

Programmeerimise süvendatud algkursus ITI0140

2014

Failidest andmete lugemine

- Avamiseks on **open** funktsioon
- Ridu saab failist lugeda **for** tsükli abil
- Kõik failid tuleb sulgeda **close** funktsiooniga

```
file = open("scarlet.txt")  
  
for line in file:  
    print(line)  
  
file.close()
```

```
l»æThe Project Gutenberg EBook of A Study In Scarlet,  
by Arthur Conan Doyle
```

```
...
```

Funktsioonid?

- Miks peaks kasutama funktsioone?
- Miks on järgnev kood halb?

```
# operate with file 1
file = open("scarlet.txt")
for line in file:
    # a lot of stuff
    pass
file.close()
```

```
# operate with file 2
file = open("hound.txt")
for line in file:
    # a lot of same stuff
    pass
file.close()
```

Funktsioonid!

- Aitavad vähendada koodi kordumist
- Koodi muutub mõistetavateks klotsideks
- Järgnev kood on juba parem

```
def process_file_contents(file):  
    for line in file:  
        # a lot of stuff  
        pass  
  
# operate with file 1  
file = open("scarlet.txt")  
process_file_contents(file)  
file.close()  
  
# operate with file 2  
file = open("hound.txt")  
process_file_contents(file)  
file.close()
```

Funktsioonid!!

- Ja järgnev kood on veel parem

```
def process_file_contents(file):  
    for line in file:  
        # a lot of stuff  
        pass  
  
def process_file(filename):  
    file = open(filename)  
    process_file_contents(file)  
    file.close()  
  
# operate with file 1  
process_file("scarlet.txt")  
  
# operate with file 2  
process_file("hound.txt")
```

Millal kirjutada funktsioone?

- Kui olete konveiertöölise rollis, st kirjutate midagi, mida te juba olete kirjutanud
- Kui tundub, et mingi koodijupp võiks veel kuskil kasuks tulla
- Kui teie funktsioon muutub pikkuse tõttu arusaamatuks
 - Üks mõistlik funktsioon jääb 10-20 rea piiresse
 - Üks mõistlik funktsioon täidab ühte ja selget ülesannet

Mõned funktsioonide näiteid

- loeb failist ridu
- tükeldab ühe rea sõnadeks
- leiab unikaalsed sõnad

Ülesanne: unikaalsed sõnad

Tekstitötluse ülesande alus on kaks A. C. Doyle'i raamatut, mis on vabalt kättesaadavad Project Gutenberg kaudu:

- 1) *A Study in Scarlet*: <http://www.gutenberg.org/cache/epub/244/pg244.txt>
- 2) *The Hound of the Baskervilles*: <http://www.gutenberg.org/cache/epub/3070/pg3070.txt>

Tekstid on vaja jaotada sõnadeks. Sõnaks loeme antud juhul kõik järjest asetsevad tähed, mille kohta Pythoni **isalpha()** funktsioon vastab tõeselt. Sõnade ümber asetsevad muud sümbolid. Sõnadeks jaotamine ei peaks olema tõstutundlik.

Leida:

- kui palju on sõnu kokku tekstis 1
- kui palju on sõnu kokku tekstis 2
- kui palju on unikaalseid sõnu tekstis 1
- kui palju on unikaalseid sõnu tekstis 2
- kui palju on unikaalseid sõnu tekstis 1 ja tekstis 2 kokku
- kui palju on unikaalseid sõnu tekstis 1, mida tekstis 2 pole
- kui palju on unikaalseid sõnu tekstis 2, mida tekstis 1 pole
- kui palju on unikaalseid sõnu, mis esinevad mõlemas tekstis

Märkused:

- Kasutage funktsioone!
- Lisage funktsioonidele kommentaarid