

21.) TUPLE TÍPUS

Most egy kevésbé használt típussal ismerkedünk meg. De vannak olyan feladatok, helyzetek, amikor kimondottan hasznos a **tuple típus** használata.

A legegyszerűbb megérteni a lényegét, ha a listából indulunk ki.

Lista:

rendezett, indexelet
egy elem többször szerepelhet
többféle típust tárolhat
meg lehet változtatni az elemek
db számát, értékét, sorrendjét

Tuple:

rendezett, indexelet
egy elem többször szerepelhet
többféle típust tárolhat
**NEM LEHET MEGVÁLTOZTATNI az elemek
értékét, sorrendjét, számát**



Tehát a **listához képest kevesebb képességgel rendelkezik**, akkor miért alkalmazzuk? Hát pont ezért, **néha ez a cél, hogy ne tudjunk rajta változtatni**. Néha szükségünk lehet olyan tárolóra, melynek az értékén nem tudunk változtatni.

Hogy mikor célszerű alkalmazni? Amikor egy komolyabb programot készítünk és az egész feladat alatt szükségünk va állandó, megváltoztathatatlan értékekre. Például fix koordináták használatánál, vagy RGB színek alkalmazásánál, vagy éppen felbontások beállításánál.

Nézzünk példákat:

- Képzeljünk el egy koordináta tengelyt, melyen megadunk két pontot. (koordinata_1, és koordinata_2) Úgy tudunk megadni egy tuple típust, hogy a név megadása után egyenlőségjelet teszünk, majd sima zárójelek között megadjuk az értékeket, vesszővel elválasztva. (1-2. sor)
- Írassuk ki egymás alá, majd nézzük meg a típusukat! (3-4. sor)
- Adjunk meg még két koordinátát az előzőek szerint, de most zárójelek nélkül! Látjuk, hogy így is működik. (5-6. sor)
- Nézzünk példát a tuple adattípus kicsomagolására!
- Adjunk a 11. sorban az x,y –nak a koordinata_1-et, és ha külön-külön sorban kiíratjuk az elemeket és megnézzük a típusait, láthatjuk, hogy „int” típusokat kapunk. (12-13. sor)
- Aztán nézzünk különböző típusú elemekkel feltöltött tuple-t! (15. sor)
- Írassuk ki a tuple-t (16. sor), majd nézzük meg a típusát (17. sor)
- Ahogy az elején megmondtuk, indexelhető is a tuple. Írassuk ki az egyik értéket! (18. sor)
- Végül próbáljuk meg felülírni a 3. indexű elemet! (20. sor)
- Futtassuk a programot, és látni fogjuk, hogy hibát jelez, hiszen a tuple típust nem lehet felülírni.

```

21a.py x
1   koordinata_1 = (1,2)
2   koordinata_2 = (5,4)
3   print(koordinata_1,koordinata_2, sep='\n')
4   print(type(koordinata_1),type(koordinata_2),sep='\n')
5   print('-----')
6   koordinata_3 = -3,-6
7   koordinata_4 = 2,-7
8   print(koordinata_3,koordinata_4, sep='\n')
9   print(type(koordinata_3),type(koordinata_4),sep='\n')
10  print('-----')
11  x,y = koordinata_1
12  print('x = ',x, type(x))
13  print('y = ',y, type(y))
14  print('-----')
15  kulon_tip_tuple = ('nevek',3,6.7,True)
16  print(kulon_tip_tuple)
17  print(type(kulon_tip_tuple))
18  print(kulon_tip_tuple[1])
19  print('-----')
20  kulon_tip_tuple[3]=False
21

```

```

Run: 21a x
E:\00_MM\12_Python_programozas\programok\venv\Scripts\python.exe E:/00_MM/12_Py
(1, 2)
(5, 4)
<class 'tuple'>
<class 'tuple'>
-----
(-3, -6)
(2, -7)
<class 'tuple'>
<class 'tuple'>
-----
x = 1 <class 'int'>
y = 2 <class 'int'>
-----
('nevek', 3, 6.7, True)
<class 'tuple'>
3
-----
Traceback (most recent call last):
  File "E:\00_MM\12_Python_programozas\programok\21a.py", line 20, in <module>
    kulon_tip_tuple[3]=False
TypeError: 'tuple' object does not support item assignment

```