

Tartalomjegyzék

- 1 Ismétlés
- 2 Analitikus számítások sage-ben
- 3 Python
 - ◆ 3.1 Bevezetés
 - ◆ 3.2 Célok
 - ◆ 3.3 Néhány hasznos tudnivaló
 - ◇ 3.3.1 Interaktív mód
 - ◇ 3.3.2 Scriptek
 - ◇ 3.3.3 Függvények
 - ◇ 3.3.4 Argumentumok
 - ◇ 3.3.5 File I/O
 - ◆ 3.4 Feladatok
 - ◆ 3.5 Feladatok Megoldásai
 - ◇ 3.5.1 1. Feladat
 - ◇ 3.5.2 2. Feladat

Ismétlés

- el?z? gyakorlat sage worksheet
- BFS
- DFS
- Dijkstra algoritmus
- Bináris keresés rendezett listában (el?adás)
- Kódolni jó! coding
- https://docs.google.com/document/d/1sn7MQl8hyO0ixGrzFWusmmqFbv1x_rwyDEP2PFUHpXA/edit

Analitikus számítások sage-ben

- http://www.sagemath.org/doc/tutorial/tour_algebra.html
- Egyenletek

```
sage: x = var('x')
sage: solve(x^2 + 3*x + 2, x)

sage: x, b, c = var('x b c')
sage: solve([x^2 + b*x + c == 0], x)
```

- Deriválás

```
sage: u = var('u')
sage: diff(sin(u), u)
```

- n-edik derivált:

```
sage: diff(sin(x^2), x, 4)
```

- Határozatlan integrálás

```
sage: integral(x*sin(x^2), x)
```

- Határozott integrálás

```
sage: integral(x/(x^2+1), x, 0, 1)
```

- Egyenletrendszerek, differenciálegyenletek ...

Python

Bevezetés

- magas szintű programozási nyelv
- interpreteres nyelv
- objektumorientált
- free and open source
- terminál -> Python -> interaktív mód
- Python scriptek

Célok

- Python scriptek írása
- file I/O
- modulok és importálásuk
- függvények
- argumentumok

Néhány hasznos tudnivaló

Interaktív mód

```
robee64@M14Z:~$ python
```

- Kilépés python-ból:

```
>>> exit()
```

Scriptek

- Python script meghívása

```
robee64@M14Z:~$ python proba.py
```

- Modulok importálása pythonban

```
>>> import random
>>> import math
>>> import sys
```

- Egy példa scriptre (és az input() függvényre):

```
#!/usr/bin/python
#coding=UTF-8
s = input("Mondj egy számot:")
print "Ennél eggyel kisebbet mondtál: ", str(s+1)
```

Függvények

- Függvény definiálása

```
def proba():
    ...
```

- Példa függvényekre:

```
def main():
    userNumber=getNumber()
    print "Ennél eggyel kisebbet mondtál: ", str(userNumber+1)

def getNumber():
    num = input("Mondj egy számot:")
    return num

main()
```

- Ciklusok, stb. hasonlóak, mint sage-ben.

Argumentumok

- indexelés!

```
#!/usr/bin/python
#coding: utf-8

import sys

for arg in sys.argv:
    print arg

#!/usr/bin/python
#coding: utf-8
x=float(sys.argv[1])
```

File I/O

- MIT nyitok meg
- "http://wiki.math.bme.huHOGYAN"http://wiki.math.bme.hu: 'r','w','a' (read, write, append)

```
f = open('input.txt', 'r')
#fileString=f.read()
#lineString = f.readline()
for line in f:
    L=[]
    L=line.split(" ")
    for i in L:
        print i
f_out = open('out.txt', 'w')
f_out.write('This is how we write out number into a file:' + str(6) + '\n')
```

Scriptek

Feladatok

- Írj programot, mely egy argumentumként kapott számig kiírja 1-től az egész számokat a szintén argumentumként megadott file-ba.
- Írj programot, ami egy két oszlopos file-t (az oszlopokat "http://wiki.math.bme.hu" és "http://wiki.math.bme.hu választja el) minden sorában összeadja az elemeket. A file bemeneti és kimeneti file-ok nevét a program argumentumként kapja meg. A kimenet 3 oszlopból álljon, mely tartalmazza az eredeti oszlopokat, és - harmadikként - az összeget. Törekedj beszélgető változónevek megadására.
- Struktúráld át úgy az előző programodat, az összeadásra írd egy külön függvényt.

Feladatok Megoldásai

1. Feladat

```
import sys

def main():
    n=int(sys.argv[1])
    f_out=open(sys.argv[2], 'w')
    for i in range(n):
        f_out.write(str(i+1) + "\n")

main()
```

2. Feladat

```
import sys

def main():
    f_in=open(sys.argv[1], 'r')
    f_out=open(sys.argv[2], 'w')
    for line in f_in:
        L=line.split(" ")
        x1=float(L[0])
        x2=float(L[1])
        sum=x1+x2
        f_out.write(str(x1) + " " + str(x2) + " " + str(sum) + "\n")

main()
```