

Viikkotehtävät 10 – Tiedostonkäsittely

1. Lataa Moodlesta tiedosto nimeltä **"artists.txt"**. Lue tiedoston sisältö Pythonilla muuttujaan, sekä **tulosta tiedoston sisältö silmukassa jokainen rivi erikseen**. Jokaisen rivin alussa tulee olla myös nuoli, joka koostuu merkeistä "->".

Muista kopioida tiedosto samaan kansioon missä Python-tiedostosi ovat!
(Voit käyttää PyCharmia tiedoston kopioimiseen)

Esimerkki ohjelman toiminnasta:

```
Kaikkien aikojen myydyimpiä kotimaisia äänitteitä:  
  
-> Eppu Normaali, Repullinen hittejä, 1996  
-> Kirka, Surun pyyhit silmistäni, 1988  
-> Dingo, Kerjäläisten valtakunta, 1985  
-> Joel Hallikainen, Joel Hallikainen, 1992  
-> Eri esittäjiä, Vain elämää, 2012  
-> Kaija Koo, Tuulten viemää, 1993  
-> Hurriganes, Roadrunner, 1974  
-> Vesa-Matti Loiri, Sydämeen joulun teen, 1988  
-> Topi Sorsakoski & Agents, Besame Mucho, 1987  
-> Anssi Kela, Nummela, 2001
```

Tehtävän tiedostonimi = **exercise10_1.py**

Tyypillinen koodimäärä: **5-12 riviä** (sisältää myös funktioiden koodit, tyhjiä rivejä ja kommentteja ei lasketa)

Huom! artists.txt –tiedostoa ei tarvitse ladata codePostiin, se löytyy sieltä jo valmiiksi!

Pieni lisätehtävä: Tee tiedostonkäsittelylle oma funktio, joka palauttaa tiedoston sisällön listana! Tulosta lopputulos ohjelmakoodin puolella yllä olevan esimerkkikuvan mukaisesti.



2. Tee vieraskirjaohjelma. Kysy ensin käyttäjältä, haluaako hän lukea vai kirjoittaa vieraskirjaan (l/k). Jos hän haluaa lukea vieraskirjaa, hae tiedostosta rivit, ja tulosta ne näytölle käyttäen silmukkaa. Jos käyttäjä haluaa kirjoittaa vieraskirjaan, tallenna rivi tiedoston loppuun (ohjausmerkki **a**)

Voit käyttää tiedostonnimenä **guestbook.txt**. Luo aluksi tyhjä tiedosto tällä nimellä samaan kansioon jossa Python –ohjelmasi on. (Vinkki: Voit luoda tiedoston PyCharmissa!)

Muista palauttaa codePostiin myös tekemäsi guestbook.txt –tiedosto!

Lisätehtävä: Käytä JSON-formaattia vieraskirjassa (guestbook.json). Tallenna tässä tapauksessa jokaisesta viestistä itse viesti sekä päivämäärä + kellonaika (datetime-moduuli). **Huom:** helpointa on pitää viestien dictionaryja listassa niin JSONissa kuin Pythonissa!

Jos teet JSON-pohjaisen ratkaisun, älä palauta samassa koodissa txt-tiedosto –pohjaista ratkaisua!

Esimerkkejä ohjelman toiminnasta:

```
Haluatko lukea vai kirjoittaa vieraskirjaan? (l/k)
k
Kirjoita uusi viesti:
Hieno ohjelma tämä!
```

```
Haluatko lukea vai kirjoittaa vieraskirjaan? (l/k)
l
Terveisiä Suomesta!
Hieno ohjelma tämä!
```

Tehtävän tiedostonimi = **exercise10_2.py**

Tyypillinen koodimäärä: **12-24 riviä** (sisältää myös funktioiden koodit, tyhjiä rivejä ja kommentteja ei lasketa)



3. Tee ohjelma, joka antaa käyttäjälle satunnaisen mietelauseen! Käytä Moodlesta löytyviä mietelauseita ohjelmassasi (**wisewords.txt**).

Vinkki: Lataa **wisewords.txt** –tiedosto koodissasi muuttujaan, ja tee tiedostossa olevista mietelauseista oma lista. Tarkoitus on tuottaa satunnainen listan indeksi, jolla voidaan hakea satunnainen mietelause. Esim. jos listassa on 20 mietelauseita, silloin satunnainen indeksi on **random.randint(0, 19)**

Huom: wisewords.txt –tiedostoa ei tarvitse ladata codePostiin, se löytyy sieltä jo valmiiksi.

Kaikkien mietelauseiden määrä tiedostossa on juuri tehdyn listan koko (mutta muista - 1, koska indeksit alkavat nolasta)
esim. **max_index = len(fortunes_list) - 1**

Muista:

import random

Esimerkkejä ohjelman toiminnasta:

```
Päivän mietelause:  
Oppia ikä kaikki
```

```
Päivän mietelause:  
Rehellisyys maan perii
```

```
Päivän mietelause:  
Joka kerran keksitään, sitä aina arvelaan
```

Tehtävän tiedostonimi = **exercise10_3.py**

Tyypillinen koodimäärä: **7-14 riviä** (sisältää myös funktioiden koodit, tyhjiä rivejä ja kommentteja ei lasketa)



4. Lataa Moodlesta tiedosto nimeltä **"countries.json"**, ja lisää se samaan kansioon jossa Python-tiedostosi ovat. Lataa tiedosto Pythonilla omaan muuttujaansa, ja tulosta listan sisältö allekkain alla olevan esimerkin mukaisesti.

Jos aukaiset JSON-tiedoston PyCharmissa, huomaat, että se on lista, joka sisältää dictionaryjä. Dictionaryjen avaimet ovat "country" ja "capital".

countries.json –tiedostoa ei tarvitse ladata codePostiin, se löytyy sieltä jo valmiiksi!

Esimerkki ohjelman toiminnasta:

```
Kaikki maat ja pääkaupungit:  
Afghanistan: Kabul  
Albania: Tirana  
Algeria: Alger  
American Samoa: Fagatogo  
Andorra: Andorra la Vella  
Angola: Luanda
```



Tehtävän tiedostonimi = **exercise10_4.py**

Typillinen koodimäärä: **7-12 riviä** (sisältää myös funktioiden koodit, tyhjiä rivejä ja kommentteja ei lasketa).

Lisätehtävän myötä koodirivejä voi olla **20-30 riviä**

Lisätehtävä:

Kysy käyttäjältä listan tulostamisen jälkeen kirjain A-Z, ja tulosta vain sillä kirjaimella alkavien maiden tiedot.

Huom: Jos teet tämän lisätehtävän, muista tulostaa ohjelman alussa ensin kaikki maat ja pääkaupungit tiedostosta normaalisti. Esimerkiksi:

```
Minkä alkukirjaimen maat ja pääkaupungit tulostetaan?  
S  
Saint Helena: Jamestown  
Saint Kitts and Nevis: Basseterre  
Saint Lucia: Castries  
Saint Pierre and Miquelon: Saint-Pierre  
Saint Vincent and the Grenadines: Kingstown  
Samoa: Apia  
San Marino: San Marino  
Sao Tome and Principe: S  
Saudi Arabia: Riyadh  
Scotland: Edinburgh
```

Lisätehtäviä!

*Huom: Hyvä arvosana ei vaadi kaikkien lisätehtävien tekemistä.
Tärkeintä on, että jokaiselle löytyy jokin mielenkiintoinen vaihtoehto!*

5. Tee huoltokirjausohjelma, joka kysyy seuraavat tiedot käyttäjältä:

- Kirjaajan nimi
- Tilannekoodi
- Selite

Lisää tietoihin myös automaattisesti tämän hetkinen päivämäärä. (date -moduuli)

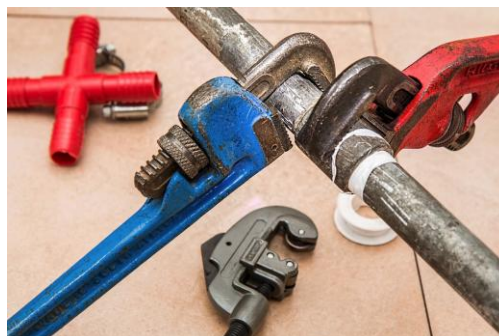
Tee kysytyistä tiedoista dictionary, ja tallenna se JSON-muodossa tiedostoon (esim. **maintenance.json**) ennen ohjelman päättymistä. Kun käynnistät ohjelman uudestaan, ja jos tiedosto on entuudestaan olemassa, näytä sen sisältö ennen kuin annat käyttäjälle mahdollisuuden kirjoittaa uudet tiedot sen tilalle. Esimerkki:

```
Viimeisin kirjaus...  
Pvm: 5.11.2019  
Hlö: Matti Meikäläinen  
Tilannekoodi: 36  
Viesti: Nurkkahuoneen ikkunassa vetoa, tiivisteet todnäk. vaihdon tarpeessa.
```

Tehtävän tiedostonimi = *exercise10_5.py*

Lisätehtävä:

Tallenna kaikki tiedostoon lisätyt huoltokirjaukset (älä kirjoita vanhan yli), ja lisää aina vain perään uusi huoltokirjaus. Näytä ohjelman käynnistyessä aina viimeisin huoltokirjaus. JSON on tässä hyödyllinen tallennusmuoto (lista joka koostuu dictionaryistä).



6. Excel/CSV:

Lataa Moodlesta serialtest.xlsx, ja avaa se Excelissä. Tallenna tiedosto CSV-muodossa, ja lataa se samaan kansioon jossa Python-tiedostosi ovat.

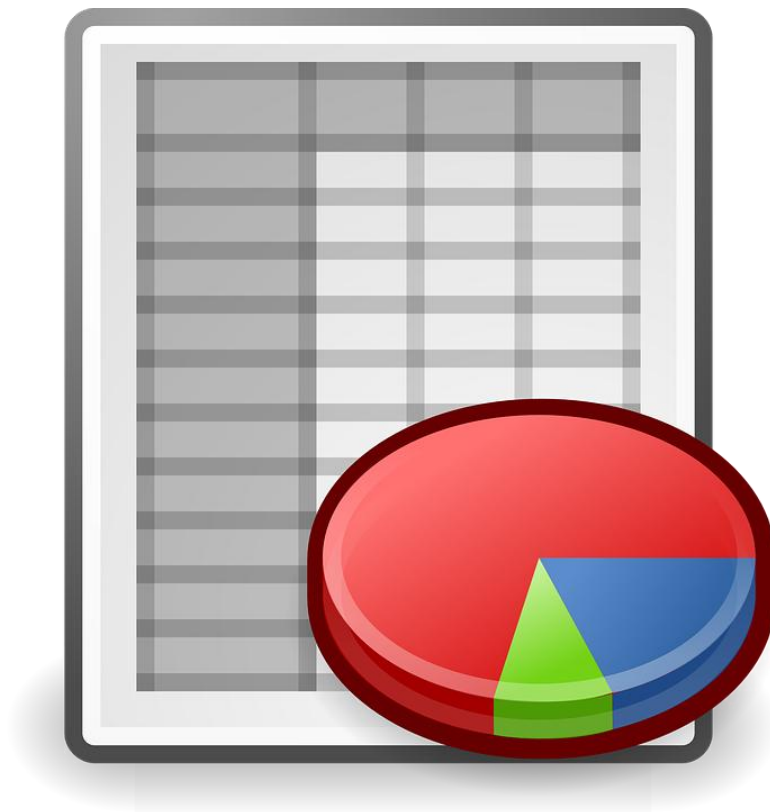
Huom: CSV:ssä arvot on eroteltu esim. pilkulla tai puolipisteellä, ja rivit rivinvaihdolla (\n).

Käytä Pythonin CSV-moduulia, ja lataa tiedoston sisältö kokoelmaan. Tulosta sisältö silmukoimalla Pythonissa.

Tehtävän tiedostonimi = *exercise10_6.py*

Lisähaaste:

Muokkaa lataamaasi sisältöä Pythonissa siten, että nostat jokaista palkkaa 100 €:lla, ja tallenna lopputulos toiseen CSV-tiedostoon, ja avaa se lopuksi Excelissä.



7. Tiedonkäsittely (CRUD = Create, Read, Update, Delete):

Tee ohjelma, joka hakee tiedostosta yrityksen tuotteiden nimet. Voit päättää itse mitä tuotteita kyseinen kuvitteellinen yritys myy. Näytä tuotteista numeroitu lista käyttäjälle, ja anna käyttäjälle mahdollisuus valita tuote rivinumerolla, jonka nimeä hän haluaa muokata. Muuta valittu tuote käyttäjän antamaan arvoon, ja tulosta numeroitu lista niin kauan kunnes käyttäjä haluaa lopettaa ohjelman käytön. Kun käyttäjä lopettaa ohjelman käytön, tallenna muutettu lista vanhan tiedoston päälle.

Tehtävän tiedostonimi = *exercise10_7.py*

Lisähaaste 1:

Muuta listaa niin, että se sisältää dictionaryjä eri tuotteista. Yhdestä tuotteesta tulee olla seuraavat tiedot: *nimi, kategoria, hinta*

Anna käyttäjälle mahdollisuus muokata minkä tahansa tuotteen mitä tahansa tietoa.

Lisähaaste 2:

Luokittele edellisen lisähaasteen tiedot niin, että käyttäjä voi selata yhden kategorian tuotteita aina kerrallaan.

Lisähaaste 3:

Anna käyttäjälle mahdollisuus myös lisätä uusia ja poistaa vanhoja tuotteita.



Tästä tehtävästä voi saada enemmän lisäpisteitä kuin tyypillisestä lisätehtävästä!