

Ohjelmoinnin perusteet Y Python

T-106.1208

25.2.2009

Syötteessä useita lukuja samalla rivillä

- ▶ Seuraavassa esimerkissä käyttäjä antaa useita lukuja samalla rivillä.
- ▶ Luvut on erotettu toisistaan välilyönnillä.
- ▶ Rivi jaetaan ensin osiin `split`-metodilla. Tämän jälkeen luvut ovat listassa, mutta merkkijonoina, ei lukuina.
- ▶ Listassa olevat merkkijonot voidaan muuttaa desimaaliluvuiksi tyyppinmuunnoksella.
- ▶ Ohjelma laskee rivillä olevien lukujen summan ja tulostaa sen.

Lukuja samalla rivillä, koodi

```
def main():  
    print "Anna luvut samalla rivillä,"  
    print "erota toisistaan valilyonnilla."  
    lukurivi = raw_input()  
    luvut_tekstina = lukurivi.split()  
    summa = 0.0  
    for luku in luvut_tekstina:  
        summa += float(luku)  
    print "Lukujen summa on %.2f" % (summa)  
  
main()
```

Listan alkiona lista

- ▶ Miten esitetään matriiseja Python-ohjelmissa?
- ▶ Ratkaisumahdollisuus: käytetään listaa, jonka kukin alkio on matriisin yksi rivi.
- ▶ Kutakin riviä esitetään desimaaliluvuista koostuvalla listalla.
- ▶ Matriisia kuvaavan listan alkiot ovat siis itsekin listoja.

Esimerkki

- ▶ Esimerkki listan luomisesta:

```
>>> matriisi1 = [[1.0, 5.5, 2.7], [4.3, 2.2, 8.9]]
```

- ▶ `matriisi1` viittaa siis nyt listaan, jonka alkiona on kaksi listaa.
- ▶ `matriisi1[1]` viittaa puolestaan listaan, jonka alkioina on kolme desimaalilukua.
- ▶ Alkion `matriisi1[1][2]` arvo on 8.9.

Esimerkkiohjelma: matriisien yhteenlasku

- ▶ Seuraava esimerkkiohjelma lukee käyttäjältä kaksi matriisia ja laskee niiden summan.
- ▶ Ohjelmassa on omat funktiot yhden matriisin lukemiseen, kahden parametrina annetun matriisin summan laskemiseen ja yhden matriisin tulostamiseen.
- ▶ Pääohjelmassa on pidetty huolta siitä, että matriisit ovat samankokoisia ja että sekä rivien että sarakkeiden määrä on nollaa suurempi. Jos tätä ei tarkisteta pääohjelmassa, pitäisi vastaavat tarkistukset tehdä matriiseja käsittelevissä funktioissa.
- ▶ Matriisin tulostuksessa `print`-käskyn lopussa on pilkku. Tällä saadaan aikaiseksi se, että tulostuksen loppuun ei tule rivinvaihtoa. Näin matriisin rivin kaikki alkiot saadaan tulostuksessa samalle riville.

Matriisien yhteenlasku, koodi

```
def lue_matriisi(rivilkm, sarakelkm):  
    matriisi = []  
    print "Anna matriisin alkiot riveittäin,"  
    print rivilkm, "rivia ja", sarakelkm, "saraketta."  
    for i in range(rivilkm):  
        rivi = [0.0] * sarakelkm  
        for j in range(sarakelkm):  
            syote = raw_input()  
            rivi[j] = float(syote)  
        matriisi.append(rivi)  
    return matriisi
```

Matriisien yhteenlasku, koodi jatkuu

```
def laske_summa(mat1, mat2):  
    summamat = []  
    rivimaara = len(mat1)  
    sarakemaara = len(mat1[0])  
    for i in range(rivimaara):  
        summarivi = [0.0] * sarakemaara  
        for j in range(sarakemaara):  
            summarivi[j] = mat1[i][j] + mat2[i][j]  
        summamat.append(summarivi)  
    return summamat
```

Matriisien yhteenlasku, koodi jatkuu

```
def tulosta_matriisi(matri):  
    rivit = len(matri)  
    sarakkeet = len(matri[0])  
    for i in range(rivit):  
        for j in range(sarakkeet):  
            print "%8.2f" % (matri[i][j]),  
        print  
  
def main():  
    print "Ohjelma laskee kahden matriisin summan."  
    syote = raw_input("Anna rivien lukumaara: ")  
    riveja = int(syote)  
    syote = raw_input("Anna sarakkeiden lukumaara: ")  
    sarakkeita = int(syote)
```

Matriisien yhteenlasku, koodi jatkuu

```
if riveja <= 0 or sarakkeita <= 0:
    print "Liian vahan riveja tai sarakkeita."
else:
    matriisi1 = lue_matriisi(riveja, sarakkeita)
    matriisi2 = lue_matriisi(riveja, sarakkeita)
    summa = laske_summa(matriisi1, matriisi2)
    print "Matriisin"
    tulosta_matriisi(matriisi1)
    print "ja matriisin"
    tulosta_matriisi(matriisi2)
    print "summa on"
    tulosta_matriisi(summa)
```

```
main()
```

Sanakirja

- ▶ Monissa sovelluksissa on tallennettava rakenteeseen *avain-arvo-pareja*. Myöhemmin rakenteesta halutaan etsiä tiettyyn avaimeen liittyvää arvoa.
- ▶ Esimerkkejä: puhelinluettelo, opiskelijarekisteri, yrityksen asiakasrekisteri, autorekisteri.
- ▶ Halutaan, että tehokkaan hakemisen lisäksi rakenteeseen pystyy myös helposti lisäämään uusia avain-arvo-pareja sekä poistamaan pareja. Lisäksi avaimeen liittyvää arvoa voidaan muuttaa.
- ▶ Yksinkertainen ratkaisu: käytetään listaa, jonka alkoina on avain-arvo-pareja. Ongelmia:
 - ▶ Hidas haku
 - ▶ Jos hakua nopeutetaan pitämällä avaimet järjestyksessä, lisäys ja poisto hankaloituvat.
- ▶ Pythonissa on valmis rakenne, *sanakirja* (engl. dictionary), jossa sekä haut, lisäykset että poistot pystytään tekemään tehokkaasti.

Sanakirjan luonti ja käyttö

- ▶ Tyhjän sanakirjan voi luoda aaltosulkujen avulla:

```
>>> puh_luettelo = {}
```

- ▶ Sanakirjaa luodessa voi samalla jo antaa siihen liitettäviä avain-arvo-pareja:

```
>>> puhelinluettelo = {"Teekkari Teemu" : "050-12345", \
... "Fyysikko Tiina" : "045-234567", "Kemisti Kalle" : \
... "040-765432"}
```

- ▶ Haluttuun avaimeen liittyvän arvon saa selville ilmauksella sanakirja[avain], esimerkiksi

```
>>> print puhelinluettelo["Fyysikko Tiina"]
045-234567
```

Sanakirja: avaimen haku ja olemassaolo

- ▶ Edellisen kalvon hakutapa johtaa kuitenkin ohjelman kaatumiseen, jos haettua avainta ei löydy sanakirjasta:

```
>>> print puhelinluettelo["Virtanen Maija"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Virtanen Maija'
```

- ▶ Operaattorin `in` avulla voi tutkia, onko haettava avain sanakirjassa:

```
>>> nimi = "Virtanen Maija"
>>> if nimi in puhelinluettelo:
...     print puhelinluettelo[nimi]
... else:
...     print "Nimea ei löydy luettelosta"
...
Nimea ei löydy luettelosta
```

Sanakirja: avainten lisääminen ja arvon muuttaminen

- Sijoituskäskyn avulla sanakirjaan voi lisätä uusia avain–arvo-pareja ja muuttaa sanakirjassa jo oleviin avaimiin liittyviä arvoja.

```
>>> puhelinluettelo["Rakentaja Niina"] = "0400-123"  
>>> puhelinluettelo["Kemisti Kalle"] = "041-56789"  
>>> print puhelinluettelo  
{'Kemisti Kalle': '041-56789', 'Fyysikko Tiina':  
'045-234567', 'Teekkari Teemu': '050-12345',  
'Rakentaja Niina': '0400-123'}
```

Sanakirja: avainten läpikäynti

- Sanakirjan avaimet voi käydä läpi for-käskyn avulla samaan tapaan kuin listan alkiot.

```
>>> for nimi in puhelinluettelo:  
...     print nimi  
...
```

Kemisti Kalle

Fyysikko Tiina

Teekkari Teemu

Rakentaja Niina

```
>>> for nimi in puhelinluettelo:
```

```
...     print "%16s %12s" % (nimi, puhelinluettelo[nimi])  
...
```

Kemisti Kalle 041-56789

Fyysikko Tiina 045-234567

Teekkari Teemu 050-12345

Rakentaja Niina 0400-123

Sanakirja: avaimen poistaminen

- ▶ Sanakirjasta voi poistaa avaimen ja siihen liittyvän arvon del-operaattorilla:

```
>>> del puhelinluettelo["Kemisti Kalle"]
>>> print puhelinluettelo
{'Fyysikko Tiina': '045-234567', 'Teekkari Teemu':
'050-12345', 'Rakentaja Niina': '0400-123'}
```

Esimerkki: puhelinluettelo

- ▶ Seuraavassa esimerkkiohjelmassa käyttäjä syöttää ensin haluamansa määrän nimiä ja puhelinnumeroita sanakirjarakenteen avulla toteutettuun puhelinluetteloon.
- ▶ Tämän jälkeen käyttäjä voi hakea luettelosta yhteen nimeen liittyvän puhelinnumeron.
- ▶ Harjoitustehtävänä on toteuttaa monipuolisempi puhelinluettelo.
- ▶ Käytännössä puhelinluettelon tiedot kannattaisi lukea tiedostosta, niin ne olisivat käytössä ohjelman suorituskerrasta toiseen.

Puhelinluettelo, koodi

```
def lue_puhelinnumerot():  
    print "Anna lisattavat nimet ja numerot."  
    print "Nimi ja puhelinnumero samalla rivilla,"  
    print "valissa kaksoispiste."  
    print "Lopeta tyhjalla rivilla."  
    puhelinluettelo = {}  
    rivi = raw_input()  
    while len(rivi) > 0:  
        tiedot = rivi.split(":")  
        nimi = tiedot[0]  
        numero = tiedot[1]  
        puhelinluettelo[nimi] = numero  
        rivi = raw_input()  
    return puhelinluettelo
```

Puhelinluettelo, koodi jatkuu

```
def etsi_numero(puhelintiedot):  
    etsitty = raw_input("Kenen numero haetaan? ")  
    if etsitty in puhelintiedot:  
        print "Numero on", puhelintiedot[etsitty]  
    else:  
        print "Nimea ei löydy luettelosta."  
  
def main():  
    luettelo = lue_puhelinnumerot()  
    etsi_numero(luettelo)  
  
main()
```