand(30,300);

//a jeżeli byśmy nie znali rozmiaru tablicy

foreach ($tablica as $wartosc\_elementu)

echo $wartosc\_elementu.', ';

?>

*Ćwiczenie4* Skrypt z poprzedniego ćwiczenia zmodyfikujemy tak, aby wyświetlał oprócz wartości kolejnych elementów, również ich indeksy.

<?php

$rozmiar\_tablicy=rand(2,10);

for ($i=0; $i<$rozmiar\_tablicy; $i++)

$tablica[$i]=rand(30,300);

//a jeżeli byśmy nie znali rozmiaru tablicy

foreach ($tablica as $indeks=>$wartosc\_elementu)

echo '$tablica['.$indeks.']='.$wartosc\_elementu.', ';

?>

*Ćwiczenie5* Sposób prezentowany w ostatnim ćwiczeniu jest szczególnie przydatny do przeglądania tablic asocjacyjnych.

<?php

$uczen=array('imie'=>'Franek', 'nazwisko'=>'Dolas',

'wiek'=>25, 'wzrost'=>180);

foreach ($uczen as $klucz=>$wartosc\_elementu)

echo $klucz.' - '.$wartosc\_elementu.'<br />';

?>

**Zadanie 1** W pliku ***funkcje.php*** zapisz funkcję:

1. ***utworz\_tab10()***, która zwraca tablicę 10-ciu liczb całkowitych losowanych z przedziału <1,100>;
2. ***utworz\_tabn($n)***, która zwraca tablicę n liczb całkowitych losowanych z przedziału <1,100>;
3. ***utworz\_tabassocn($n)***, która zwraca tablicę asocjacyjną n liczb całkowitych losowanych z przedziału <1,100>, a klucze w tej tablicy są postaci: klucz0, klucz1, klucz2,…
4. ***utworz\_tab2d($n, $k)***, która zwraca tablicę o n wierszach i k kolumnach, której elementami są kolejne liczby całkowite od 1 do n\*k;

Napisz skrypt, w którym dołączony będzie plik funkcje.php oraz pokazane będą elementy z tablicy wraz ze swoimi indeksami.

**Zadanie 2** *Znajdowanie wartości maksymalnej w tablicy liczb*. Napisz skrypt, który wygeneruje tablicę 20-tu liczb losowych (funkcja z zad1b)) i pokaże ją na stronie. Znajdź i wypisz maksymalny element tej tablicy wraz z podaniem indeksu.

**Zadanie 3** *Zmiana wartości elementu tablicy.* Napisz skrypt, który wygeneruje tablicę 10-tu liczb losowych (funkcja z zad1b)) i pokaże ją na stronie. Do każdego parzystego elementu tablicy dodaj 1, od każdego nieparzystego elementu odejmij 1. Pokaż tablicę po zmianach i wypisz ilość liczb nieparzystych (Uwaga! Zera nie liczymy!).

**Zadanie 4** *Zmiana kolejności elementów tablicy.* Napisz skrypt, który wygeneruje tablicę 20-tu liczb losowych (funkcja z zad1b)) i pokaże ją na stronie. Następnie skrypt zamieni kolejność elementów w tablicy wg zasady: ostatni staje się pierwszym, przedostatni – drugim, itd. Na koniec skrypt ponownie pokaże tablicę.

**Zadanie 5** *Usuwanie elementów z tablicy.* Napisz skrypt, który wygeneruje tablicę 10-tu liczb losowych (funkcja z zad1b)) i pokaże ją na stronie. Następnie skrypt usunie z tej tablicy liczby większe od 60. Wykorzystaj do tego drugą tablicę. Na koniec skrypt wypisze ile zostało liczb po usunięciu.