

# HALMAZOK dictionary

készítette: Vastag Atila  
2020

# DICTIONARY

*A dictionary*, azaz szótár, kulcs-érték párok rendezetlen halmaza. Amikor a szótárhoz hozzáadsz egy kulcsot, akkor a kulcs értékét is meg kell adnod. (Az érték később bármikor módosítható.) A Python szótárai az ismert kulcsú értékek lekérésére vannak optimalizálva, de ez nem működik fordítva.

A szótárai egy másik összetett adattípust képeznek. Bizonyos fókig a listákra emlékeztetnek (módosíthatók mint a listák), de nem szekvenciák. A bejegyzett elemek nem megváltoztathatatlan sorrendben lesznek elrendezve. Egy **kulcs**nak nevezett speciális indexszel ami lehet alfabetikus, numerikus vagy bizonyos feltételek mellett egy összetett típus bármelyikükhöz hozzá férhetünk.

Egy szótárban, a listákhoz hasonlóan, bármilyen típusú elemeket tárolhatunk. Az elemek lehetnek : numerikus, string, lista, tuple, szótárértékek, de függvények, classok, vagy objektumok is (lásd később).

Példaként egy nyelvi szótárat hozunk létre, ami angol informatikai kifejezések magyarra fordítására szolgál. Ebben a szótárban a **kulcsok** karakterláncok lesznek.

```
from typing import Dict
```

```
alaktreszek: Dict[str, str] =
```

```
{'computer': 'számítógép', 'keyboard': 'billentyűzet', 'mouse': 'egér'}
```

A kimeneten megfigyelhetjük : formailag egy szótár elemek vesszővel elválasztott sorozataként jelenik meg (amit kapcsos zárójelek vesznek körül). Mindegyik elem egy objektumpárból áll : egy indexből és egy értékből, amiket kettőspont választ el.

```
from typing import Dict
```

```
alaktreszek: Dict[str, str] =  
{'computer': 'számítógép', 'keyboard': 'billentyűzet', 'mouse': 'egér'}
```

Az indexeket **kulcs**nak, az elemeket pedig **kulcsérték párok**nak hívjuk. Megállapíthatjuk, hogy a kiírt sorrend, amiben az elemek megjelennek, nem felel meg annak a sorrendnek, amiben megadtuk őket. Ennek semmilyen komoly következménye sincs: értékhez sose fogunk numerikus indexszel hozzáférni a szótárban. Ehelyett kulcsokat fogunk használni :

```
print alaktreszek['mouse']  
#egér
```

Mivel a szótártípus egy módosítható típus, kezdhetünk egy üres szótár létrehozásával, amit majd apránként töltünk fel. Szintaktikai szempontból annak alapján ismerünk fel egy adatstruktúrát szótár típusúként, hogy az elemei kapcsos zárójelbe vannak zárva. Tehát a `{ }` egy üres szótárat fog jelölni :

```
dictionary = {} #üres szótár
```

Elem törlésére a **del** utasítást használjuk.

Példaként hozzunk létre egy másik szótárat, ami most egy gyümölcsraktár leltárát fogja tartalmazni. A kulcsok (vagy indexek) a gyümölcsnevek lesznek és az elemek értékei a raktározott gyümölcsök tömegei lesznek (ez alkalommal az értékek numerikus típusúak).

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274,
'barack' : 137}
print leltar
#{'narancs': 274, 'alma': 430, 'banan': 312, 'barack': 137}
```

Ha a tulajdonos az almakészlet felszámolásáról dönt, akkor a szótárból eltávolíthatjuk ezt a bejegyzést :

```
del leltar['alma']
print leltar
#{'narancs': 274, 'banan': 312, 'barack': 137}
```

## Műveletek szótárakkal

- **len()** függvény használható szótárral : az elemek számát adja meg.

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274,  
'barack' : 137}
```

```
elemekSzama : int = len(leltar)
```

```
print(elemekSzama)
```

```
#4
```

- **keys()** metódus a szótárban használt kulcsok listáját adja meg :
- **values()** metódus a szótárban tárolt értékek listáját adja meg:

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274,  
'barack' : 137}
```

```
kulcsok: List[str] = leltar.keys()
```

```
ertekek: List[int] = leltar.values()
```

```
print(kulcsok)
```

```
print(ertekek)
```

```
#['alma', 'banan', 'narancs', 'barack']
```

```
#[430, 312, 274, 137]
```

## Műveletek szótárakkal

- **has\_key()** metódus megadja, hogy a szótár tartalmaz-e egy meghatározott kulcsot.

A kulcsot argumentumban adjuk meg és a metódus egy «igaz» vagy «hamis» értéket ad vissza (valójában 1et vagy 0t), annak megfelelően, hogy a kulcs létezik, vagy sem.

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274, 'barack' : 137}
```

```
if (leltar.has_key("alma")):  
    print("van almánk")  
else:  
    print("bocsánat, de nincs almánk")
```



## Műveletek szótárakkal

- **items()** metódus a szótárból egy vele egyenértékű, tuplekből álló listát készít :

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274,  
'barack' : 137}
```

```
print leltar.items()
```

```
#[(‘alma', 430), ('narancs', 274), ('banan', 312), ('barack', 137)]
```

## Műveletek szótárakkal

- **copy()** metódussal a szótár valódi másolatát készíthetjük el. Tudnunk kell, hogy egy létező szótárnak egy új változóhoz való hozzárendelése csak egy új hivatkozást hoz létre ugyanarra az objektumra, nem pedig egy új objektumot (aliasing). Például az alábbi utasítás (a látszat ellenére) nem definiál új szótárat :

```
leltar: Dict[str, int] = {'alma': 430, 'banan': 312, 'narancs' : 274,
'barack' : 137}
készlet = leltar
print(készlet)
#{'narancs': 274, 'alma': 430, 'banan': 312, 'barack': 137}
```

Ha módosítjuk az leltar at, akkor a készlet is módosul, és viszont (ez a két név valójában ugyanazt a szótárobjektumot jelöli a számítógép memóriájában) :

```
del leltar['banan']
print(készlet)
#{'narancs': 274, 'alma': 430, 'barack': 137}
```

## Műveletek szótárakkal

Egy már létező szótár valódi (független) másolatának elkészítéséhez a **copy()** metódust kell használni :

```
leltar: Dict[str, int] = {'alma': 430, 'narancs' : 274, 'barack' : 137}
```

```
bolt = leltar.copy()
```

```
bolt['szilva'] = 561
```

```
print(bolt)
```

```
#{'narancs': 274, 'alma': 430, 'barack': 137 , 'szilva': 561}
```

```
print(leltar)
```

```
#{'narancs': 274, 'alma': 430, 'barack': 137}
```

# Szótár bejárása

A for ciklust felhasználhatjuk a szótárban lévő elemeket egymás után történő kezelésére, de vigyázat :

- Az iterráció során a kulcsok lesznek egymás után hozzá rendelve a ciklusváltozóhoz (**kulcs**), nem pedig az értékek.
- Nem lehet előre látni az elemekhez való hozzáférés sorrendjét (mivel a szótár nem szekvencia).

```
keszlet: Dict[str, int] = { "narancs" : 274, "barack" : 137, "banan" : 312 }
```

```
for kulcs in keszlet:
```

```
    print(kulcs)
```

```
#barack banan narancs
```

Ha az értékeket akarjuk kezelni, akkor a megfelelő kulcsok segítségével férünk hozzájuk :

```
for kulcs in keszlet:
```

```
    print(keszlet[kulcs])
```

```
#274 137 312
```

## Szótár bejárása

Ez az eljárás teljesítményben kifejezve sem, és az olvashatóság szempontjából sem ideális. Inkább az előzőekben leírt **items()** metódus hívása a javasolt.

```
keszlet: Dict[str, int] = { "narancs" : 274, "barack" :137, "banan" :  
312}
```

```
for kulcs, ertek in leltar.items():  
    print(kulcs, ertek)
```

```
#barack 137 banana 312 narancs 274
```