

Ohjelmoinnin peruskurssi Y1

CS-A1111

Tiedostojen käsittely

Oppimistavoitteet: tämän videon jälkeen

- ▶ Osaat kirjoittaa ohjelman, joka lukee rivejä tekstitiedostosta ja tekee luetuille riveille haluttuja asioita.

Tiedoston rivien lukeminen for-käskyllä

- ▶ Jos ohjelman on luettava kaikki tiedoston rivit, on usein helpointa käydä ne läpi for-käskyn avulla.
- ▶ Käskyn yleinen muoto on

```
for rivimuuttuja in lahtotiedosto:  
    tee jotain riville rivimuuttuja
```
- ▶ Käskyyn ei tarvitse kirjoittaa lainkaan rivin tiedostosta lukevaa käskyä (esim. `readline`).
- ▶ **Tärkeää:** jos ohjelmassa luetaan tiedoston rivit for-käskyn avulla, ei samassa ohjelmassa pidä käyttää `readline`-metodia.
 - ▶ *Poikkeus:* jos tiedetään varmasti, että tiedoston alussa on 1–2 otsikkoriviä, joita ei ole tarkoitus käsitellä, voidaan ne "lukea pois" `readline`-metodilla ennen loput rivit läpikäyvää for-käskyä.

Esimerkki tiedoston lukemisesta for-käskyllä

```
def main():
    nimi = input("Anna luettavan tiedoston nimi: ")
    try:
        lahtotiedosto = open(nimi, "r")
        for rivi in lahtotiedosto:
            rivi = rivi.rstrip()
            print(rivi)
        lahtotiedosto.close()
    except OSError:
        print("Virhe tiedoston", nimi,
              "lukemisessa. Ohjelma paattyy.")

main()
```

Tekstitiedosto vs. binääritiedosto

- ▶ Tiedostot jaetaan tekstitiedostoihin ja binääritiedostoihin.
- ▶ Tekstitiedostossa tiedot on tallennettu merkkeinä, esimerkiksi luku 147 merkkeinä 1, 4 ja 7.
- ▶ Tekstitiedostoa voi muokata millä tahansa tekstieditorilla.
- ▶ Binääritiedostossa tiedot on esitetty binääriesitysmuodossa, esimerkiksi luku 147 vastaavana binäärilukuna.
- ▶ Binääritiedostoa ei yleensä pysty käsittelemään järkevästi tavallisella tekstieditorilla.
- ▶ Tällä kurssilla opetetaan ainoastaan tekstitiedostojen käsittely.

Lukujen lukeminen tiedostosta

- ▶ Python-tulkki palauttaa tiedostosta luetut rivit aina merkkijonoina.
- ▶ Jos tiedostossa on lukuja, pitää luetut rivit muuttaa tyyppimuunnoksella oikeantyyppiksiksi.
- ▶ Jos riviä ei voida muuttaa luvuksi, aiheutuu `ValueError`, joka on syytä käsitellä.
- ▶ Jos samalla rivillä on useita lukuja, pitää rivi ensin jakaa. Tyyppimuunnos tehdään vasta jaon tuloksena syntyneille luvuille.

Desimaalilukuja tiedostosta, koodi

```
def main():
    nimi = input("Mista tiedostosta lampotilat luetaan: ")
    summa = 0.0
    lkm = 0
    try:
        lampotiedosto = open(nimi, "r")
        for rivi in lampotiedosto:
            rivi = rivi.rstrip()
            lampotila = float(rivi)
            summa += lampotila
            lkm += 1
        lampotiedosto.close()
```


Desimaalilukuja tiedostosta, koodi jatkuu

```
if lkm == 0:
    print("Tiedostossa ei ollut yhtään lampotilaa.")
else:
    keskiarvo = summa / lkm
    print("Lampotilojen keskiarvo on", keskiarvo)
except OSError:
    print("Virhe tiedoston", nimi,
          "lukemisessa. Ohjelma paattyy.")
except ValueError:
    print("Virheellinen rivi tiedostossa", nimi,
          "- ohjelma paattyy.")
```

main()

Lukujen lukeminen csv-tiedostosta

- ▶ Entä, jos tiedostossa on jokaisella rivillä lämpötilan lisäksi päivämäärä, jolloin lämpötila on mitattu?
- ▶ Rivillä on ensin päivämäärä, sitten pilkku ja sen jälkeen lämpötila.
- ▶ Tällaisia *csv-tiedostoja* voi tehdä helposti esimerkiksi taulukkolaskentaohjelmilla (csv = comma-separated values).

Lukeminen csv-tiedostosta, koodi

```
def main():
    nimi = input("Mista tiedostosta lampotilat luetaan: ")
    summa = 0.0
    lkm = 0
    try:
        lampotiedosto = open(nimi, "r")
        for rivi in lampotiedosto:
            rivi = rivi.rstrip()
            osat = rivi.split(",")
            if len(osat) == 2:
                lampotila = float(osat[1])
                summa += lampotila
                lkm += 1
            else:
                print("Virheellinen rivi", rivi)
        lampotiedosto.close()
```

Ohjelma jatkuu

```
if lkm == 0:
    print("Tiedostossa ei ollut yhtään lampotilaa.")
else:
    keskiarvo = summa / lkm
    print("Lampotilojen keskiarvo on", keskiarvo)
except OSError:
    print("Virhe tiedoston", nimi,
          "lukemisessa. Ohjelma paattyy.")
except ValueError:
    print("Virheellinen rivi tiedostossa", nimi,
          "- ohjelma paattyy.")
```

```
main()
```