

15. LISTÁK

Vannak olyan esetek, amikor sok (pl.: 50 db) változóra van szükségünk. Ebben az esetben sok időbe telne a az értékadás és nehézkes lenne a nevek megjegyzése. Ezért megoldást jelent, a sok változónak egyetlen nevet adunk, de mellette sorszámozzuk őket. Például (hónapok(4), napok(6), stb).



A lista névvel és sorszámokkal ellátott összetett változó. A lista elemeire nevével és a sorszámukkal hivatkozunk.

A listaelem sorszámát **indexnek** nevezzük.

Arra kell figyelni, hogy a **sorszámozás nulla értékkel indul**. (Pl.: január -> 0)

Tehát: hónap(2):= „március”

A 12 elemű lista utolsó tömbindexe a 11.

A lista egy sokoldalú adatszerkezet, műveleteket hajthatunk végre rajtuk;

törölhetünk benne, rendezhetjük az értékeket, stb.

Fontos, hogy a kör alakú zárójelek helyett négyzetes zárójeleket kell használni!

pl.: hónapok=[„Január”, „Február”, ..., „December”]

0	Január
1	Február
2	Március
3	Április
4	Május
5	Június
6	Július
7	Augusztus
8	Szeptember
9	Október
10	November
11	December

(15a.py)

Az első listás feladat nagyon egyszerű. Hozunk létre egy „honap” nevű listát, melyet feltöltünk a hónapok neveivel! Majd kiíratjuk a lista hatos számú, pontosabban a hetedik elemét a júliust!

```

15a.py x
1 hónapok=["Január", "Február", "Március", "Április", "Május", "Június",
2         "Július", "Augusztus", "Szeptember", "Október", "November", "December"]
3 print(hónapok[6])
    
```

Run: 15a x
C:\Users\kolma
Július

(15b.py)

Mentsd el másként az előző programot, és változtasd meg úgy, hogy írja ki a hónapok neveit külön sorokba a mint szerint!

Ehhez használjuk a „for” utasítást!

```

15b.py x
1 hónapok=["Január", "Február", "Március", "Április", "Május", "Június",
2         "Július", "Augusztus", "Szeptember", "Október", "November", "December"]
3 for i in range(12):
4     print(hónapok[i])
    
```

Run: 15b x
C:\Users\ko
Január
Február
Március
Április
Május
Június
Július
Augusztus
Szeptember
Október
November
December

Tehát többféle módon lehetséges egy **új lista létrehozása**; legegyszerűbb az elemek szögletes zárójelbe való felsorolása ([és]):

szamok = [10, 20, 30, 40]

gyumolcsok = ["alma", "eper", "barack", "körte"]

Az első példa egy lista, amely négy egész számot tartalmaz. A második lista pedig három sztringet tartalmaz. A lista elemeinek nem kell azonos típusúnak lennie. A következő lista tartalmaz egy sztringet, egy valós számot, egy egész számot és (érdekességképpen) egy másik listát.

vegyes_lista = ["hello", 2.0, 5, [10, 20]]

A listában szereplő másik listáról azt mondjuk, hogy beágyazott lista.

Végül azt a listát, amely nem tartalmaz elemeket, üres listának nevezzük, és [] jelöljük.



(15c.py)

A lista hosszának használata a már tanult len() függvénnyel lehetséges. Tehát ha nem tudjuk konkrétan hogy a lista hossza mekkora, akkor a len() függvény megmondja. Nézzünk példát erre!
 Soroljuk fel a sakkfigurák neveit egy listába, majd for ciklussal „i”-től menjünk a listán végig! Használjuk a len() függvényt a mint szerint!



```

1 sakk_figurak=["Királynő", "Király", "Bástya", "Futó", "Ló", "Gyalog"]
2
3 for i in range(len(sakk_figurak)):
4     print(sakk_figurak[i])
    
```

```

Run: 15c x
E:\00_MM\
Királynő
Király
Bástya
Futó
Ló
Gyalog
    
```

(15d.py)

Ebben a példában létrehozunk egy olyan listát, amelyben beágyazások is találhatóak. De amikor kiíratjuk a lista hosszát, akkor csak a "fő" lista darabszámát írja ki!

```

1 lista_1=["sakkfigurák", 8, ["Gyalog", "Futó", "Király"], [1, 2, 3], 2.5]
2 hossz = len(lista_1)
3 print(hossz)
    
```

```

Run: 15d x
E:\00_MM\
5
    
```

(15e.py)

Listaműveletek:

listák összefűzése „+” operátorral

listák többszörözése „*” operátorral

```

1 a = [1, 2, 3]
2 b = [4, 5, 6]
3 c = a + b
4 print(c)
5 print("-----")
6 d = [0] * 4
7 print(d)
8 e = [1, 2, 3] * 3
9 print(e)
10 print("-----")
    
```

```

Run: 15e x
C:\Users\kolmank\AppData\Local\Programs\Python\Python38\Python.exe
[1, 2, 3, 4, 5, 6]
-----
[0, 0, 0, 0]
[1, 2, 3, 1, 2, 3, 1, 2, 3]
-----
    
```

(15f.py)

Listák szeletelése

Listák módosítása

Új elem beszúrása adott helyre

Listában lévő tagok törlése

```

1 sakk_figurak=["Királynő", "Király", "Bástya", "Futó", "Ló", "Gyalog"]
2
3 print(sakk_figurak[1:3])
4 print(sakk_figurak[:4])
5 print(sakk_figurak[3:])
6 print(sakk_figurak[:])
7
8 print("-----")
9 sakk_figurak=["Király", "Királynő", "Bástya", "Futó", "Ló", "Gyalog"]
10
11 sakk_figurak[1]="Vezér"
12 sakk_figurak[4]="Huszár"
13 print(sakk_figurak)
14
15 print("-----")
16 sakk_figurak=["Királynő", "Futó", "Ló", "Gyalog"]
17 sakk_figurak[1:1]="Király", "Bástya"
18 print(sakk_figurak)
19
20 print("-----")
21 del(sakk_figurak[2:4])
22 print(sakk_figurak)
    
```

```

Run: 15f x
C:\Users\kolmank\AppData\Local\Programs\Python\Python38\Python.exe
['Király', 'Bástya']
['Királynő', 'Király', 'Bástya', 'Futó']
['Futó', 'Ló', 'Gyalog']
['Királynő', 'Király', 'Bástya', 'Futó', 'Ló', 'Gyalog']
-----
['Király', 'Vezér', 'Bástya', 'Futó', 'Huszár', 'Gyalog']
-----
['Királynő', 'Király', 'Bástya', 'Futó', 'Ló', 'Gyalog']
-----
['Királynő', 'Király', 'Ló', 'Gyalog']
    
```

Ebben a feladatban a mintaprogram alapján értelmezzük a kódokat, hogy mi miért történik!