

# List, Tuple, Set és Dict

Oliver Kiss  
Central European University

August 10, 2021

## 1 Container objektumok

### 1.1 List

A list több objektum tárolására használható indexált objektum, melyet szögletes zárójelek között definiálunk.

```
[1]: a = [7, 6.2, 'alma', True]
```

```
[2]: type(a)
```

```
[2]: list
```

```
[3]: a
```

```
[3]: [7, 6.2, 'alma', True]
```

```
[4]: # Pythonban zero indexing van, az első elem indexe mindig 0  
a[0]
```

```
[4]: 7
```

```
[5]: a[1]
```

```
[5]: 6.2
```

```
[6]: a[2]
```

```
[6]: 'alma'
```

```
[7]: a[3]
```

```
[7]: True
```

```
[8]: a[4]
```

```
-----  
IndexError
```

```
Traceback (most recent call last)
```

```
/tmp/ipykernel_9828/3944406842.py in <module>
----> 1 a[4]
```

```
IndexError: list index out of range
```

```
[9]: a[-1]
```

```
[9]: True
```

```
[10]: a[-2]
```

```
[10]: 'alma'
```

```
[11]: a[-3]
```

```
[11]: 6.2
```

```
[12]: a[-4]
```

```
[12]: 7
```

```
[13]: a[-5]
```

```
-----
IndexError                                Traceback (most recent call last)
/tmp/ipykernel_9828/639495242.py in <module>
----> 1 a[-5]
```

```
IndexError: list index out of range
```

```
[14]: # Intervallumok is kiválaszthatók, ekkor a nyitó indexben zárt, a záró indexben
      ↪ nyitott az intervallum
```

```
[15]: a[0:2]
```

```
[15]: [7, 6.2]
```

```
[16]: a[0:3]
```

```
[16]: [7, 6.2, 'alma']
```

```
[17]: a[2:]
```

```
[17]: ['alma', True]
```

```
[18]: a[:3]
```

```
[18]: [7, 6.2, 'alma']
```

```
[19]: a[1:2]
```

```
[19]: [6.2]
```

```
[20]: # Vegyük észre, hogy intervallum kiválasztásánál, még ha az egy elemű is, lista  
      ↪ az output
```

```
[21]: a[2]
```

```
[21]: 'alma'
```

```
[22]: a[2:3]
```

```
[22]: ['alma']
```

```
[23]: # A listának vannak ún. methodjai, amikkel műveleteket végezhetünk rajta
```

```
[24]: # Append: Elem hozzáadása a listához
```

```
[25]: a
```

```
[25]: [7, 6.2, 'alma', True]
```

```
[26]: a.append(False)
```

```
[27]: a
```

```
[27]: [7, 6.2, 'alma', True, False]
```

```
[28]: # A listán azonos elem többször is szerepelhet
```

```
[29]: a.append(5)
```

```
[30]: a.append(5)
```

```
[31]: a.append(5)
```

```
[32]: a
```

```
[32]: [7, 6.2, 'alma', True, False, 5, 5, 5]
```

```
[33]: # Count: megszámolja, hogy egy adott elem hányszor szerepel a listán
```

```
[34]: a.count(5)
```

[34]: 3

```
[35]: # Remove: eltávolítja az adott elem első előfordulását a listáról  
a.remove(True)
```

[36]: a

[36]: [7, 6.2, 'alma', False, 5, 5, 5]

```
[37]: a.remove(5)
```

[38]: a

[38]: [7, 6.2, 'alma', False, 5, 5]

```
[39]: # Pop: eltávolítja az adott indexű elemet a listáról
```

```
[40]: a.pop(2)
```

[40]: 'alma'

[41]: a

[41]: [7, 6.2, False, 5, 5]

```
[42]: # Reverse: megfordítja a lista elemek sorrendjét
```

```
[43]: a.reverse()
```

[44]: a

[44]: [5, 5, False, 6.2, 7]

```
[45]: # Sort: sortolja a listát
```

```
[46]: a.sort()
```

[47]: a

[47]: [False, 5, 5, 6.2, 7]

A lista elemei felülírhatók

```
[48]: a[0] = 'körte'
```

[49]: a

[49]: ['körte', 5, 5, 6.2, 7]

```
[50]: a[0:2] = [2, 3]
```

```
[51]: a
```

```
[51]: [2, 3, 5, 6.2, 7]
```

Még több a list methodokról a Python dokumentációjában:  
<https://docs.python.org/3/tutorial/datastructures.html>

## 1.2 Műveletek list-ekkel

```
[52]: # List hosszának meghatározása
```

```
[53]: len(a)
```

```
[53]: 5
```

```
[54]: # List elemeinek összege - csak ha összeadhatók a listaelemek  
sum(a)
```

```
[54]: 23.2
```

```
[55]: # Listelemek ismétlése  
a*3
```

```
[55]: [2, 3, 5, 6.2, 7, 2, 3, 5, 6.2, 7, 2, 3, 5, 6.2, 7]
```

```
[56]: # Szöveges listáknál konkaténálás lehetséges  
strlist = ['Ez', 'egy', 'lista', 'szavakkal']  
' '.join(strlist)
```

```
[56]: 'Ez egy lista szavakkal'
```

```
[57]: # Szöveg könnyen listába bontható
```

```
[58]: stringem = "Ez egy mondat, amit szavakra fogok majd bontani"
```

```
[59]: stringem.split()
```

```
[59]: ['Ez', 'egy', 'mondat,', 'amit', 'szavakra', 'fogok', 'majd', 'bontani']
```

## 1.3 Tuple

List-hez hasonló objektum, azonban az elemei nem módosíthatók és nem adható hozzá elem. Zárójelek között definiáljuk.

```
[60]: b = (1, 2, 3, 'alma', True)
```

```
[61]: # Ugyanúgy indexálható  
b[0]
```

```
[61]: 1
```

```
[62]: b[3]
```

```
[62]: 'alma'
```

```
[63]: b[-4]
```

```
[63]: 2
```

```
[64]: # Több elem kiválasztásánál egy újabb tuple-t kapunk vissza  
b[2:5]
```

```
[64]: (3, 'alma', True)
```

```
[65]: b[2:3]
```

```
[65]: (3,)
```

```
[66]: # Nem írható felül egy eleme önmagában  
b[2] = 3
```

```
-----  
TypeError                                Traceback (most recent call last)  
/tmp/ipykernel_9828/2963503365.py in <module>  
      1 # Nem írható felül egy eleme önmagában  
----> 2 b[2] = 3  
  
TypeError: 'tuple' object does not support item assignment
```

```
[67]: # A tuple könnyen unpackelhető változókba
```

```
[68]: b
```

```
[68]: (1, 2, 3, 'alma', True)
```

```
[69]: b1, b2, b3, b4, b5 = b
```

```
[70]: b1
```

```
[70]: 1
```

```
[71]: b4
```

```
[71]: 'alma'
```

```
[72]: # A tuple egy eleme bármilyen objektum lehet, lista is
```

```
[73]: c = (3, "alma", [2, 5, 6])
```

```
[74]: c[2]
```

```
[74]: [2, 5, 6]
```

```
[75]: c[2][1]
```

```
[75]: 5
```

```
[76]: # Ez a listára is igaz
```

```
[77]: d = [2, 4, ["alma", True], ('paprika', 'narancs')]
```

```
[78]: d[2]
```

```
[78]: ['alma', True]
```

```
[79]: d[3]
```

```
[79]: ('paprika', 'narancs')
```

```
[80]: d[3][1]
```

```
[80]: 'narancs'
```

```
[81]: d[2].append("alma")
```

```
[82]: d
```

```
[82]: [2, 4, ['alma', True, 'alma'], ('paprika', 'narancs')]
```

```
[83]: # Bár a tuple elemei nem módosíthatók, egyben felülírható az egész tuple
```

```
[84]: b
```

```
[84]: (1, 2, 3, 'alma', True)
```

```
[85]: b = 3
```

```
[86]: b
```

```
[86]: 3
```

```

[87]: b = (2, 4, 6)

[88]: b

[88]: (2, 4, 6)

[89]: b = ('alma', 'narancs')

[90]: b

[90]: ('alma', 'narancs')

[91]: # Ugyanez igaz akkor is, ha a tuple egy list elem

[92]: d

[92]: [2, 4, ['alma', True, 'alma'], ('paprika', 'narancs')]

[93]: d[3] = ('paradicsom', 'hagyma')

[94]: d

[94]: [2, 4, ['alma', True, 'alma'], ('paradicsom', 'hagyma')]

[95]: # De ha a külső objektum egy tuple, annak az elemei nem módosíthatók, kivéve ha
      ↪ az adott elem egy ú.n. mutable object, mint a lista

[96]: c

[96]: (3, 'alma', [2, 5, 6])

[97]: c[2]

[97]: [2, 5, 6]

[98]: c[2][1] = 3

[99]: c

[99]: (3, 'alma', [2, 3, 6])

[100]: # De ezek már nem fognak működni:

[101]: c[2] = 2

```

**TypeError**

Traceback (most recent call last)



```
/tmp/ipykernel_9828/2137047556.py in <module>
----> 1 c[2] = 2
```

**TypeError:** 'tuple' object does not support item assignment

```
[102]: c[1] = 3
```

```
-----
TypeError                                Traceback (most recent call last)
/tmp/ipykernel_9828/2460144158.py in <module>
----> 1 c[1] = 3
```

**TypeError:** 'tuple' object does not support item assignment

## 1.4 Set

```
[103]: # A set egy rendezetlen halmaz, ami egy adott elemet maximum egyszer tartalmaz.
      ↪Kapcsos zárójelek között definiáljuk.
```

```
[104]: e = {2, 3, 4, 'alma', 'körte'}
```

```
[105]: e
```

```
[105]: {2, 3, 4, 'alma', 'körte'}
```

```
[106]: f = {2, 6, 3.4, 'alma', 'narancs'}
```

```
[107]: # Set elemek
      2 in e
```

```
[107]: True
```

```
[108]: 'alma' in e
```

```
[108]: True
```

```
[109]: 'narancs' in e
```

```
[109]: False
```

```
[110]: # Set bővítés
```

```
[111]: e.add('narancs')
```

```
[112]: e
```

```
[112]: {2, 3, 4, 'alma', 'körte', 'narancs'}
```

```
[113]: 'narancs' in e
```

```
[113]: True
```

```
[114]: # De minden elem csak egyszer
```

```
[115]: e.add(2)
```

```
[116]: e
```

```
[116]: {2, 3, 4, 'alma', 'körte', 'narancs'}
```

```
[117]: # Set műveletek
```

```
[118]: # Metszet  
e & f
```

```
[118]: {2, 'alma', 'narancs'}
```

```
[119]: # Unio  
e | f
```

```
[119]: {2, 3, 3.4, 4, 6, 'alma', 'körte', 'narancs'}
```

```
[120]: # Különbség  
e-f
```

```
[120]: {3, 4, 'körte'}
```

```
[121]: f-e
```

```
[121]: {3.4, 6}
```

```
[122]: # Set nem indexelhető, az elemeknek nincs sorrendje  
e[2]
```

```
-----  
TypeError                                Traceback (most recent call last)  
/tmp/ipykernel_9828/675013487.py in <module>  
      1 # Set nem indexelhető, az elemeknek nincs sorrendje  
----> 2 e[2]  
  
TypeError: 'set' object is not subscriptable
```

```
[123]: # Set listté konvertálható, az elemek sorrendje véletlenszerű  
list(e)
```

```
[123]: ['körte', 2, 3, 4, 'alma', 'narancs']
```

```
[124]: # List setté alakítható, duplikált elemekből egy marad
```

```
[125]: a
```

```
[125]: [2, 3, 5, 6.2, 7]
```

```
[126]: a.append(2)
```

```
[127]: a
```

```
[127]: [2, 3, 5, 6.2, 7, 2]
```

```
[128]: set(a)
```

```
[128]: {2, 3, 5, 6.2, 7}
```

```
[129]: # Duplikált elemek egyszerűen eltávolíthatók  
list(set(a))
```

```
[129]: [2, 3, 5, 6.2, 7]
```

## 1.5 Dictionary

```
[130]: # Key - value párok definiálására szolgál, integer indexek helyett a key az  
→ index amivel a value-ra hivatkozunk. Kapcsos zárójelek között key:value  
→ felsorolásokkal definiáljuk
```

```
[131]: mydict = {'alma': 'piros', 'dinnye': 'sárga', 'uborka': 'zöld'}
```

```
[132]: mydict['alma']
```

```
[132]: 'piros'
```

```
[133]: # Új elem hozzáadása  
mydict['szilva'] = 'lila'
```

```
[134]: mydict
```

```
[134]: {'alma': 'piros', 'dinnye': 'sárga', 'uborka': 'zöld', 'szilva': 'lila'}
```

```
[135]: # Key és value bármilyen objektum lehet  
mydict[3] = True
```

```
[136]: mydict
```

```
[136]: {'alma': 'piros',  
       'dinnye': 'sárga',  
       'uborka': 'zöld',  
       'szilva': 'lila',  
       3: True}
```

```
[137]: mydict['randomlista'] = [2, 3, 4, 5]
```

```
[138]: mydict
```

```
[138]: {'alma': 'piros',  
       'dinnye': 'sárga',  
       'uborka': 'zöld',  
       'szilva': 'lila',  
       3: True,  
       'randomlista': [2, 3, 4, 5]}
```

```
[139]: piac = {'alma':{'ár':500,'osztály':1},'narancs':{'ár':800,'osztály':2}}
```

```
[140]: piac['alma']['ár']
```

```
[140]: 500
```

```
[141]: # Dictionary-k update-elhetők egymással
```

```
[142]: ujpiac = {'alma':{'ár':400,'osztály':2},'paprika':{'ár':400,'osztály':2}}
```

```
[143]: piac.update(ujpiac)
```

```
[144]: piac
```

```
[144]: {'alma': {'ár': 400, 'osztály': 2},  
       'narancs': {'ár': 800, 'osztály': 2},  
       'paprika': {'ár': 400, 'osztály': 2}}
```