

HALMAZOK

tuple

készítette: Vastag Atila
2020

TUPLE

A tuple egy *megváltoztathatatlan* (*immutable*) lista. A tuple létrehozása után semmilyen módon sem módosítható.

1. A tuple ugyanúgy definiálható, mint egy lista, kivéve hogy az elemek halmaza zárójelek és nem szögletes zárójelek közt van.
2. Egy tuple elemeinek meghatározott sorrendje van, mint a listák esetén. A tuple indexei nullától kezdődnek, mint a listák esetén, így egy nem üres tuple első eleme mindig 0 index-ű.
3. A negatív indexek a tuple végétől számítnak, mint a listák esetén.
4. A szeletelés is működik, mint a listák esetén. Lista szeletelésekor egy új listát kapsz, tuple szeletelésekor egy új tuple-t

```
tuple = ("a", "b", "mpilgrim", "z", "példa")
```

```
print(tuple)  
#('a', 'b', 'mpilgrim', 'z', 'példa')
```

```
item = tuple[0]  
print(item)  
#'a'
```

```
item = tuple[-1]  
print(item)  
#'példa'
```

```
slice = tuple[1:3]  
print(slice)  
# ('b', 'mpilgrim')
```

```
exists = "z" in tuple  
print(exists)  
#True
```

Akkor mire jók a tuple-ök?

- A tuple-ök gyorsabbak a listáknál. Ha értékek konstans halmazát definiálsz, és a végighaladáson kívül semmit sem akarsz vele csinálni, akkor használj tuple-t a lista helyett.
- A kódot biztonságosabbá teszi, ha a nem módosítandó adatokat „írásvédetté” teszed. A tuple használata lista helyett olyan, mintha egy közvetett assert utasításod lenne, amely jelzi, hogy ezek az adatok konstansak, és külön elhatározás (és egy bizonyos függvény) szükséges a felülbírálásához.
- Egyes tuple-ök használhatók szótárkulcsokként (konkrétan olyan tuple-ök, amelyek megváltoztathatatlan értékeket, például karakterláncokat, számokat és más tuple-öket tartalmaznak).

A tuple és a lista közti legnagyobb különbség, hogy a tuple nem módosítható. Technikai kifejezéssel élve a tuple-ök megváltoztathatatlanok. Gyakorlati szempontból nincsenek olyan metódusaik, amelyek lehetővé tennék a módosításukat. A listák rendelkeznek olyan metódusokkal, mint az **append()**, **extend()**, **insert()**, **remove()** és **pop()**. A tuple-ök ezek egyikével sem. A tuple szeletelhető (mert az egy új tuple-t hoz létre), és ellenőrizheted, hogy egy tuple tartalmaz-e egy adott értéket (mert ez nem változtatja meg a tuple-t), és ez minden.

Használhatod az **in** operátort is egy elem meglétének ellenőrzésére a tuple-ben.

Több érték hozzárendelése egyszerre

Itt egy menő programozási trükk: Pythonban egy tuple használatával egyszerre több értéket is hozzárendelhetsz változókhoz.

```
tuple = ('a', 2, True)
(x, y, z) = tuple
```

```
print(x)
#a'
```

```
print(y)
2
```

```
print(z)
True
```

A `v` egy három elemű tuple, az `(x, y, z)` pedig egy három változót tartalmazó tuple. Az egymáshoz rendelésükkel a `v` értékei a változókhoz lesznek rendelve, a megadott sorrendben.

Ezt sok mindenre fel lehet használni.

Tegyük fel, hogy neveket szeretnél rendelni egy értéktartományhoz. A beépített **range()** függvényt többváltozós értékadásban az egymást követő értékek hozzárendeléséhez használhatod.

```
(HÉTFŐ, KEDD, SZERDA, CSÜTÖRTÖK, PÉNTEK, SZOMBAT, VASÁRNAP) = range(7)
```

```
print(HÉTFŐ)  
#0
```

```
print(KEDD)  
#1
```

```
print(VASÁRNAP)  
#6
```

A beépített **range()** függvény egészek sorozatát állítja elő. (Technikailag a **range()** függvény egy **iterátort** ad vissza, nem pedig listát vagy **tuple**-t). A HÉTFŐ, KEDD, SZERDA, CSÜTÖRTÖK, PÉNTEK, SZOMBAT és VASÁRNAP a definiált változók.

Most minden változó rendelkezik egy értékkel: a HÉTFŐ a 0, a KEDD 1 és így tovább.