

# Tartalomjegyzék

- 1 Elmélet
  - ◆ 1.1  
Adattípusok
  - ◆ 1.2  
Szöveg
  - ◆ 1.3 Input,  
Output
  - ◆ 1.4 File  
I/O
- 2 Feladatok
  - ◆ 2.1  
Eloszlásfüggvény
  - ◆ 2.2  
Szavak  
gyakorisága
  - ◆ 2.3  
Vásárlók

## Elmélet

### Adattípusok

- None
- numerikus
- egész
- lebegőpontos
- (complex)
- long
- boolean
- string
- tuple

```
x = (1, 2, 3, 4)
```

- lista

```
x = [1, 2, 3, 4]
x.append(5)
len(x)
x[1] = x[2]
```

- szótár

```
A = {}
phonebook = {'Andrew Parson':8806336, 'Emily Everett':6784346, 'Peter Power':7658344, 'Lewis Lane':7658344}
for key in phonebook:
    ...
if key in phonebook:
    print phonebook[key]
...
```

- (halmaz)

```
>>> A=set([1,2,3])
>>> B=set([2,3,4])
>>> A | B
set([1, 2, 3, 4])
>>> A & B
set([2, 3])
>>> A - B
set([1])
>>> A ^ B
set([1, 4])
```

## Szöveg

- Erről majd lesz részletesebben szó előadáson.
- Itt néhány olyan metódus szerepel, amit a gyakorlaton használunk

```
>>> A='abcdefgh'
>>> A.replace('a','x')
'xbcd efgh'
>>> A='abc dce'
>>> L = A.split(" ")
>>> L
['abc', 'dce']
>>> A = 'AbCdEf'
>>> A.lower()
'abcdef'
```

- [\[1\]](#)

## Input, Output

- **print**
- **str()** - string-et csinál más típusú objektumból, **print változó** és **print str(változó)** ekvivalens
- **input()** - Kiírja a neki adott string-et, és vár egy bemenetet, a visszatérési értéke a bemenet automatikusan értelmezve, pl:

```
m = input("Magasság centimeterben:")
print "Magasság meterben:", m / 100.0
```

- **raw\_input()** - Kiírja a neki adott string-et, és vár egy bemenetet, a visszatérési értéke a bemenet stringként, pl:

```
n = raw_input("Neved:")
print "Hello", n
```

## File I/O

- **open(file\_neve, megnyitási\_mód)** - megnyit egy file-t, első paramétere a file neve, második a megnyitási mód: 'w', 'a', 'r' a szokásosak. Pl:

```
f = open('test.txt', 'w')
```

- **write()** - metódusa a file objektumoknak (az előző példában **f** file objektum), a neki adott string-et a file-ba írja. Pl:

```
f.write("Which witch watches which witch's watches?\nKövetkező sor\n")
```

- **read()** - metódusa a file objektumoknak, beolvassa az egész file-t egy (potenciálisan) jó nagy string-be. Pl:

```
f = open('test.txt', 'r')
s = f.read()
print s
```

- **readline()** - metódusa a file objektumoknak, beolvas egy sort a file-ból, az újsor jelet is beolvassa, az üres sorokat '\n'-ként olvassa be. Ha a file végére ért üres sort olvas be. Pl:

```
line = f.readline()
print line
```

- **readlines()** - metódusa a file objektumoknak, beolvassa az összes sort mint listát. Pl:

```
lines = f.readlines()
for line in lines:
    print line
```

- A file objektumok iterálhatóak, azaz az előző példa ekvivalens ezzel:

```
for line in f:
    print line
```

- **close()** - metódusa a file objektumoknak, bezárja a file-t. Pl:

```
f.close()
```

## Feladatok

### Eloszlásfüggvény

- adat.txt
- Írj függvényt, mely megszámolja egy paraméterként kapott listában az egyes elemek gyakoriságát. A végeredményt egy szótárban menti el, melyet visszatérési értéként ad vissza.
- Például

```
[1,1,2,1]
```

listára a visszatérési érték:

```
{1:3,2:1}
```

- Írj programot, mely beolvas egy egyoszlopos file-t, és kiszámítja az oszlopban szereplő egész számok eloszlásfüggvényét.
- A program az eredményt írja ki egy két oszlopos file-ba
- Egészítsd ki a programot úgy, hogy kiszámolja a mintában található értékek
  - ♦ Átlagát
  - ♦ Maximumát
  - ♦ Minimumát
- Ezeket az értékeket a program egyszer? "<http://wiki.math.bme.hu/print>"<http://wiki.math.bme.hu> paranccsal írja ki a felhasználónak.
- Jelenítsd meg a kapott eloszlásokat gnuplot programban (részletek órán).
- Változtasd meg a programot úgy, hogy tetsz?leges valós számokon számoljon gyakoriságot (részletek órán).

## Szavak gyakorisága

- Számold meg egy szöveges fájlban az egyes szavak gyakoriságát.
- Ügyelj arra, hogy egy beolvasott sor végénél töröld le a sorvége karaktert:

```
line = line.rstrip("\n")
```

- A sorokban space mentén tördeld a szavakat.
- Ne különböztess meg a kis és nagybetűket a szavakban.
- Egészítsd ki a programot úgy, hogy kiírja a top K szót a szövegben (ahol K paraméter).

## Vásárlók

- Írj programot, mely vásárlások között keresi meg a két leginkább azonosat.
- A program bemenete egy olyan file, melyben soronként vannak felsorolva különböző vásárlások.  
Például:

```
kenyér paradicsom joghurt  
tej tejföl hagyma  
kenyér hagyma  
...
```

- A program olvassa be a file minden egyes sorát.
- A beolvasás után páronként vizsgálja meg a vásárlásokat.
- A hasonlóságot Jaccard-hasonlóság alapján számoljátok (használatok halmazokat!)