

Tartalomjegyzék

- 1 Ismétlés az előző gyakról
- 2 Ismétlés az előadásról
- 3 Feladatok
 - ◆ 3.1 Lista ismétlés
 - ◆ 3.2 Elágazások
 - ◆ 3.3 Vezérlési szerkezetek (if, for, while)
 - ◆ 3.4 Listák kezelése
 - ◆ 3.5 Házi előkészítés

Ismétlés az előző gyakról

- String: 'kutya'
- Lista: [1, 8, 'ketto', 1, 9]
- Lista indexek **0**-tól kezdődnek, így az utolsó eleme egy **n** elemű listának **n - 1**
- Listák **3.** elemének lekérdezése, feltéve, hogy **l** a lista: l[2]
- Listák hosszának lekérdezése: len(l)
- Új elem hozzáadása: l.append('nyolc')
- stb.

Ismétlés az előadásról

3. előadás

- Szótár: {'one':1,'two':2}
- Szótár elemének lekérdezése (**d** a szótár): d['one']
- Elágazás:

```
if feltétel1:
    művelet1
elif feltétel2:
    művelet2
else:
    művelet3
```

- Minden aminek igaz-hamis értéke van lehet feltétel. Pl: $x > 5$, vagy akár $x * 6 + 16 == 15 * y + 1$
- For ciklus:

```
l = [4, 5, 2]
for e in l:
    print e * 2
```

- Ennek az eredménye:

```
8
10
4
```

- While ciklus:

```
i = 1
x = 254
while i <= x:
```

```
if x % i == 0:  
    print i  
i = i + 1
```

- Ezt találjátok ki mit csinál!
- Függvény definiálás:

```
def titok(x):  
    i = 1  
    while i <= x:  
        if x % i == 0:  
            print i  
        i = i + 1
```

- Függvény meghívása:

```
titok(4752)
```

- Függvényeknek akárhány paramétere (bemenete) lehet, akár 0 is.
- Nagy különbség van a kiírás és érték visszaadás közt:

```
def fv1(x):  
    print x * x + 2 * x - 1
```

```
def fv2(x):  
    return x * x + 2 * x - 1
```

```
ertek1 = fv1(5)  
ertek2 = fv2(5)
```

- Az **ertek1** értéke **None** mivel a **fv1** nem ad vissza semmilyen értéket, csak kiírja.
- Az **ertek2** értéke 34, mert a **fv2** visszaadta azt az értéket (**return**) és nem kiírta
- Függvény csak egy értéket tud visszaadni, de akárhányat ki tud írni:

```
def pelda():  
    print 22  
    return 5  
    print 15
```

```
x = pelda()
```

- A 15 nem lesz kiírva, mert a **return 5** paranccsal visszatér a függvény, azaz ezt az értéket visszaadja és értékül adódik az 5 x-nek

Feladatok

Lista ismételés

- Készítsétek el a 2, 4, 'tizenöt', 4 elemeket tartalmazó listát
- Kérdezzétek le a méretét
- Kérdezzétek le a 3. elemét
- Fűzzétek a lista végére a 'negy' stringet
- Kérdezzétek most le a 2. elemtől utolsó elemig tartó részlistát, mentsétek ezt egy másik változóba
- A részlistának is kérjétek le a méretét

Elágazások

Vezérlési szerkezetek (if, for, while)

- Egészítsd ki a függvényt, hogy ha az első paraméter kisebb mint a második, akkor azt a karakterláncot adja vissza, hogy "http://wiki.math.bme.hukisebb"http://wiki.math.bme.hu, ha nagyobb, akkor azt, hogy "http://wiki.math.bme.hunagyobb"http://wiki.math.bme.hu és ha egyenlő akkor "http://wiki.math.bme.huegyenlo"http://wiki.math.bme.hu-t!

```
<!-- hasonlit(a, b):
if a < b:
    <!-- "http://wiki.math.bme.hukisebb"http://wiki.math.bme.hu
elif a > b:
    <!-- "http://wiki.math.bme.hunagyobb"http://wiki.math.bme.hu
-->:
    <!-- "http://wiki.math.bme.huegyenlo"http://wiki.math.bme.hu
```

- Definiálj egy Sage függvényt *elovel* néven, amelynek egy bemenete van (a), és a "http://wiki.math.bme.hupozitiv"http://wiki.math.bme.hu karakterláncot (vagyis stringet) írja ki ha a kapott paraméter pozitív, "http://wiki.math.bme.hunegativ"http://wiki.math.bme.hu-at ad vissza ha a szám negatív, és "http://wiki.math.bme.hunulla"http://wiki.math.bme.hu-ta ad ha nulla volt a paraméter értéke.

Listák kezelése

- Egészítsd ki a kódot, hogy a függvény a kapott lista legkisebb elemét adja vissza:

```
def minimum(l):
    minelem = <!--
    <!-- e in l:
        if e < <!--:
            minelem = e
    <!-- minelem
```

- Írj függvényt ami a kapott lista elemeinek a szummáját visszaadja. (Ne használjátok a **sum** beépített függvényt.)
- Írj függvényt *sokszorozzo* néven, amely bemenetként kap egy számot (n) és még egy paramétert, a -t (ennek a típusa bármi lehet). A függvény adjon vissza egy listát, amiben n -szer szerepel az a értéke. Például: *sokszorozzo(3, "http://wiki.math.bme.hubla"http://wiki.math.bme.hu)* kimenete ["http://wiki.math.bme.hubla"http://wiki.math.bme.hu, "http://wiki.math.bme.hubla"http://wiki.math.bme.hu, "http://wiki.math.bme.hubla"http://wiki.math.bme.hu] legyen.
- Írj Sage függvényt amely megfordít egy bemenetként kapott listát!

Házi előkészítés

Mentsünk el egy notebookot, hogy háziban ne legyen ezzel gond.