



# ALGORITMUSOK ÉS ADATSZERKEZETEK

Python 6.

Oktatási hét	Témakör - Gyakorlat
1	Bevezetés a Python programozási nyelvbe, alapvető szintaxis és kifejezések, változók, típusok és alapvető műveletek, alapvető be- és kimenet, feltételes elágazások (if, elif, else)
2	Ciklusok (for, while), listák és egyszerű iterációk
3	Függvények és modulok, függvények definiálása és hívása, paraméterek és visszatérési értékek, beépített modulok használata, fájlkezelés, kivételkezelés
4	Algoritmusok és optimalizálási stratégiák rekurziós példán keresztül (brute force, dinamikus programozás, mohó algoritmusok, visszalépéses keresés)
5	Adatszerkezetek I. - Lista, tömb, sor, verem, listaműveletek, queue és stack implementációja statikusan (tömb) és dinamikusán (lista), gyakorlati feladatok
6	Adatszerkezetek II. – Halmaz és szótár, halmazműveletek és szótárműveletek, gyakorlati feladatok
7	<b>Zárthelyi dolgozat 1.</b>
8	Adatszerkezetek III. – Láncolt lista implementálása
9	Adatszerkezetek IV. – Bináris keresőfa implementálása
10	Gráfok alapjai, reprezentációik, gráfalgoritmusok implementálása
11	<i>Rektori szünet</i>
12	<i>Féléves beadandó feladat leadása és bemutatása</i>
13	<b>Zárthelyi dolgozat 2.</b>
14	<b>Félév zárása, javító/pótló zárthelyi dolgozat</b>





# Halmaz és szótárműveletek

*Gyakorlati feladatok*

# HALMAZOK {}, HALMAZMŰVELETEK

- **Rendezetlen, egyedi elemek:** A halmazokban az elemek nincsenek sorrendben, és minden elem csak egyszer szerepelhet.
- **Halmazműveletek:** Egyesítés, metszet, különbség, szimmetrikus különbség.
- **Felhasználási területek:**
  - **Adatok tisztítása:** Duplikált elemek eltávolítása.
  - **Egyedi elemek meghatározása:** Egy listában található egyedi elemek megkeresése.
  - **Közös elemek keresése:** Két halmaz közös elemeinek meghatározása.
  - **Különbségek meghatározása:** Két halmaz közötti különbségek meghatározása.
  - **Algoritmusok:** Gráfelméleti algoritmusokban, például az összefüggőségi komponensek meghatározásában.

 python\_05\_01\_halmazmuveletek.py ×

```

1  # Létrehozunk egy halmazt
2  gyumolcsok = {"alma", "banán", "cseresznye", "dinnye", "eper"}
3  print("Csak gyümölcsök: ", gyumolcsok)
4
5  # Írjuk ki a halmaz elemeinek számát!
6  print("gyumolcsok elemeinek a száma: ", len(gyumolcsok))
7
8  # Az elemek nem indexelhetők.
9  #print(gyumolcsok[1])
10 print("gyumolcsok elemenként: ")
11 # Menjünk végig az elemeken!
12 for x in gyumolcsok:
13     print(x, end="-")

```

```

Csak gyümölcsök: {'banán', 'dinnye', 'cseresznye', 'eper', 'alma'}
gyumolcsok elemeinek a száma: 5
gyumolcsok elemenként:
banán-dinnye-cseresznye-eper-alma-

```



# HALMAZMŰVELETEK

```

16 # Halmaz tagság
17 # Nézzük meg, benne van-e a dinnye a halmazban!
18 print("dinnye van a gyumolcsok halmazban?")
19 print("dinnye" in gyumolcsok) # True
20 # Nézzük meg, benne van-e a barack a halmazban!
21 print("barack van a gyumolcsok halmazban?")
22 print("barack" in gyumolcsok) # False
23 # Nézzük meg, nincs-e benne a barack a halmazban!
24 print("barack nincs a gyumolcsok halmazban?")
25 print("barack" not in gyumolcsok) # True
26

```

Csak gyümölcsök: {'banán', 'dinnye', 'cseresznye', 'eper', 'alma'}

```

dinnye van a gyumolcsok halmazban?
True
barack van a gyumolcsok halmazban?
False
barack nincs a gyumolcsok halmazban?
True

```

# HALMAZMŰVELETEK

```
27 # Halmazműveletek
28 # Vegyük a zöldség és gyümölcs halmaz unióját!
29 zoldsegek = {"krumpli", "répa", "sütőtök"}
30 print("Csak zöldségek: ", zoldsegek)
31 novenyek = gyumolcsok.union(zoldsegek)
32 novenyek2 = gyumolcsok | zoldsegek
33 print("Növények halmaz (unió): ", novenyek)
34 print("Növények2 halmaz (unió): ", novenyek2)
35
36 # Vegyük a gyümölcs és déli gyümölcs halmaz metszetét!
37 deli_gyumolcsok = {'citrom', 'banán', 'narancs', 'mangó'}
38 print("Déli gyümölcsök: ", deli_gyumolcsok)
39 metszet = gyumolcsok.intersection(deli_gyumolcsok)
40 metszet2 = gyumolcsok & deli_gyumolcsok
41 print("Gyümölcsök közül ami déli (metszet): ", metszet)
```

Csak zöldségek: {'sütőtök', 'répa', 'krumpli'}

Növények halmaz (unió): {'banán', 'eper', 'krumpli', 'sütőtök', 'répa', 'dinnye', 'cseresznye', 'alma'}

Növények2 halmaz (unió): {'banán', 'eper', 'krumpli', 'sütőtök', 'répa', 'dinnye', 'cseresznye', 'alma'}

Déli gyümölcsök: {'narancs', 'banán', 'mangó', 'citrom'}

Gyümölcsök közül ami déli (metszet): {'banán'}

# HALMAZMŰVELETEK

```
43 # Vegyük a gyümölcs és déli gyümölcs halmaz különbségét!
44 kulonbseg = gyumolcsok.difference(deli_gyumolcsok)
45 kulonbseg2 = gyumolcsok - deli_gyumolcsok
46 print("Gyümölcsök közül ami NEM déli (különbség): ", kulonbseg)
47
48 # Nézzük meg, hogy a déli gyümölcs halmaz részhalmaza-e a gyümölcs halmaznak
49 print("Részhalmaz:", deli_gyumolcsok.issubset(gyumolcsok))
50 print("Részhalmaz:", deli_gyumolcsok <= gyumolcsok)
51 # A gyümölcs halmaz tartalmazza-e a déli gyümölcs halmazt.
52 # Ezzel ugyanazt nézzük meg, mint az előbb.
53 print("Tartalmazó halmaz:", gyumolcsok.issuperset(deli_gyumolcsok))
54 print("Tartalmazó halmaz:", gyumolcsok >= deli_gyumolcsok)
55
56 # A halmaz összetétele módosítható
57 # Egy adott elem értéke nem írható át, viszont kivehetőek, berakhatóak elemek.
58 # Adjuk hozzá a halmazhoz a kedvenc gyümölcsünket!
59 gyumolcsok.add("barack")
60 # Vegyük ki a halmazból a nem finom gyümölcsöt!
61 gyumolcsok.remove("dinnye")
62 print("Módosítás után a gyümölcsök: ", gyumolcsok)
```

```
Gyümölcsök közül ami NEM déli (különbség): {'eper', 'dinnye', 'cseresznye', 'alma'}
Részhalmaz: False
Részhalmaz: False
Tartalmazó halmaz: False
Tartalmazó halmaz: False
Módosítás után a gyümölcsök: {'banán', 'cseresznye', 'eper', 'barack', 'alma'}
```



# SZÓTÁRAK, SZÓTÁRMŰVELETEK

- **Kulcs-érték párok:** Minden elemhez egy egyedi kulcs tartozik.
- **Szótárműveletek:** Érték hozzáadása, módosítása, törlése, kulcs alapján történő keresés.
- **Felhasználási területek:**
  - **Adatok asszociatív tárolása:** pl egy személy nevéhez tartozó telefonszám tárolása.
  - **Konfigurációs fájlok:** beállítások tárolása kulcsok és értékek segítségével.
  - **Szöveges adatok feldolgozása:** szójegyzék készítése, szó gyakoriságának számolása.
  - **Webfejlesztés:** session adatok tárolása, HTTP kérések és válaszok feldolgozása.

# SZÓTÁRMŰVELETEK

python\_05\_szotar.py ×

```
1  # Létrehozunk egy szótárat, amely gyümölcsöket és azok árait tartalmazza.
2  gyumolcsok = {
3      "alma": 150,
4      "banán": 200,
5      "cseresznye": 300,
6      "dinnye": 500,
7      "eper": 400
8  }
9
10 # Írjuk ki a szótár elemeinek számát!
11 print("a szótár elemeinek száma: ", len(gyumolcsok))
12
13 # Szótár tagság
14 # Nézzük meg, benne van-e a dinnye a szótárban!
15 print("Van dinnye a szótárban? ", "dinnye" in gyumolcsok) # True
16 # Nézzük meg, benne van-e a barack a szótárban!
17 print("Van barack a szótárban? ", "barack" in gyumolcsok) # False
18
```

```
a szótár elemeinek száma: 5
Van dinnye a szótárban? True
Van barack a szótárban? False
```

# SZÓTÁRMŰVELETEK

```

19 # A keys() függvény: Kiírja a szótár összes kulcsát.
20 print("Gyümölcsök kulcsai (keys):", gymolcsok.keys())
21 # A values() függvény: Kiírja a szótár összes értékét.
22 print("Gyümölcsök árai (values):", gymolcsok.values())
23 # Az items() függvény: Kiírja a szótár összes kulcs-érték párját.
24 print("Gyümölcsök szótár kulcs-érték párai (items):", gymolcsok.items())
25
26 print("Gyümölcsök szótár elemei: ")
27 # Menjünk végig a kulcs-érték párokon az items() használatával:
28 for gymolcs, ar in gymolcsok.items():
29     print(f"A(z) {gymolcs} ára: {ar} Ft")

```

Gyümölcsök kulcsai (keys): dict\_keys(['alma', 'banán', 'cseresznye', 'dinnye', 'eper'])

Gyümölcsök árai (values): dict\_values([150, 200, 300, 500, 400])

Gyümölcsök szótár kulcs-érték párai (items): dict\_items([('alma', 150), ('banán', 200), ('cseresznye', 300), ('dinnye', 500), ('eper', 400)])

Gyümölcsök szótár elemei:

A(z) alma ára: 150 Ft

A(z) banán ára: 200 Ft

A(z) cseresznye ára: 300 Ft

A(z) dinnye ára: 500 Ft

A(z) eper ára: 400 Ft

# SZÓTÁRMŰVELETEK

```

31 # A get() függvény: visszaadja egy adott kulcs értékét, ha létezik.
32 # Ha a kulcs nem található a szótárban, akkor a második paraméterrel megadott
33 # alapértelmezett értéket adja vissza (vagy None-t, ha nincs).
34 print("A banán ára (get):", gyumolcsok.get("banán")) # 200 Ft
35 print("A barack ára (get):", gyumolcsok.get("barack", "Nincs ilyen gyümölcs")) # Nincs ilyen gyümölcs
36
37 # Adjuk hozzá a kedvenc gyümölcsünket a szótárhoz!
38 gyumolcsok["barack"] = 250
39 print(gyumolcsok)
40 # Adjuk hozzá a körtét is a szótárhoz a get függvény segítségével!
41 if gyumolcsok.get("körte") is None:
42     gyumolcsok["körte"] = 250
43 print("Frissített gyümölcsök:", gyumolcsok)

```

```

A banán ára (get): 200
A barack ára (get): Nincs ilyen gyümölcs
{'alma': 150, 'banán': 200, 'cseresznye': 300, 'dinnye': 500, 'eper': 400, 'barack': 250}
Frissített gyümölcsök: {'alma': 150, 'banán': 200, 'cseresznye': 300, 'dinnye': 500, 'eper': 400, 'barack': 250, 'körte': 250}

```



# ÓTÁRMŰVELETEK

```
46 # Szótárból elem törlése
47 #del gyumolcsok["dinnye"]
48 gyumolcsok.pop("dinnye")
49 print("Dinnye törlése után: ", gyumolcsok)
50 # Egy másik szótár létrehozása zöldségekkel és azok árával
51 zoldsegek = {
52     "krumpli": 180,
53     "répa": 100,
54     "sütőtök": 350
55 }
56
57 # Egyesítsük a gyümölcs és zöldség szótárakat egy nagyobb szótárba!
58 # ** szótár argumentum vagy | python3.9 verziótól - gyumolcsok | zoldsegek
59 novenyek = {**gyumolcsok , **zoldsegek}
60 print("Növények (egyesítés):", novenyek)
61 novenyek2 = gyumolcsok | zoldsegek
62 print("Növények2 (egyesítés):", novenyek)
63
64 # Készítünk egy másolatot a szótárról
65 gyumolcsok_masolat = gyumolcsok.copy()
66 print("Gyümölcsök másolat elemei: ", gyumolcsok_masolat)
```

Dinnye törlése után: {'alma': 150, 'banán': 200, 'cseresznye': 300, 'eper': 400, 'barack': 250, 'körte': 250}

Növények (egyesítés): {'alma': 150, 'banán': 200, 'cseresznye': 300, 'eper': 400, 'barack': 250, 'körte': 250, 'krumpli': 180, 'répa': 100, 'sütőtök': 350}

Növények2 (egyesítés): {'alma': 150, 'banán': 200, 'cseresznye': 300, 'eper': 400, 'barack': 250, 'körte': 250, 'krumpli': 180, 'répa': 100, 'sütőtök': 350}

Gyümölcsök másolat elemei: {'alma': 150, 'banán': 200, 'cseresznye': 300, 'eper': 400, 'barack': 250, 'körte': 250}

# SZÓTÁRMŰVELETEK

```

67
68 # Egy adott gyümölcs (eper) árának módosítása (450 Ft-ra)
69 gyumolcsok["eper"] = 450
70 print("Módosított árak (eper):", gyumolcsok)
71
72 # Szótárból sztring: gyümölcsnevek összefűzése vesszővel elválasztva
73 koz = ", "
74 s = koz.join(gyumolcsok.keys())
75 print("Gyümölcsök:", s)

```

Módosított árak (eper): {'alma': 150, 'banán': 200, 'cseresznye': 300, 'eper': 450, 'barack': 250, 'körte': 250}

Gyümölcsök: alma, banán, cseresznye, eper, barack, körte

# PYTHON\_05\_01.PY FELADATLEÍRÁS

## (ÖNÁLLÓ KÓDOLÁS)

- Hozz létre egy halmazt (`harommal_oszthato`), amely tartalmazza az 0-tól 21-ig a 3-al osztható számokat.
- Írd ki az eredményt a képernyőre.
- Add hozzá a halmazhoz újra a 21-et, majd nézd meg mi változik az eredményhalmazba ha két ugyanolyan elem kerül a halmazba?

# PYTHON\_05\_01.PY FOLYTATÁS

## (ÖNÁLLÓ KÓDOLÁS)

- Hozz létre egy halmazt (neggyel\_oszthato), amely tartalmazza az 0-tól 20-ig a 4-el osztható számokat. Írd ki az eredményt a képernyőre.
- Hozz létre egy másik halmazt (paros), amely tartalmazza az 0-tól 20-ig a páros számokat. Írd ki az eredményt a képernyőre.



# PYTHON\_05\_01.PY FOLYTATÁS

(ÖNÁLLÓ KÓDOLÁS)

- Add hozzá a páros számok halmazához a 22-t.
- Majd írd ki újra az eredményhalmazt.
- Ellenőrizd, hogy a négyvel osztható számok halmaza részhalmaza-e a páros számok halmazának.

# PYTHON\_05\_01.PY FOLYTATÁS

(ÖNÁLLÓ KÓDOLÁS)

- Készíts egy új halmazt (hattal osztható), amely tartalmazza azokat a számokat, amelyek mindkét halmazban (paros, hárommal\_osztható) megtalálhatók *(valójában a metszet)*.
- Írd ki a hárommal és négyel osztható számok unióját.

## A feladat:

- Írd ki a képernyőre a gyümölcsök szótár tartalmát.
- Kérj be a felhasználótól egy gyümölcs nevet, ha megtalálható a szótárban akkor írja ki a program a gyümölcs árát.  
*(ha nincs tájékoztassa a felhasználót)*

# PYTHON\_05\_02.PY FOLYJTATÁS

- Adj hozzá egy új elemet (név és ár) az új szótárhoz, amit a felhasználó ír be!
- Majd listázd ki a szótár tartalmát újra.



# PYTHON\_05\_02.PY FOLYTATÁS

- Rendezd a szótárt kulcsok szerint csökkenő sorrendbe!
- Jelenítsd meg.
  
- Egyesítsd a gyumolcs és a több\_gyumolcs szótárakat.
- Majd jelenítsd meg a képernyőn.

# PYTHON\_05\_02.PY FOLYTTATÁS

- Rendezd a szótárat értékek szerint növekvő sorrendbe!
- Majd jelenítsd meg a képernyőn.
- Fűzd össze a kulcsokat egy stringgá a – elválasztóval.
- Írd ki az eredményt a képernyőre.

## A feladat:

- Palindrom ellenőrzés rekurzívan, iteratívan (felezés módszerrel) és brute force elveket követve kerüljön megvalósításra. Minden egyes elv szerint külön függvénybe rendezve.
- Az elkészült függvényeket ellenőrizzük. A felhasználó adjon meg egy vizsgálandó kifejezést, majd vizsgáljuk meg a különböző módszerekkel, írjuk ki az eredményt a képernyőre.
- *Tippek, segítség:*
  - Palindrom (palindróma) az a szókapcsolat, ami visszafelé olvasva ugyanazt a szót adja.
    - pl. Géza kék az ég. Indul a görög aludni.
  - *A felhasználói ellenőrzéskor alakítsuk át a beírt kifejezést (kisbetűs, szóközök nélkül)*

## A feladat:

- Legnagyobb közös osztó kiszámítása rekurzívan
- Kettő szám olyan közös osztója (mely mindkét számot osztja), és az összes közös osztót tekintve a legnagyobb.



## A feladat:

- 1. rész - Írj egy programot, ami verem segítségével ellenőrzi, hogy a felhasználó által beírt szövegben minden zárójelnek van-e párja! Például:
  - Be: `((()((()))))` Ki: helyes
  - Be: `()(((` Ki: helytelen
- 2. rész - Írd meg úgy a programot, hogy a szögletes zárójeleket is figyelje! Például:
  - Be: `(()[(())])` Ki: helyes
  - Be: `()[(])` Ki: helytelen



ÓBUDAI EGYETEM  
ÓBUDA UNIVERSITY



# KÖSZÖNÖM

A MEGTISZTELŐ FIGYELMET!

*Módné Takács Judit*  *modne.t.judit@amk.uni-obuda.hu*