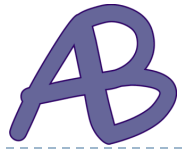


Tablice



Tablice

Tablica to zmienna, która może przechowywać wiele elementów.
Elementy w tablicy są indeksowane, co oznacza, że można je odwoływać się do nich za pomocą numeru indeksu lub klucza.

W PHP istnieje kilka rodzajów tablic:

Indeksowane numerycznie: ich elementy są indeksowane liczbami całkowitymi i domyślnie zaczynają się od zera.

Asocjacyjne: ich elementy są indeksowane za pomocą nazw kluczy (ciągów znaków), co pozwala na bardziej elastyczne i opisowe odwoływanie się do elementów.

Wielowymiarowe: to tablice, które same w sobie zawierają inne tablice. Pozwalają na tworzenie bardziej złożonych struktur danych.

funkcje `print_r()` i `var_dump()`

Funkcje `print_r()` i `var_dump()` są przydatne do wyświetlenia struktury tablicy w celu debugowania i analizy zawartości:

```
<?php
    // Wyświetlenie zawartości tablicy za pomocą print_r
    print_r($tablica);

    // Wyświetlenie bardziej szczegółowych informacji za pomocą var_dump
    var_dump($tablica);
?>
```

Tablice Indeksowane numerycznie:

```
<?php
    $indeksowanaTablica = array("jabłko", "banan", "śliwka");
?>
```

lub

```
<?php
    $indeksowanaTablica = ["jabłko", "banan", "śliwka"];
?>
```

Tablice asocjacyjne (klucz – wartość):

```
<?php
    $asocjacyjnaTablica = array(„x” => 3, „y” => 5, „z” => 2);
    // lub
    $asocjacyjnaTablica = [„x” => 3, „y” => 5, „z” => 2];
?>
```

lub

```
<?php
    $asocjacyjnaTablica = ["owoc1" => "jabłko" , "owoc1" => "banan", "owoc1" => "śliwka"];
?>
```

Tablice mieszane – zawierające dane różnych typów

```
<?php
    $mieszanaTablica = [1, "jabłko", 3.14, true];
?>
```

Tablica z indeksami niezaczynającymi się od zera

```
<?php
    $niestandardowaTablica = array(10 => "pierwszy", 20 => "drugi", 30 => "trzeci");
?>
```

Tablice wielowymiarowe:

Wielowymiarowe tablice to tablice, które zawierają inne tablice. Mogą być indeksowane numerycznie lub asocjacyjnie.

```
<?php
    $wielowymiarowaTablica = array(
        "owoce" => array("jabłko", "banan", "śliwka"),
        "warzywa" => array("marchew", "ziemniak", "pomidor")
    );
?>
```


Tablice wielowymiarowe indeksowane numerycznie:

```
<?php
    $wielowymiarowaTablica = array(
        array("jabłko", "banan", "śliwka"),
        array("marchew", "ziemniak", "pomidor"),
        array("czerwony", "zielony", "niebieski")
    );
?>
```

lub

```
<?php
    $wielowymiarowaTablica = [
        ["jabłko", "banan", "śliwka"],
        ["marchew", "ziemniak", "pomidor"],
        ["czerwony", "zielony", "niebieski"]
    ];
?>
```

```
echo $wielowymiarowaTablica[0][0];
// Wyświetli "jabłko"
echo $wielowymiarowaTablica[1][2];
// Wyświetli "pomidor"
echo $wielowymiarowaTablica[2][1];
// Wyświetli "zielony"
```

Jawne tworzenie
tablic

```
<?php
// Jawne tworzenie prostej tablicy
$tab01[0] = "tekst 01";
$tab01[1] = "tekst 02";
$tab01[] = "tekst 03";
// przypisanie do pierwszego wolnego indeksu (w typ przypadku 2)
$tab01[] = "tekst 04";
echo "$tab01[0], $tab01[1], $tab01[2], $tab01[3]<br>";

// Tworzenie tablicy asocjacyjnej
$color["niebieski"] = "#0000FF";
$color["zielony"] = "#00FF00";
$color["czerwony"] = "#FF0000";
echo "Wartość szesnastkowa koloru czerwonego wynosi
{$color['czerwony']}<br>";

?>
```

Deklarowanie tablic

Jawne tworzenie
tablic c.d.



```
<?php
// Tworzenie tej samej co poprzedniej tablicy asocjacyjnej
// inny zapis
$color = array( "niebieski" => "#0000FF",
               "zielony"   => "#00FF00",
               "czerwony"  => "#FF0000" );
echo "Wartość szesnastkowa koloru zielonego wynosi
{$color['zielony']}<br>";

// Ręczne tworzenie tablicy wielowymiarowej
$tab02[0][0] = "Zero Zero";
$tab02[0][1] = "Zero Jeden";
echo "Wartością \${tab02[0][1]} jest {\${tab02[0][1]}<br>";

// Ręczne tworzenie asocjacyjnej tablicy wielowymiarowej
$tab03["Idaho"][0] = "Ada";
$tab03["Idaho"][1] = "Adams";
$tab03["Idaho"][2] = "Bannock";
$tab03["Arizona"][0] = "Apache";
$tab03["Arizona"][1] = "Cochise";
$tab03["Arizona"][2] = "Coconino";
echo "\$counties['Idaho'][0] = {\${tab03['Idaho']}[0]}<br>" ;
?>
```

Wypisywanie tablicy indeksowanej numerycznie



Tablice indeksowane numerycznie to tablice, w których elementy są indeksowane liczbami całkowitymi. Możesz użyć pętli for lub foreach do wyświetlenia zawartości takiej tablicy:

```
<?php
    for ($i = 0; $i < 10; $i++) {
        echo $tablica[$i];
    }
?>
```

```
<?php
    for ($i = 0; $i < count($tablica); $i++) {
        echo $tablica[$i];
    }
?>
```

sizeof() to alias dla funkcji count(), więc można go używać w ten sam sposób.

```
<?php
    foreach ($tablica as $element) {
        echo $element . "<br>";
    }
?>
```

Wypisywanie tablicy asocjacyjnej



Tablice asocjacyjne zawierają elementy indeksowane za pomocą kluczy (ciągów znaków). Możesz użyć pętli foreach, aby wyświetlić zawartość takiej tablicy:

```
<?php
    $asocjacyjnaTablica = array("jabłko" => 3, "banan" => 5, "śliwka" => 2);

    // Użycie pętli foreach
    foreach ($asocjacyjnaTablica as $klucz => $wartosc) {
        echo $klucz . ": " . $wartosc . "<br>";
    }
?>
```

Wypisywanie tablicy wielowymiarowej



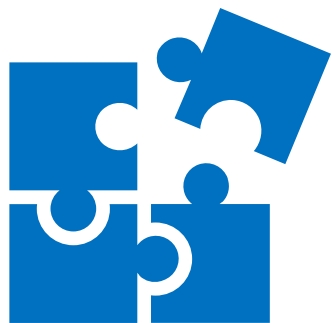
Tablice asocjacyjne zawierają elementy indeksowane za pomocą kluczy (ciągów znaków). Możesz użyć pętli foreach, aby wyświetlić zawartość takiej tablicy:

```
<?php
$wielowymiarowaTablica = [
    ["jabłko", "banan", "śliwka"],
    ["marchew", "ziemniak", "pomidor"],
    ["czerwony", "zielony", "niebieski"]
];

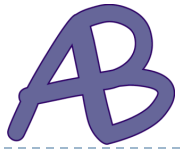
// Użycie zagnieżdżonych pętli foreach
foreach ($wielowymiarowaTablica as $wiersz) {
    foreach ($wiersz as $element) {
        echo $element . " ";
    }
    echo "<br>";
}
?>
```



Przykład do wykonania



Program zapełnia tablicę losowymi liczbami i sprawdza, ile razy znajduje się w niej szukana liczba.



Przykład do wykonania

```
<?php
$szukana = 5;

for ($i = 0; $i < 20; $i++) {
    $liczby[$i] = rand(1, 20);
}

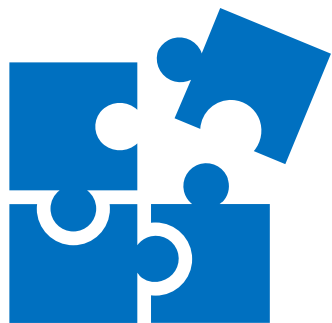
$ileZnaleziono = 0;

for ($i = 0; $i < 20; $i++) {
    print "$liczby[$i] ";
    if ($liczby[$i] == $szukana) {
        $ileZnaleziono++;
    }
}

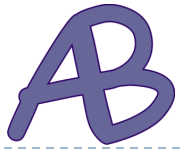
if ($ileZnaleziono == 0) {
    print("<p>Liczby $szukana w tablicy nie odnaleziono.</p>");
} else {
    print("<p>Liczbę $szukana w tablicy odnaleziono " .
        "$ileZnaleziono razy.</p>");
}
?>
```




Przykład do wykonania



Należy napisać skrypt który zamieni numer dnia tygodnia na jego nazwę



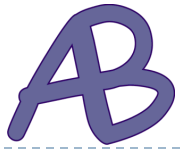
Przykład do wykonania

```
<?php
// Tablica z polskimi nazwami dni tygodnia
$dzienTygodnia = array(
    1 => "poniedziałek",
    2 => "wtorek",
    3 => "środa",
    4 => "czwartek",
    5 => "piątek",
    6 => "sobota",
    7 => "niedziela"
);

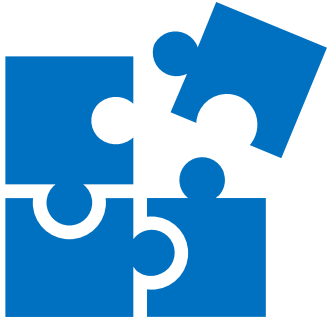
// Numer dnia tygodnia
$numerDnia = 3; // Możesz zmieniać ten numer, aby uzyskać różne nazwy dni

// Sprawdzenie, czy numer dnia mieści się w zakresie 1-7
if ($numerDnia >= 1 && $numerDnia <= 7) {
    // Pobranie nazwy dnia z tablicy
    $nazwaDnia = $dzienTygodnia[$numerDnia];
    echo "Numer $numerDnia odpowiada dniu tygodnia: $nazwaDnia";
} else {
    echo "Nieprawidłowy numer dnia tygodnia. Numer dnia musi być w zakresie od 1 do 7.";
}
?>
```





Przykład do wykonania



Należy napisać skrypt który zada użytkownikowi pytanie o ulubiony sport (w formie wyboru) i zapamięta odpowiedź;
W skrypcie wykorzystaj tablicę asocjacyjną zawierającą listę sportów, które można wybrać.



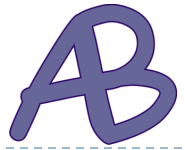
Przykład do wykonania

```
<?php
$opcje = array ('pilka' => 'piłka nożna', 'kosz' => 'koszykówka',
               'siat' => 'siatkówka', 'nar' => 'narciarstwo',
               'hokej' => 'hokej', 'boks' => 'boks',
               'inny' => 'inny sport');
```

Tablica zawiera pełną nazwę sportu oraz krótką nazwę którą łatwiej używać w programie

```
if (isset($_POST['opcja'])) { // wartości w formularzu są wpisane
    foreach ($opcje as $klucz => $wartosc) {
        if ($klucz == $_POST['opcja']) { $cwybrano = $wartosc; }
    }
    print ("Użytkownik wybrał opcję: <b>$cwybrano</b>.<br />");
    print '<br /><a href="p02.php">Powrót do formularza</a>';
} else { // nie ma wpisanych danych, wyświetlamy formularz
    print '<form action="p02.php" method="post"><div>';
    print 'Wybierz, jak sport lubisz:<br /><br />';
    foreach ($opcje as $klucz => $wartosc) {
        print ("<input type=\"radio\" name=\"opcja\" value=\""$klucz\"" />");
        print ($wartosc."<br />");
    }
    print '<p><input type="submit" value="Wyślij" /></p>';
    print '</div></form>';
}
?>
```

Przyciski typu radiobutton generowane są automatycznie na podstawie zawartości tablicy



Tablice ⇔ łańcuchy znaków

Zamiana tablic na stringi i odwrotnie

Funkcja **explode()** rozdziela ciąg znaków i tworzy z powstałych elementów tablicę.

Pobiera dwa argumenty:

- ciąg znaków lub znak, który stanowi element rozdzielający,
- zmienną zawierającą ciąg, który należy rozdzielić.

Przykład: Podział tekstu na pojedyncze zdania - separatorem będzie kropka

```
<?php
$tekst = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
convallis, quam nec malesuada rhoncus, odio metus vestibulum quam, in laoreet ex odio
scelerisque mi. Sed sit amet egestas augue. Proin fermentum sit amet nibh eget
facilisis. Suspendisse ultrices et lacus at placerat";

$tablica = explode('.', $tekst);
foreach($tablica as $wpis){
    echo "$wpis <br>";
}
?>
```

Zamiana tablic na stringi i odwrotnie

Funkcja **implode()** pobiera dwa argumenty:

- ciąg znaków, którymi mają być połączone elementy tablicy,
- nazwa zmiennej tablicy z danymi.

Przykład: Zapis do jednego łańcucha tekstu wszystkich elementów tablicy rozdzielonych średnikami

```
<?php
$tablica = array("Tekst 01", "Tekst 02", "Tekst 03", "Tekst 04");

$tekst = implode(";", $tablica);

echo $tekst;
?>
```

Istnieje wiele funkcji sortujących tablicę oto kilka z nich:

- **asort()** – sortuje rosnąco tablice asocjacyjne według wartości kluczy, zachowując przypisanie kluczy do wartości,
- **arsort()** – sortuje malejąco tablice asocjacyjne według wartości kluczy, zachowując przypisanie kluczy do wartości,
- **ksort()** – sortuje rosnąco tablice asocjacyjne według kluczy, zachowując przypisanie kluczy do wartości,
- **krsort()** – sortuje malejąco tablice asocjacyjne według kluczy, zachowując przypisanie kluczy do wartości,
- **sort()** – sortuje rosnąco zwykłe tablice,
- **rsort()** – sortuje malejąco zwykłe tablice,
- **uasort()** – funkcja sortująca tablice asocjacyjne za pomocą zdefiniowanej przez użytkownika funkcji porównującej elementy (nazwa funkcji jest podawana za pomocą drugiego parametru),
- **usort()** – funkcja sortująca zwykłe tablice za pomocą funkcji zdefiniowanej przez użytkownika,
- **uksort()** – funkcja sortująca tablice asocjacyjne według klucza za pomocą funkcji zdefiniowanej przez użytkownika.

Sortowanie tablic



asort() sortuje tablice asocjacyjne według wartości zachowując przypisanie kluczy do wartości

```
<?php
$owoce = array("d" => "mango", "a" => "papaja", "b" => "banan", "c" => "aronia");
asort($owoce);
foreach ($owoce as $klucz => $wartosc) {
    echo "$klucz = $wartosc<br>";
}
?>
```

c = aronia
b = banan
d = mango
a = papaja

ksort() sortuje tablice asocjacyjne kluczy

a = papaja
b = banan
c = aronia
d = mango

```
<?php
$owoce = array("d" => "mango", "a" => "papaja", "b" => "banan", "c" => "aronia");
ksort($owoce);
foreach ($owoce as $klucz => $wartosc) {
    echo "$klucz = $wartosc<br>";
}
?>
```

usort() - Sortowanie niestandardowe według funkcji porównującej:

```
<?php
function mojaFunkcjaPorownujaca($a, $b)
{
    // Niestandardowe kryterium sortowania
    if ($a == $b) {
        return 0;
    }
    ksort() sortuje tablice asocjacyjne kluczy
    return ($a < $b) ? -1 : 1;
}

$tablica = [3, 1, 2, 4, 5];
usort($tablica, "mojaFunkcjaPorownujaca");
?>
```



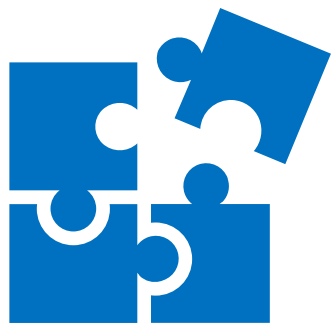
- **array_change_key_case (array wejście [, int wielkość])** – zmienia klucze w tablicy wejście, tak aby były one pisane tylko dużymi lub tylko małymi literami. Zmiana zależy od ostatniego opcjonalnego parametru case. Można do niego przekazać jedną z dwóch stałych:
 - CASE_UPPER lub CASE_LOWER. Domyślną wartością jest CASE_LOWER. Indeksy liczbowe będą pozostawione,
- **array_flip (trans)** – zwraca tablicę w odwróconym porządku, tzn. klucze z tabeli trans stają się wartościami, a wartości trans stają się kluczami. Zauważmy, że wszystkie wartości tablicy trans muszą mieć poprawne klucze, tzn. muszą być albo typu integer, albo string. Jeśli wartość nie ma prawidłowego typu, wyświetlone zostanie ostrzeżenie, a para klucz i wartość nie będzie odwrócona,
- **array_fill (indeks_początkowy, num, wartość)** – wypełnia tablicę wartością wartość, począwszy od indeksu indeks_początkowy przez num kolejnych elementów tablicy,
- **array_pop (tablica)** – zdejmuje i zwraca ostatnią wartość tablicy tablica, skracając tę tablicę o jeden element. Jeśli tablica jest pusta (lub nie jest tablicą), zwracana jest wartość NULL,
- **array_push (tablica, wartosc [, wartosc ...])** – traktuje zmienną tablica jako stos i wstawia przekazane parametry na koniec podanej tablicy. Długość parametru tablica zwiększa się o liczbę przekazanych wartości. Funkcja zwraca nową liczbę elementów tablicy,



- **array_shift (tablica)** – usuwa pierwszą wartość parametru tablica i zwraca go, skracając tę tablicę o jeden element i przesuwając wszystkie pozostałe elementy w dół. Jeśli tablica jest pusta (lub nie jest tablicą), zwracana jest wartość NULL,
- **array_unshift (tablica, wartość, [wartosc...])** – wstawia jeden lub więcej przekazanych jako parametry elementów na początek tablicy tablica. Zauważmy, że lista elementów jest wstawiana jako całość, więc elementy zostają w takim samym porządku. Funkcja zwraca nową liczbę elementów w tablicy tablica,
- **array_search (ciąg, tablica[, ścisły])** – przeszukuje tablicę w poszukiwaniu parametru "ciąg" i zwraca odpowiedni klucz, jeśli został on znaleziony, lub FALSE w przeciwnym wypadku. Jeżeli trzeci parametr ścisły jest ustawiony na TRUE, to array_search() porówna także typy parametru „ciąg” z tymi z parametru „tablica”.



Przykład do wykonania



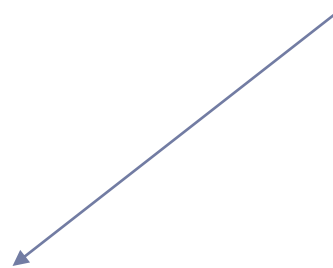
Należy napisać skrypt który otrzyma tablicę i wyświetli ją w tabeli

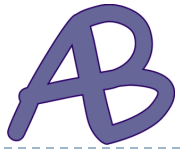


Przykład do wykonania

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" type="text/css" href="styl.css">
    <title></title>
  </head>
  <body>
    <form action="wynik.php" method="post">
      <div>
        <p>Wprowadź dane:</p>
        <input type="text" name="dane[]"><br>
        <input type="text" name="dane[]"><br>
        <input type="text" name="dane[]"><br>
        <input type="submit" name="Submit" value="Wyślij">
      </div>
    </form>
  </body>
</html>
```

Trzy linie na dane. Używamy notacji tablicowej [] aby zaznaczyć użycie wielu wierszy tablicy





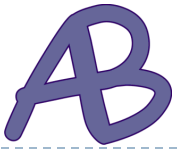
Przykład do wykonania

```
table
{
  border: none;
  margin-top: 20px;
  margin-left: 50px;
}
table td, table th, div
{
  font-size: 1.5rem;
  border: 1px solid rgb(130, 123, 114);
  padding: 10px;
  background-color: antiquewhite;
}
div {
  max-width: fit-content;
}
```

Po dodaniu stylu otrzymamy w efekcie:

Wprowadź dane:

Ten sam styl zastosujemy też do skryptu php



Z tablicy globalnej post pobieramy element o identyfikatorze dane

```
<?php
$tab = $_POST['dane'];
if (is_array($tab) && (count($tab) > 0)) {
    echo "<table>";
    echo " <tr><th>Klucz</th><th>Wartość</th></tr>";
    foreach ($tab as $klucz => $wartosc) {
        echo "<tr>";
        if (empty($wartosc)) {
            echo "<td>$klucz</td><td><i>pusty</i></td>";
        } else {
            echo "<td>$klucz</td><td><i>$wartosc</i></td>";
        }
        echo "</tr>";
    }
    echo "</table>";
} else {
    echo "<i>pusty lub nieprawidłowy</i>";
}
?>
```

Upewniamy się, czy \$tab jest na pewno tablicą

Rozpoczęcie tabeli I wyświetlenie nagłówka

Wyświetlenie wszystkich par klucz/wartość z tabeli

Sprawdzamy czy wartość jest pusta jeżeli tak, informujemy o tym

Klucz	Wartość
0	<i>dane 01</i>
1	<i>pusty</i>
2	<i>dane III</i>



Literatura

W prezentacji użyto przykładów z książki:

- Żygłowicz Jerzy - PHP - Kompendium wiedzy, Helion

- <https://www.php.net>