

Dr. Holló Csaba, SZTE TTIK, Szoftverfejlesztés Tanszék

Programozási alapismeretek kurzus

Olvasási idő:
5 perc



Alapvető programozásnyelvi elemek C-ben és PHP-ban

II. Tömbök

A lecke célja A tömbök használatának jobb megértése.

Tudás A lecke megtanulása után a hallgató ismeri a tömbök felépítését, megvalósítását, és műveleteit.

Képesség A tömbökkel alapvető műveleteket végez (végigjárás, másolás).

Tömb

☐ Értékek rendezett sorozata.

14	2	7	66	0
----	---	---	----	---

☐ Matematikai definícióját lásd

tomb

Fekete István és Hunyadvári László: Algoritmusok és adatszerkezetek jegyzetében.

C-ben:

- deklarálni kell, pl. `int tomb[5];`
- a méretét (elemszámát) is meg kell adni.

PHP-ban:

- nem kell megadni a méretét, és
- „bármeddig” bővíthető további elemekkel
- kényelem \leftrightarrow kevésbé hatékony tárolás

Tömb

C-ben: egyforma típusú elemek. Miért van ez előírva?

- az egymás után eltárolt értékek egyforma nagyságú memóriát foglalnak →
- az index értékekből képlettel kiszámolható a hivatkozott elem címe.

PHP-ban: különböző típusú elemek is lehetnek benne.

Hogyan lehet ez?

- az elemeknek a címeit tárolja el.

Ráadásul PHP-ban string kulcsok is lehetnek.

- a kulcsokat is eltárolja.

Tömb elemeinek végigjárása

C-ben: ciklusváltozókkal végigjárjuk az egymás után következő indexeket:

```
for (i=0; i<m; i++)
```

```
    for (j=0; j<n; j++) t[i][j] = 5;
```

PHP-ban viszont:

- nem csak numerikus indexek vannak
- az egész indexek sem egymás után következnek → speciális ciklus: foreach

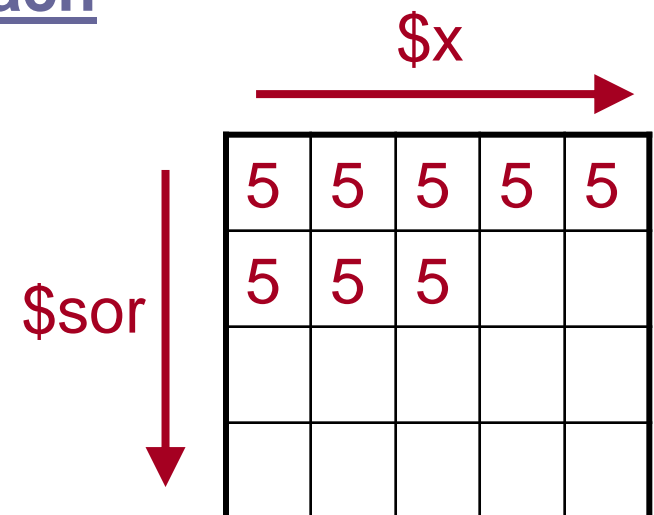
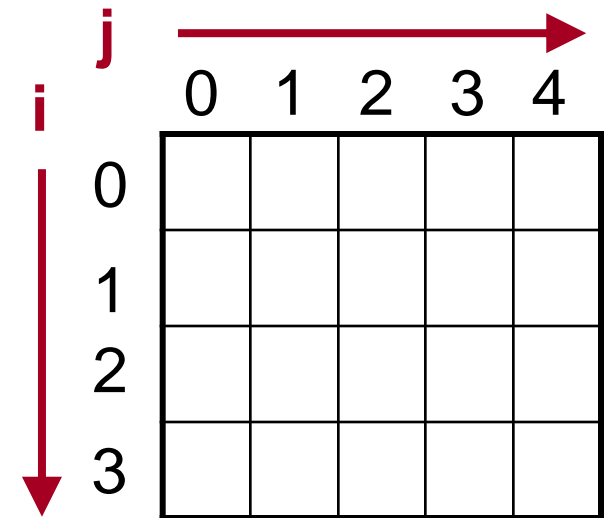
Ezért PHP-ban az elemek értékeit így

állítjuk át 5-re:

```
foreach ($t as &$sor)
```

```
    foreach($sor as &$x)
```

```
        $x = 5;
```



Kérdések, feladatok

1. Próbáljuk meg az előző kódot úgy lefuttatni, hogy a ciklusváltozók előtt nem használjuk az **&**-et, és figyeljük meg a különbséget!

Megoldás

- Az **&** használata nélkül az eredeti tömb sorai, illetve elemei átmásolódtak volna a **\$sor** és **\$x** változókba, a másolatok értékadása pedig nem módosította volna az eredeti tömböt.
- Az **&** használata esetén a **\$sor** és **\$x** változók cím szerinti értékadással az eredeti tömb értékeit módosítják.

Kérdések, feladatok

2. Milyen feltételek mellett tudjuk PHP-ban egy tömb elemeit foreach-től különböző ciklussal végigjárni?

- **Hogyha a kulcsok a kívánt sorrendben**
 - **valamilyen logika mentén legenerálhatók, vagy**
 - **a ciklussal végigjárható adatszerkezetben elvannak tárolva.**

Érdemes megnézni az alábbi oldalakat is:

- ❖ [Tömbök végigjárása részletesebben](#)
- ❖ [PHP tömbök részletesebben](#)
- ❖ [További érdekességek a C tömbökről](#)

Tömbműveletek

Legyenek $T1$, $T2$ tömbök.

Míg C-ben **nem** érvényes a ~~$T2 = T1$~~ értékadás,
addig PHP-ban **lehet** $\$T2 = \$T1$, sőt további
műveletek is vannak (pl. $\$T1 + \$T2$, $\$T1 == \$T2$).

Hogyha viszont C-ben a tömböt beletesszük egy
struktúra mezőjébe, akkor a struktúra
értékadása a tömböt is átmásolja:

```
struct ST{ int T[3][4]; } s1, s2;, nincs  $s2.T = s1.T$ ;
```

de $s2 = s1$; \rightarrow $s1.T$ -t átmásolja $s2.T$ -be

Kérdések, feladatok

3. Írjunk meg egy C programot, melyben teszteljük a struktúrákban levő tömbök átmásolását!
4. Hogyan tudnánk PHP-ban a $\$T1$ tömböt átmásolni $\$T2$ -be, ha nem lenne $\$T2 = \$T1$ művelet?

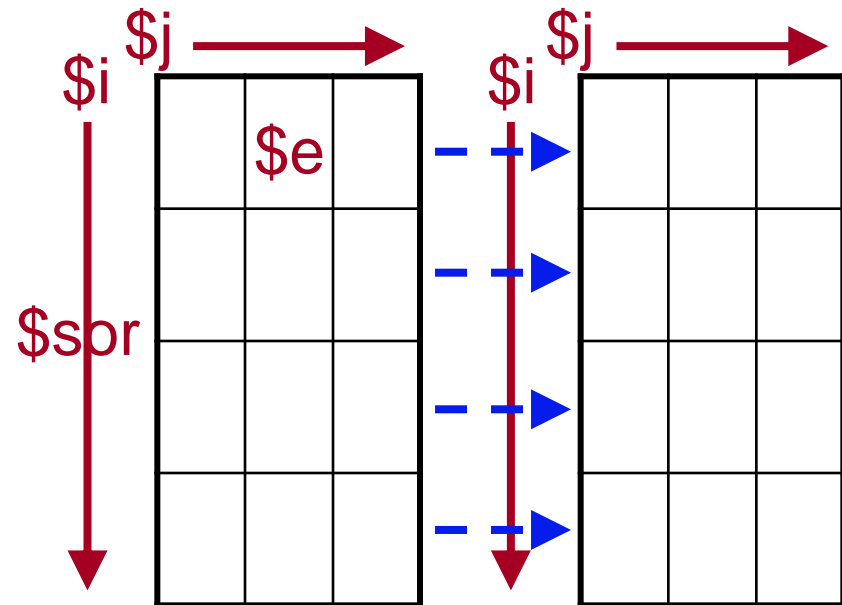
```
 $\$T2 = array();$ 
```

```
foreach ( $\$T1$  as  $\$i \Rightarrow \$sor$ ) {
```

```
     $\$T2[\$i] = array();$ 
```

```
    foreach ( $\$sor$  as  $\$j \Rightarrow \$e$ )
```

```
         $\$T2[\$i][\$j] = \$e;$     }
```



EFOP-3.4.3-16-2016-00014

AP1 HALLGATÓI DIPLOMA-SZERZÉST SEGÍTŐ SZOLGÁLTATÁSOK

Jelen tananyag a
Szegedi Tudományegyetemen
készült az
Európai Unió támogatásával.
Projekt azonosító:
EFOP-3.4.3-16-2016-00014



SZÉCHENYI 2020

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE