

Pyhton. Andmed failides

Mahukamaid andmeid on sageli otstarbekas hoida failides. Alandmed loetakse failist muutuja(te)sse (ühesuguse struktuuriga andmeid on mugav hoida näiteks loendites), programmi töö tulemused salvestatakse faili.

Fail avatakse funktsiooniga `open`. Tulemuseks on failiobjekt, mille kaudu saab infot faili kohta ja andmeid lugeda/kirjutada.

```
f = open(fnimi, töötlusviis)
```

`f` – objektimuutuja, `fnimi` – faili täisnimi: `[tee]nimi.laiend`

`'r'` – lugemine (vaikimisi), `'w'` – kirjutamine, `'a'` – lisamine, `'t'` – tekstirežiim (vaikimisi), `'b'` – kahendkood jm

Pärast kasutamist tuleks fail sulgeda meetodiga `close()`

```
f.close()
```

Failiobjekti omadusi ja meetodeid:

<code>f.name</code>	– faili nimi
<code>f.close()</code>	– faili sulgemine
<code>f.closed</code>	– kas fail on suletud
<code>f.readable()</code> , <code>f.writable()</code>	– kas fail on lugemiseks / kirjutamiseks
<code>f.read()</code>	– kõigi andmete lugemine, tulemuseks tekst
<code>f.readline()</code>	– ühe kirje (rea) lugemine, tulemuseks tekst
<code>f.readlines()</code>	– kogu teksti lugemine kirjete kaupa loendisse
<code>f.seek()</code>	– lugemisviit nihutatakse etteantud kohta
<code>f.seekable()</code>	– kas lugemisviita on võimalik nihutada
<code>f.tell()</code>	– lugemisviit – mitu sümbolit/baiti on loetud
<code>f.truncate()</code>	– faili tühjendamine
<code>f.write()</code>	– kirje (stringi) kirjutamine tekstifaili
<code>f.writelines()</code>	– loendi kirjutamine faili – ainult tekstid, elemendi lõpus soovitatavalt <code>'\n'</code>

`print(väärtus(ed), sep=' ', end='\n', file=f_obj)` – tekstifaili kirjutamine (mugavam kui meetod `write`)

Failis võib olla esitatud tabel: andmestik mis moodustub ridadest (kirjetest). Kõigil kirjetel peaks olema ühesugune struktuur. Kirje koosneb ühest või mitmest **väljast** (väärtusest), mis eraldatakse üksteisest mingi eraldajaga (tühik, tabulaator `\t` vm). Väljade väärtuste sees seda eraldajat ei ole. Reavahetuse sümbol kirje lõpus on `\n`.

NB! Tekstifail(id) ja programmifail võiks (vähemalt esialgu) paikneda samas kaustas, siis ei pea tekstifaili avamisel näitama selle asukohta.

Ülesanne:

1. Lugeda failidest [eesti.txt](#) ja [inglise.txt](#) väärtused loenditesse (eelnevalt salvestada failid kettale). Kirjutada väärtused uude tekstifaili nii, et sama tähendusega sõnad on ühes reas (eraldajaks tühik).
2. Lugeda väärtused uuest failist. Moodustada:
 - kaks loendit (eesti ja inglisekeelsed sõnad eraldi)
 - üks loend, mille elemendid on sõnapaarid (loendid)

```
import os, os.path
```

```
os.getcwd() – töökataloog
```

```
os.path.exists(failinimi) – kas eksisteerib selline fail
```

`tekst.split()` – stringi meetod, mis jagab teksti osadeks, moodustades loendi