

# *przydatne strony*

<http://phpkurs.pl/tablice/>

<http://php.net/manual/pl/book.array.php>

<http://www.programuj.com/artykuly/www/plikiphp.php>

<http://pl.wikibooks.org/wiki/PHP>

<http://phpkurs.pl/operacje-na-plikach/>

# Pętla for

```
for(wyrażenie_początkowe; warunek; wyrażenie_modyfikujące)
    instrukcja;//gdy jedna instrukcja

    //gdy kilka instrukcji:
for(wyrażenie_początkowe; warunek; wyrażenie_modyfikujące) {
    instrukcje;
}
```

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <title>petla for </title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h3>petla for </h3>
    <table border="2px">
      <?php
      for ($i = 1; $i < 14; $i++) {
        echo "<tr><td>$i</td></tr>";
      }
      ?>
    </table>
  </body>
</html>
```

petla for

1
2
3
4
5
6
7
8
9
10
11
12
13

# Pętla for

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <title>petla for </title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h1>petla for </h1>
    <?php
      $wysokosc = 13;

      for ($i = 1; $i <= $wysokosc; $i++) {
        for ($j = 1; $j <= $i ; $j++) {
          echo '*';
        }
        echo '<br>';
      }
    ?>
  </body>
</html>
```

**petla for**

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

# Pętla For

```
<body>
<h1>petla for </h1>
<table border="2px">
<tr>
  <th>a</th>

<?php
$podstawa = 10;
$wykladnik = 5;

for ($i = 0; $i <= $wykladnik ; $i++) {
  echo "<th>a<sup>$i</sup></th>\n";
}
?>

</tr>
<?php
for ($i = 2; $i <= $podstawa; $i++) {
  echo "<tr>\n";
  echo "<th>$i</th>\n";
  for ($j = 0; $j <= $wykladnik; $j++) {
    echo '<td>';
    echo pow($i, $j);
    echo "</td>\n";
  }
  echo "</tr>\n";
}
?>
</table>
</body>
```

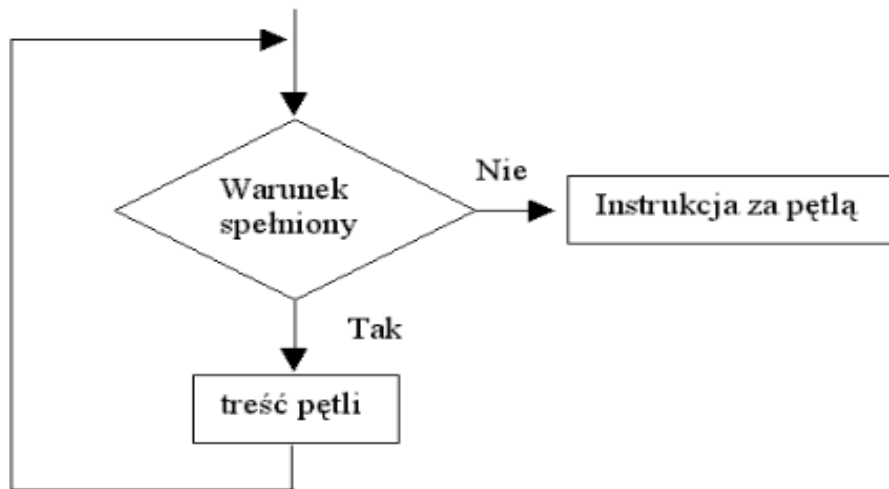
## petla for

a	a <sup>0</sup>	a <sup>1</sup>	a <sup>2</sup>	a <sup>3</sup>	a <sup>4</sup>	a <sup>5</sup>
2	1	2	4	8	16	32
3	1	3	9	27	81	243
4	1	4	16	64	256	1024
5	1	5	25	125	625	3125
6	1	6	36	216	1296	7776
7	1	7	49	343	2401	16807
8	1	8	64	512	4096	32768
9	1	9	81	729	6561	59049
10	1	10	100	1000	10000	100000

# Pętla while

```
while (warunek)
instrukcja; // gdy jeden warunek

//
while (warunek) {
instrukcja; // gdy więcej warunków:
}
```

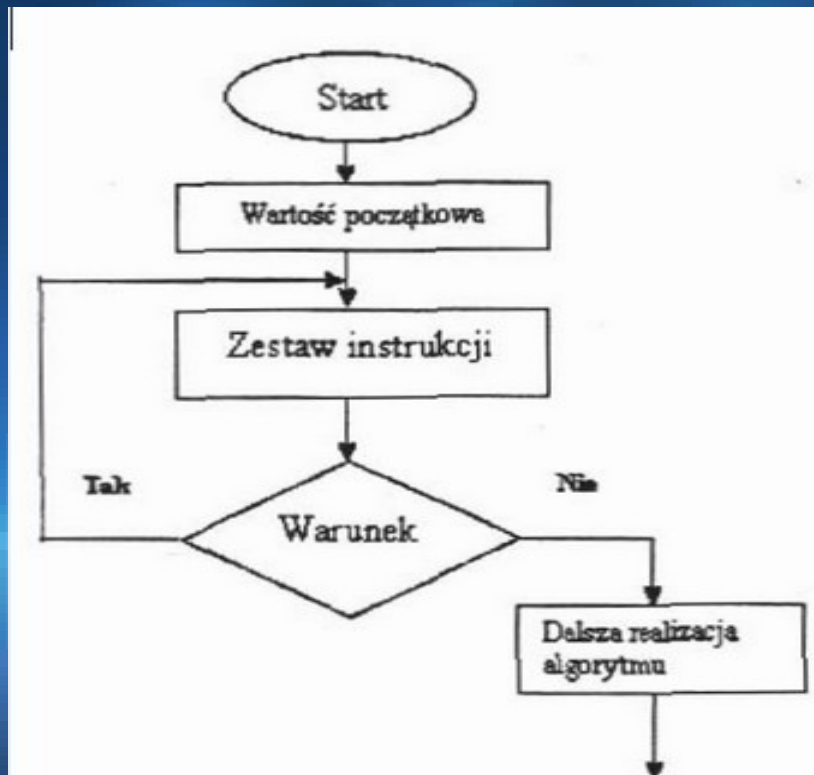


# Pętla do...while

```
do{  
instrukcje;  
}  
while(warunek) ;
```

```
<?php  
$x = 6;  
  
do {  
    echo $x.' <br>';  
    $x++;  
} while ($x<=10);  
?>
```

6  
7  
8  
9  
10



# Instrukcje: *break*, *continue*

- ▶ Instrukcja *break* pozwala na natychmiastowy „wyskok” z najbardziej zagnieżdżonej, dowolnej *instrukcji iteracyjnej* lub *instrukcji switch*.
- ▶ Instrukcja *continue* pozwala na przerwanie bieżącego przebiegu *dowolnej iteracji* i przejście do wykonania *następnego jej przebiegu od jego początku*

Dla iteracji **while** i **do-while** instrukcja **continue** powoduje przeniesienie sterowania do fazy testowania warunku kontynuacji. W przypadku iteracji **for** sterowanie przenoszone jest do wyrażenia modyfikującego a potem do testowania warunku.

```
<body>
<?php
    for ($i=0;$i<=10;$i++)
    {
        if ($i%3==0) continue;
        echo $i.'<br>';
    }
?>
</body>
```

1  
2  
4  
5  
7  
8  
10

```
<body>
<?php
    $liczba=6;
    $i=2;
    $p=true;
    if ($liczba<=1){echo 'liczba nie jest ani pierwsza ani złożona';}
    else {
        while ($i<$liczba) {
            if ($liczba%$i==0) {
                $p=false;
                break;
            }
            $i++;
        }
        if ($p==true) echo 'Liczba pierwsza';
        else echo 'Liczba złożona';
    }
?>
</body>
```

```
<?php
    $i=13;

    do {
        if ($i < 5) {
            echo 'i jest za małe';
            break;
        }

        if ($i > 15) {
            echo 'i jest za duże';
            break;
        }

        echo 'i jest poprawne';
    } while (0);
?>
```

# Tablice w PHP-uwagi ogólne

- Tablica przechowuje zbiór wartości, zgrupowanych i powiązanych ze sobą, niekoniecznie jednego typu. Jeśli elementem tablicy jest inna tablica, mamy do czynienia z tablicą wielowymiarową.
- W PHP każda tablica jest jednoznacznie scharakteryzowana przez swoją nazwę.
- Poszczególne, pojedyncze wartości, przechowywane w tablicy, noszą nazwę elementów tablicy.
- Każdy element tablicy posiada przypisany indeks (inaczej klucz), który umożliwia dostęp do tego elementu.
- W PHP elementy tablicy posiadają indeksy numeryczne, zaczynające się od zera. Indeksowanie od zera można zmienić.
- Aby odnieść się do elementu tablicy, należy użyć nawiasów kwadratowych. W nawiasach tych zawiera się indeks elementu tablicy. Taka konstrukcja, tj. nazwa tablicy z nawiasami kwadratowymi i wartością indeksu w ich wnętrzu, dokładnie odpowiada zwykłej zmiennej.
- Podobnie jak w przypadku zwykłych zmiennych, tablicę tworzymy przypisując wartości jej elementom – mówimy o tworzeniu tablicy „w locie”.
- Jeśli ma miejsce próba odniesienia się do elementu tablicy, który nie istnieje, w zależności od kontekstu PHP zwróci zero lub pusty łańcuch znakowy.
- W języku PHP istnieją również tablice asocjacyjne, w których indeksem elementu tablicy jest ciąg znaków.



# Tablice w PHP-tablice indeksowane numerycznie

Tego typu tablice są dostępne w większości języków programowania. Domyślnie, w Języku **PHP** – podobnie jak w C/C++, Javie i innych językach – indeks tablicy zaczyna się od zera, ale można tę opcję zmienić.

Zasadniczą zaletą tablic indeksowanych numerycznie jest możliwość użycia elementów tablicy w instrukcjach pętli. W ten sposób cała tablica wartości jest przetwarzana jako jeden element o tej samej nazwie, ale z różnymi wartościami indeksów (kluczy).

## Inicjowanie tablic:

### 1. Prosty (ręczny) sposób inicjowania tablicy

```
<body>
<?php
$stab=array('element1','element2',123,4,'kolejny element');
//wypiszemy elementy tablicy
$n=count($stab);
for($i=0;$i<$n;$i++){
    echo $stab[$i].<br>;
}
?>
</body>
```

array-konstruktor tablicy

count(\$nazwaTab)-zwraca ilość elementów tablicy

```
element1
element2
123
4
kolejny element
```

# Tablice w PHP-tablice indeksowane numerycznie

## Inicjowanie tablic:

2. Innym, równoważnym sposobem inicjowania tej samej tablicy może być nadanie wartości każdemu elementowi tablicy z osobna.

```
<body>
<?php
// Nowa tablica i element numer 0
// Kolejne elementy
$stab[0] = 'element1';
$stab[1] = 'element2';
$stab[2] = 123;
$stab[3] = 4;
$stab[4] = 'kolejny element';

//wypiszemy elementy tablicy
$n=count($stab);
    for($i=0;$i<$n;$i++){
        echo $stab[$i].'\n';
    }
//do wypisania elementów tablicy
//użyjmy instrukcji foreach
foreach($stab as $zmienna){
    echo $zmienna.'\n';
}
?>
</body>
```

```
element1
element2
123
4
kolejny element
```

**foreach (\$nazwaTab as \$zmienna) {  
Instrukcje do wykonania;  
}**

# Funkcje do badania wartości i typu zmiennych

W PHP występują trzy funkcje umożliwiające zbadanie typu i wartości zmiennej. Są nimi `var_dump()`, `var_export()` oraz `print_r()`. Funkcje te różnią się kilkoma szczegółami.

lp.	Funkcja	Składowe obiektów	Uwagi
1.	<code>var_dump()</code>	Tylko publiczne	-
2.	<code>var_export()</code>	Wszystkie (publiczne, chronione i prywatne)	Generowany wydruk jest poprawnym kodem PHP.
3.	<code>print_r()</code>	Wszystkie (publiczne, chronione i prywatne)	W przypadku zmiennych typu <code>string</code> , <code>integer</code> oraz <code>float</code> drukowana jest tylko wartość zmiennej (bez typu)

# Tablice w PHP-tablice indeksowane numerycznie

## Inicjowanie tablic:

3. Tablice można definiować nie podając numeru indeksu.

```
<body>
<?php
$tab2 [] = "Pierwszy";
$tab2 [] = "Drugi";
$tab2 [] = "Trzeci";
$tab2 [] = "Czwarty";
var_dump($tab2);
echo '<br>';
print_r($tab2);
?>
</body>
```

Wartość i typ zmiennej możemy sprawdzić za pomocą funkcji:  
**var\_dump(\$nazwaZmiennej).**  
Dotyczy to również tablicy.

Podobne działanie ma funkcja **print\_r(\$nazwaZmiennej)**, przy czym funkcja nie drukuje informacji o typie zmiennej, o ile typem jest string, integer lub float.

```
array(4) { [0]=> string(8) "Pierwszy" [1]=> string(5) "Drugi" [2]=> string(6) "Trzeci" [3]=> string(7) "Czwarty" }
Array ( [0] => Pierwszy [1] => Drugi [2] => Trzeci [3] => Czwarty )
```

# Tablice w PHP-tablice indeksowane numerycznie

4. Tablica utworzona z ciągu znaków:

**explode(separator,string,limit)**

Do rozdzielania ciągów na tablicę służy funkcja explode().

opcjonalnie

Jako pierwszy parametr trzeba do niej podać znak lub dłuższy ciąg który oddziela kolejne pola, jako drugi ciąg do rozdzielenia. Opcjonalnie można podać trzeci argument, który oznacza maksymalną liczbę pól – jeśli jest ich więcej niż ta liczba, to ostatnie pole będzie zawierało wszystkie pozostałe pola. Funkcja zwraca tablicę zawierającą kolejne

```
<?php
$tekstDoRozdzielenia='abc;sdf;234;56a';
$tab=explode(';', $tekstDoRozdzielenia);
foreach($tab as $a){
    echo $a.'<br>';
}
?>
```

```
abc
sdf
234
56a
```

# Tablice w PHP-zastosowanie funkcji range()

## Inicjowanie tablic:

4. Aby przechowywać w tablicy rosnącą sekwencję liczb całkowitych bądź znaków, można zastosować funkcję 'range()', która automatycznie utworzy tablicę.

**\$tab=range(min, max, step);**

**\$tab=range(min, max);**

Domyślnie step=1

```
<body>
<?php
echo 'pierwsza tablica <br>';
$tab1=range(2,10,3);
//wypiszemy elementy tablicy
$n=count($tab1);
for($i=0;$i<$n;$i++){
    echo $tab1[$i].<br>;
}
echo 'druga tablica <br>';
$tab2=range('A','Z',4);
//do wypisania elementów tablicy
//użyjemy instrukcji foreach
foreach($tab2 as $zmienna){
    echo $zmienna.<br>;
}
?>
</body>
```

pierwsza tablica  
2  
5  
8

druga tablica  
A  
E  
I  
M  
Q  
U  
Y

trzecia tablica

-3  
-2  
-1  
0  
1  
2

```
echo 'trzecia tablica<br>';
$tab3=range(-3,2);
//wypiszemy elementy tablicy
$n=count($tab3);
for($i=0;$i<$n;$i++){
    echo $tab3[$i].<br>;
}
```

# Tablice w PHP-ładowanie elementów z pliku-funkcja 'file()'

## Inicjowanie tablic:

5. Funkcja 'file()' z argumentem będącym nazwą pliku, zwraca tablicę, której każdy element odpowiada jednemu wierszowi zawartości pliku. Oznacza to, że każdy wiersz pliku staje się jednym elementem tablicy; inaczej mówiąc, element takiej tablicy to ciąg znaków, kończący się znakiem łamania wiersza.

**\$stab=file(nazwa\_pliku);//zabiera duzo pamieci**

```
<body>
<?php
echo 'Wiersze z pliku są elementami tablicy(dane z zadania maturalnego)<br>';
$stab=file('wypozyczenia.txt');
//wypiszemy elementy tablicy
$n=count($stab);
echo 'ilosc elementow tablicy ='.$n.'<br>';
for($i=0;$i<$n;$i++){
    echo $stab[$i].'\n';
}
?>
</body>
```

	ID_wyp	Data_wyp	ID_filmu	Pesel
1	1	2010-09-01	BB2005	82030192835
2	2	2010-09-09	CD2007	78062657877
3	3	2010-09-09	DD1983	77012657444
4	4	2010-09-10	AC2003	87100788765
5	5	2010-09-11	CD2008	70120675819
6	6	2010-09-12	DD2007	87100488725

Plik-'wypozyczenia.txt'

Wiersze z pliku są elementami tablicy(dane z zadania maturalnego)

ilosc elementow tablicy =501

```
ID_wyp Data_wyp ID_filmu Pesel
1 2010-09-01 BB2005 82030192835
2 2010-09-09 CD2007 78062657877
3 2010-09-09 DD1983 77012657444
4 2010-09-10 AC2003 87100788765
```

Samo wczytanie pliku do tablicy może być przydatne. Traktując plik jako bazę danych, w której każdy wiersz to jeden rekord, wczytanie takiego pliku od razu do tablicy oszczędza wiele pracy przy ręcznym rozbijaniu pliku na wiersze.

W dalszej części zajmiemy się formatowaniem wyników

# Tablice w PHP-obstługa plików -funkcja 'explode()'

## Cd obsługi pliku

Po załadowaniu tablicy, wskazane jest, by każdy wiersz został podzielony na elementy. Temu celowi służy funkcja 'explode()'. Explode pozwala na utworzenie nowej tablicy z poszczególnych elementów wiersza.

**explode(separator,string,limit)**

opcjonalnie

```
<body>
<table >
<?php
echo 'Wiersze z pliku są elementami tablicy(dane z zadania maturalnego)<br>';
$tab=file('wypozyczenia.txt');
//wypiszemy elementy tablicy
$n=count($tab);
for($i=0;$i<$n;$i++){
    $str='<tr>';
    $tab[$i]=explode("\t",$tab[$i]); //wiersz przekształcamy na tablicę
    for($j=0;$j<4;$j++){
        $str.='<td>'.$tab[$i][$j].'</td>';
    }
    $str.='</tr>';
    echo $str;
}
?>
</table>
</body>
```

cudzysłowy-cytowania dla znaków specjalnych.

ID_wyp	Data_wyp	ID_filmu	Pesel
1	2010-09-01	BB2005	82030192835
2	2010-09-09	CD2007	78062657877
3	2010-09-09	DD1983	77012657444
4	2010-09-10	AC2003	87100788765



# Tablice w PHP-obstuga plików -funkcja 'explode()'-drugi sposób

Operacje na elementach tablicy wykonujemy wykorzystując pętlę foreach. Nie sprawdzamy ilości elementów; pętla foreach sama to sprawdzi.

```
<table >
<?php
echo 'Wiersze z pliku są elementami tablicy(dane z zadania
maturalnego)<br>';
$tab=file('wypozyczenia.txt');//plik tniemy na wiersze
foreach($tab as $wiersz){
    $str='<tr>';
    $wiersz=explode("\t", $wiersz);//wiersz tniemy na komórki
    foreach($wiersz as $dane){
        $str.='<td>'.$dane.'</td>';
    }
    $str.='</tr>';
echo $str;
}
?>
</table>
```

ID_wyp	Data_wyp	ID_filmu	Pesel
1	2010-09-01	BB2005	82030192835
2	2010-09-09	CD2007	78062657877
3	2010-09-09	DD1983	77012657444
4	2010-09-10	AC2003	87100788765

# Tablice w PHP-obstuga plików -funkcja 'explode()'-sprawdzamy kod dla pliku tekstowego o danych nieregularnych

Nazwa pliku -jedyna zmiana w stosunku do poprzedniego skryptu

```
<table >
<?php
echo 'Zawartość pliku tekstowego, o dowolnych danych, oddzielonych
znakiem tabulacji<br>';
$stab=file('test.txt');//plik tniemy na wiersze
foreach($stab as $wiersz){
    $str='<tr>';
    $wiersz=explode("\t", $wiersz);//wiersz tniemy na komórki
    foreach($wiersz as $dane){
        $str.='<td>'.$dane.'</td>';
    }
    $str.='</tr>';
echo $str;
}
?>
</table>
```

test.txt

```
asdf    ref 34  s
wer sdf xs5 asw sa  23 a
23  34  xd
```

Zawartość pliku tekstowego, o dowolnych danych, oddzielonych znakiem tabulacji

asdf	ref	34	s	
wer	sdf	xs5	asw sa	23 a
23	34	xd		

# Tablice w PHP-tablice asocjacyjne

W PHP występuje też inny rodzaj tablic, tak zwane tablice asocjacyjne (zwane też czasem haszami – hash table). Są to tablice, w których zamiast indeksów liczbowych używa się identyfikatorów znakowych (kluczy).

```
$nazwaTab=array(  
    'klucz1'=>'wartosc1' ,  
    'klucz2'=>'wartosc2' ,  
    .....  
);
```

Dostęp do dowolnego elementu: `$nazwaTablicy['klucz']='wartosc';`

# Tablice w PHP-tablice asocjacyjne-przykład

```
<?php
$stolice = array(
    'Polska' => 'Warszawa',
    'Francja' => 'Paryż',
    'UK' => 'Londyn'
);
//wypisujemy elementy tablicy:
foreach($stolice as $klucz=>$wartosc ){
    echo 'państwo: '.$klucz.' stolica: '.$wartosc.'  
';
}
//dodamy kolejny element do tablicy:
$stolice['USA']='Waszyngton';
//sprawdzimy, czy element został dopisany do tablicy:
//wypisujemy elementy tablicy:
echo '<br>dodajemy kolejny element do tablicy:<br><br>';
foreach($stolice as $klucz=>$wartosc ){
    echo 'państwo: '.$klucz.' stolica: '.$wartosc.'  
';
}
?>
```

państwo: Polska stolica: Warszawa  
państwo: Francja stolica: Paryż  
państwo: UK stolica: Londyn

dodajemy kolejny element do tablicy:

państwo: Polska stolica: Warszawa  
państwo: Francja stolica: Paryż  
państwo: UK stolica: Londyn  
państwo: USA stolica: Waszyngton

# Tablice superglobalne PHP

Począwszy od wersji 4.1.0, PHP udostępnia zestaw predefiniowanych tablic, które zawierają zmienne serwera, zmienne środowiskowe oraz zmienne użytkownika. Tablice te są dość specyficzne, gdyż są one automatycznie globalne, tzn. automatycznie dostępne w każdym zasięgu. Dlatego nazywa się je "autoglobalami" albo "superglobalami". (W PHP nie ma mechanizmu pozwalającego użytkownikowi na definiowanie własnych zmiennych superglobalnych.)

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <title>tablice superglobalne</title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h3>tablice superglobalne</h3>
    <?php
      var_dump($_SERVER);
      var_dump($_GET);
    ?>
  </body>
</html>
```

# Tablice w PHP-tablice asocjacyjne-przykład

```
<!DOCTYPE html>
<html lang="pl">
  <head>
    <title>Przesyłanie formularzy - metoda GET</title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h3>Przesyłanie formularzy - metoda GET</h3>

    <?php
    if(isset($_GET['imie']) && isset($_GET['wiek'])) //sprawdzamy czy zmienne
    //istnieją (czy formularz został wysłany)
        echo 'Witaj <b>'.$_GET['imie'].'</b>! Masz <b>'.$_GET['wiek'].'</b> lat.';
    ?>

    <form action="przesylanieFormularzy.php" method="GET">
      <label>Podaj imię:<br/>
      |   <input type="text" name="imie"/>
      </label>

      <label>Ile masz lat:
      |   <input type="text" name="wiek"/>
      </label>

      <button type="submit">Wyślij</button>
    </form>
  </body>
</html>
```

## Przesyłanie formularzy - metoda GET

Witaj Alicja! Masz 17 lat.

Podaj imię:

Ile masz lat:

Wyślij