# TÖMBÖK C# NYELVEN

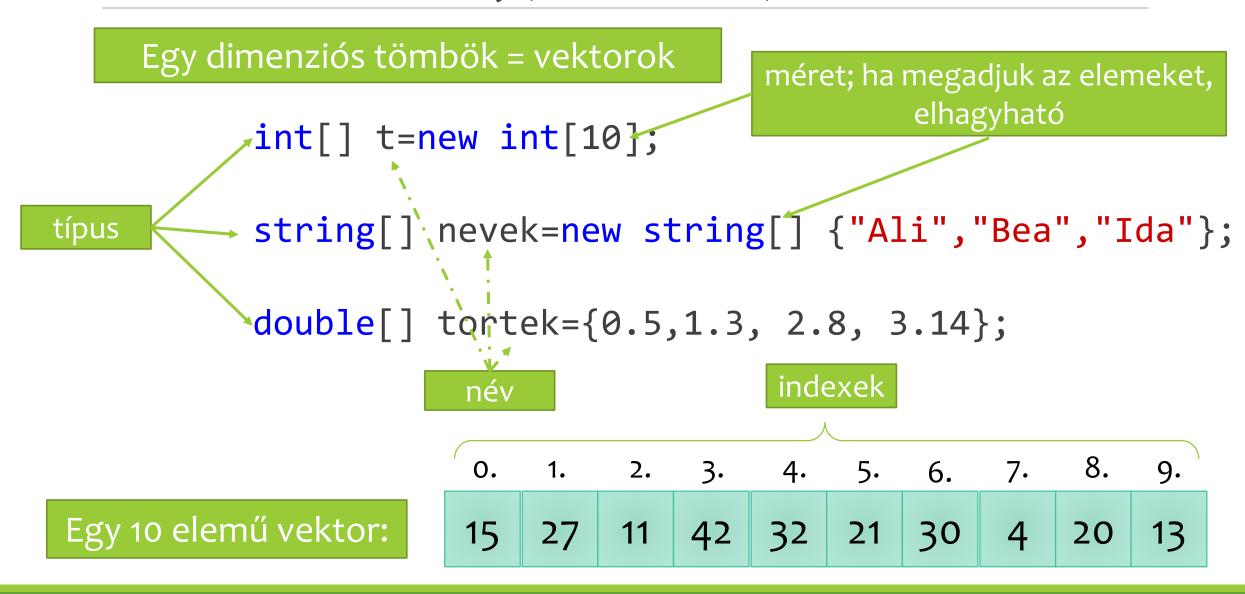
# Mire jók a tömbök?

A tömb olyan adatszerkezet, amely több azonos típusú elem tárolására alkalmas.

Statikus tömbök létrehozásakor megadjuk:

- Az elemek típusát
- A tömb dimenzióját
- A tömb nevét
- Az elemek számát
- Nem kötelezően: az elemek kezdőértékét

#### Tömbök deklarációja, definiálása, inicializálása



### A tömbelemek elérése

- A tömb elemeire az indexükkel hivatkozunk
- o Pl: t[0], nevek[2]
- A kezdőindex mindig 0
- Az utolsó elem indexe=elemszám-1
- A t tömb elemszáma: t.Length

# A tömb bejárása for ciklussal

```
for(int i=0; i<t.Length; i++)</pre>
    Console.Write(t[i]+" ");
/*végigmegyünk a tömb elemein, és kiírjuk egymás
mellé szóközzel elválasztva*/
```

# A tömb bejárása foreach ciklussal

```
foreach(int elem in t)
    Console.Write(elem+" ");
/*végigmegyünk a tömb elemein, és kiírjuk egymás
mellé szóközzel elválasztva*/
```

### A tömb rendezése

```
Array.Sort(t); /*növekvő sorrendbe rendezi a t
                 tömböt*/
Array.Sort(t1,t2); /*a t1 tömb, mint kulcs szerint
                   rendezi a t1 és t2 tömböt*/
Array.Sort(t,index,darab); /*a t tömb indexedik
                           elemétől kezdve a
                           megadott darab elemet
                           rendezi*/
```

## Tömb metódusok

```
t.Contains(elem); /*true vagy false értéket ad vissza
                   aszerint, hogy az elem megtalálható-e
                   a t tömbben*/
                  /*a t tömb elemeinek összege*/
t.Sum();
t.Average();
                  /*a t tömb elemeinek átlaga*/
t.Max();
                  /*a t tömb legnagyobb eleme*/
t.Min();
                  /*a t tömb legkisebb eleme*/
Array.IndexOf(t,elem); /*a megadott elem hányadik a t
                        tömbben; ha nincs benne, akkor -
                        1-et ad vissza*/
```

#### Több dimenziós tömbök

#### 2 dimenziós tömbök = mátrixok

t[0,0]	t[0,1]	t[0,2]
25	17	14
t[1,0]	t[1,1]	t[1,2]
5	10	8
t[2,0] 33	t[2,1] 4	t[2,2] <b>51</b>
t[3,0]	t[3,1]	t[3,2]
2	42	11