

Tömbök

A könyv előző részében a változót egy fiókhöz hasonlítottuk, amely egy értéket képes tárolni. Egy összetett tároló szerkezet a Tömb, amelyet egy fiókos szekrényhez hasonlítottunk. A fiókos szekrényben a fiókoknak sorszámuk van, és a fiókokban tárolódnak az értékek, továbbra is egy fiókban egy érték helyezkedik el.

A változó is a memóriában helyezkedik el és a tömb is a memóriában helyezkedik el, azaz hosszútávú tárolásra alkalmatlan.

5-6 értéket szeretnénk tárolni, az eddigi ismereteinkkel, 5-6 változót kellene létrehozni és kezelni, nehéz lenne egy egységként kezelni. Szeretnénk egy táblázat értékeit egy egységként kezelni, egy ciklusban kiíratni ehhez tömböt kell létrehozni. Egy ciklusban nem lehet 5 különböző változót kezelni, ez csak egyenként menne, ami nem egy hatékony kifejezési mód.

Tömb létrehozása

```
<?php
// Első mód
$tomb = array(5,6,7,8,9);
// Másik mód
$tomb1 = [5,6,7,8,9];
```

Mindkét változat megfelelő, az első kicsit hosszabb és többet kell írni hozzá.

Tömbhöz elemek hozzáadása

```
<?php
//üres tömb inicializálása
$tomb = [];
$tomb[] = 5;
$tomb[] = 76;
```

A fenti példában az első sorban felmerül az üres tömb fogalma, tömb szerkezet, de nincs egy eleme sem. Az azt követő sorokban a tömb elemeinek megadásakor automatikusan sorszámozódik a tömb indexe.

Tömb feltöltése `array_fill` függvény segítségével. Ez a függvény `tól-ig` indexek alapján egyforma értékekkel tölt fel egy tömböt. Ezt akkor érdemes használni, ha nagy mennyiségű azonos értékkel töltünk fel egy tömböt, pl.: sok nullával.

```
<?php
$a = array_fill(5, 6, 'banana');
print_r($a);
```

Tömb elemeinek elérése

```
<?php
$tomb = [4,5,64,12];
print($tomb[3]);
```

Tömb méretének lekérése count függvénnyel.

```
<?php
$tomb = [4,5,64,12];
print(count($tomb));
```

Tömb utolsó elemének lekérése a count függvény segítségével.

```
<?php
$tomb = [4,5,64,12];
$utolsoIndex = count($tomb)-1;
print($tomb[$utolsoIndex]);
```

Figyeljük meg, hogy egyet ki kellett vonni a hosszából, mivel a tömbök 0 -tól sorszámozódnak, emiatt a hossz az utolsó elem utánra mutat, így mindig ki kell vonni belőle egyet.

Tömb elemeinek bejárása, for ciklussal vagy kifejezetten erre a célra való foreach ciklussal.

```
<?php
$tomb = [4,5,64,12];
for($index = 0; $index < count(); $index++){
    print($tomb[$index]);
}
```

```
<?php
$tombElemek = [4,5,64,12];
foreach($tombElemek as $key => $tombElem){
    print($tombElem);
}
```

Tömb elemeinek kiírása hibakeresés céljából var_dump függvénnyel.

```
<?php
$tombElemek = [4,5,64,12];
```

```
var_dump($tombElemek);
```

Tömb elem(ek) törlése.

Tömbök fajtái

A tömböket több féle csoportosításban is nézhetjük.

Sorszámozott → indexelt tömb

Címkézett → asszociatív tömb

Vegyes → indexelt és asszociatív tömb is egyszerre.

Másik lehetséges csoportosítás:

Egy dimenziós (vector)

Két dimenziós(mátrix)

Sok dimenziós (multi dimenziós)

Két dimenziós tömb

A két dimenziós tömböt több feladatra is lehet használni, táblázat tárolása, 2 dimenziós kép ábrázolására. A fenti tömböt fel lehet fogni, úgy hogy tömb a tömbben. A tömb egyik eleme egy másik tömb. A PHP eléggé rugalmas, az egyes tömbök akár eltérő hosszúságúak is lehetnek.

Két dimenziós tömb egy elemére hivatkozás:

```
<?php
$d2Tomb = [0=>[1,2,3], 1=>[4,5,6]];
print($d2Tomb[1][2]);
```

Eredmény: 6

A két dimenziós tömb méretét lekérdezve nem a teljes tömb méretét kapjuk meg, csak az egyik dimenzió hosszát.

Két dimenziós tömb mérete

```
<?php
$d2Tomb =[0=>[1,2,3], 1=>[4,5,6]];
print(count($d2Tomb));
```

Eredmény: 2

Ha a másik dimenzió hosszát akarnánk lekérdezni, akkor az egyik indexet meg kell adni, az abban elhelyezkedő tömb hosszát kapjuk meg.

```
<?php
$d2Tomb =[0=>[1,2,3], 1=>[4,5,6]];
print(count($d2Tomb[0]));
```

Eredmény: 3

Két dimenziós tömb bejárása

```
<?php
$d2Tomb =[0=>[1,2,3], 1=>[4,5,6]];
for($sor=0; $sor < count($d2Tomb); $sor++){
    for($oszlop = 0; $oszlop < count($d2Tomb[$sor]); $oszlop++){
        print($d2Tomb[$sor][$oszlop]);
    }
}
```

Tömbök lehetőségei (több dimenzió , kevert , hierarchia)

Gyakorlatok

Bankszámlaszám ellenőrzése

Taj szám ellenőrzése

- Alapvető programozási tételek
 - Összegzés
 - Megszámlálás
 - Eldöntés
 - Kiválasztás
 - Keresés
 - Maximumkiválasztás