Adatszerkezetek II. – dict, set

Dictionary: kulcs–érték párok gyűjteménye, mint egy valódi szótár. Hasonlít a C nyelv struktúrájára.

- Kapcsos zárójel: {}
- Kulcs és érték kettősponttal elválasztva
- Kulcsok egyediek
- print(szemely[“nev”])

```
szemely = {
    "nev": "Anna",
    "kor": 25,
    "varos": "Budapest"
}
```

Példa: Hozz létre egy dictionary-t saját adataiddal (név, kor, kedvenc szám). Írd ki külön a nevet. Írd ki az egész dictionary-t.

```
adat = {
    "nev": "Anna",
    "kor": 20,
    "kedvenc_szam": 7
}

print("Név:", adat["nev"])
print("Teljes adat:", adat)
```

Módosítás és bővítés

- Új kulcs hozzáadása
- Érték módosítása

```
szemely["kor"] = 26
szemely["foglalkozas"] = "mérnök"
```

Példa: Hozz létre egy üres dictionary-t. Kérd be a felhasználó nevét és életkorát. Tárold el dictionary-ben. Írd ki az eredményt.

```
adat = {}

nev = input("Név: ")
kor = int(input("Kor: "))
```

```
adat["nev"] = nev
adat["kor"] = kor

print(adat)
```

Dictionary bejárása

```
szemely = {"nev": "Anna", "kor": 25}

for kulcs in szemely:
    print(kulcs, szemely[kulcs])
#Kulcs–érték együtt:
for kulcs, ertek in szemely.items():
    print(kulcs, ertek)
```

Példa: Hozz létre egy dictionary-t 3 tantárggyal és jeggyel. Írd ki az átlagot.

```
jegyek = {
    "matek": 4,
    "fizika": 5,
    "angol": 3
}

osszeg = 0
for ertek in jegyek.values():
    osszeg += ertek

atlag = osszeg / len(jegyek)
print("Átlag:", atlag)
```

Dict fő metódusai

Metódus	**Funkció**	Példa
dict.keys()	Visszaadja a kulcsok listáját (view object)	d.keys()
dict.values()	Visszaadja az értékek listáját	d.values()
dict.items()	Visszaadja a kulcs-érték párokat	d.items()
dict.get(key[, default])	Visszaadja a kulcshoz tartozó értéket, ha nincs → default	d.get(“x”, 0)
dict.pop(key[, default])	Törli a kulcsot és visszaadja az értékét	d.pop(“x”)
dict.popitem()	Törli és visszaad egy tetszőleges kulcs-érték párt	d.popitem()
dict.update([other])	Másik szótár vagy iterable alapján frissít	d.update({“y”:10})
dict.clear()	Törli az összes elemet	d.clear()
dict.setdefault(key[, default])	Ha nincs kulcs, létrehozza alapértelmezett értékkel	d.setdefault(“x”,0)
dict.copy()	Felületes másolat készít	d2 = d.copy()

Set alapjai

Set: egyedi elemek gyűjteménye, halmaza.

- Nincs indexelés
- Nem tartalmaz duplikációt

```
szamok = {1, 2, 3, 3, 2}
print(szamok)
```

Példa: Kérj be 5 számot. Tárold el listában. Alakítsd át set-té. Írd ki az egyedi elemeket

```
szamok = []

for i in range(5):
    szam = int(input("Adj meg egy számot: "))
    szamok.append(szam)

egyedi = set(szamok)
print("Egyedi elemek:", egyedi)
```

Set fő metódusai

set.add(x)	**Hozzáad egy elemet**	s.add(5)
set.update(iterable)	Több elemet ad hozzá	s.update([1,2,3])
set.remove(x)	Eltávolítja az elemet (ha nincs → KeyError)	s.remove(2)
set.discard(x)	Eltávolítja az elemet, ha van (hiba nélkül)	s.discard(2)
set.pop()	Eltávolít egy tetszőleges elemet	s.pop()
set.clear()	Összes elem törlése	s.clear()
set.copy()	Felületes másolat	s2 = s.copy()
set.union(other)	Egyesítés	s.union({4,5})
set.intersection(other)	Metszet	s.intersection({2,3})
set.difference(other)	Különbség	s.difference({1,2})
set.symmetric_difference(other)	Szimmetrikus különbség	s.symmetric_difference({1,2})
set.issubset(other)	Részhalmaz-e	s.issubset({1,2,3})
set.issuperset(other)	Felülhalmaz-e	s.issuperset({1})

Összefoglaló feladatok

1. feladat Írj programot, ami:

- Bekér 5 nevet
- Minden névhez bekér egy pontszámot
- Dictionary-ben tárolja
- Kiírja a legmagasabb pontszámot elért nevet

```
eredmenyek = {}

for i in range(5):
    nev = input("Név: ")
    pont = int(input("Pontszám: "))
    eredmenyek[nev] = pont

max_nev = None
max_pont = -1

for nev, pont in eredmenyek.items():
    if pont > max_pont:
        max_pont = pont
        max_nev = nev

print("Legjobb eredmény:", max_nev, max_pont)
```

2. feladat: Van egy dict, ahol a kulcs a diák neve, az érték egy set az általa elvégzett tantárgyak jegyeivel.

- Adj hozzá egy új tantárgyat minden diáknak
- Ellenőrizd, hogy van-e 5-ös jegye minden diáknak
- Számold meg, hány diák kapott 1-es jegyet

```
diakok = {
    "Anna": {5, 4, 3},
    "Bence": {2, 3, 5},
    "Csaba": {1, 2, 3}
}

# Új tantárgy 4-es jeggyel minden diáknak
for nev, jegyek in diakok.items():
    jegyek.add(4)

# Ellenőrzés: van-e 5-ös jegy
for nev, jegyek in diakok.items():
    if 5 in jegyek:
        print(f"{nev} kapott 5-öst")

# 1-es jegyek száma
egyessel = sum(1 in jegyek for jegyek in diakok.values())
```

```
print("1-es jegyet kapott diákok száma:", egyessel)
```

3. feladat: Van egy bolt dict-ben a termékek és mennyiségek formájában.

- Egy új szállítmány érkezik egy másik dict-ből → frissítsd a raktárt
- Távolítsd el azokat a termékeket, amelyek elfogytak (0)
- Írd ki a teljes készletet

```
raktar = {"alma": 10, "korte": 5, "banan": 0}
szallitas = {"alma": 5, "banan": 7, "narancs": 3}

# Raktár frissítése
for termek, db in szallitas.items():
    raktar[termek] = raktar.get(termek, 0) + db

# Elfogyott termékek törlése
torlendo = [k for k,v in raktar.items() if v == 0]
for k in torlendo:
    raktar.pop(k)

print("Aktuális készlet:", raktar)
```

4. feladat: Adott két set a bolt A és B kínálatával.

- Írd ki, mely termékek vannak mindkét boltban (metszet)
- Melyek vannak csak az egyik boltban (szimmetrikus különbség)
- Egyesítsd a készleteket

```
boltA = {"alma", "banan", "narancs"}
boltB = {"korte", "banan", "narancs", "szilva"}

print("Mindkettőben:", boltA.intersection(boltB))
print("Csak az egyikben:", boltA.symmetric_difference(boltB))
boltA.update(boltB)
print("Összes termék:", boltA)
```

5. feladat: Egy dict dolgozó neve → set projektek.

- Adj hozzá új projektet minden dolgozónak
- Töröld azokat a projekteket, amelyeken senki sem dolgozik
- Írd ki a dolgozónkénti projektlistát

```
dolgozok = {
    "Eva": {"A", "B"},
```

```
    "Gabor": {"B", "C"},
    "Ilona": {"C"}
}

# Új projekt D mindenkihez
for projektek in dolgozok.values():
    projektek.add("D")

# Ellenőrizzük, mely projektek senkinél sem szerepelnek
osszes = set().union(*dolgozok.values())
nem_aktiv = {"E", "F"} - osszes
print("Nem aktív projektek:", nem_aktiv)

print("Dolgozók projektjei:")
for nev, projektek in dolgozok.items():
    print(nev, ":", projektek)
```

6. feladat: Adott egy szöveg string.

- Készíts dict-et, ahol a kulcs a szó, az érték a darabszám
- Távolítsd el a ritkán előforduló szavakat (pl. egyszer előforduló)
- Írd ki a leggyakoribb szót

```
szoveg = "python lista ciklus python python adatszerkezet lista ciklus"

szavak = szoveg.split()
szavak_szam = {}

for szo in szavak:
    szavak_szam[szo] = szavak_szam.get(szo, 0) + 1

# Ritka szavak törlése
for szo in list(szavak_szam.keys()):
    if szavak_szam[szo] == 1:
        szavak_szam.pop(szo)

# Leggyakoribb szó
leggyakoribb = max(szavak_szam, key=szavak_szam.get)
print("Leggyakoribb szó:", leggyakoribb)
print("Szó előfordulások:", szavak_szam)
```

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:muszaki_informatika:adatszerkezetek_ii._dict_set_gyakorlo_feladatok

Last update: **2026/02/19 19:56**