

# Python-ohjelmointi

# Harjoitus 11

## TAVOITTEET

- Kerrataan harjoituksen 10 keskeisiä sisältöjä.
- Kerrataan ja vahvistetaan ohjelman suunnittelun vaiheita.
- Tutustutaan vuokaavioon ohjelman suunnittelun työkaluna.
- Tutustutaan aliohjelmien käyttämiseen.

## Matematiikan sisällöt

Tehtävissä harjoitellaan ja syvennetään verrannollisuuden ja yhtälöiden osaamista.

---

Ohjelmointiin käytettävä alusta löytyy osoitteesta <http://sade-oppiminen.herokuapp.com/>

---

## KERTAUSTA HARJOITUKSESTA 10

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen string-kirjaston käyttö yhdessä random-kirjaston kanssa.

Run

Tee ohjelma, jonka arpoo ja tulostaa nelinumeroisen avainluvun.

### TEHTÄVÄ 2

Tavoitteena on palauttaa mieleen merkkijonoja.

Run

Mitä alla oleva ohjelma tekee? Selitä jokaisen rivin toiminta ja mitä ohjelma tulostaa. Yritä selittää ohjelman toimintaa sanallisesti, ennen kuin kirjoitat koodin ohjelmointialustalle ja tarkistat ohjelman toiminnan.

```
import string

a = string.ascii_lowercase
print a[10]+a[4]+2*a[17]+a[14]+a[8]+a[13]
print a[21]+a[4]+2*a[17]+a[0]+a[13]+a[19]+a[14]
```

## KERTAUSTA OHJELMAN SUUNNITTELUN TYÖVAIHEISTA

Säde 1 Harjoituskokoelmassa 6 harjoiteltiin ohjelman suunnittelua muuttamalla ratkaistavana oleva ongelma pseudokielelle eli yhdistelmäksi suomen kieltä ja ohjelmointikieltä. Ohjelmasuunnitelman avulla saadaan kirjoitettua ohjelman tarvitsema koodi.

Ohjelmasuunnitelmassa suunnitellaan tarvittavat muuttujat sekä apumuuttujat ja muut ohjelmalliset rakenteet. Python-kielen komennot ja muut kielen perusrakenteet kirjoitetaan Python-kielillä. Ohjelman toiminta ja ulospäin näkyvät kohdat kirjoitetaan lyhyillä, kuvaavilla lauseilla.

### TEHTÄVÄ 3

Tavoitteena on palauttaa mieleen ohjelmasuunnitelman yhteys lopulliseen koodiin.

Run

Bändi on tehnyt ohjelman, jolla he saavat laskettua, kuinka paljon kukin jäsen saa keikkapalkkiosta.

Ohjelmasuunnitelma on seuraava:

```
# palkkio = int(Kysytään keikkapalkkio.)
# Vähennetään keikkapalkkiosta keikkabussin kulut 75 €.
# jakajat = int(Kysytään bändin jäsenmäärä.)
# Lasketaan ja tulostetaan palkkio_per_jasen.
# Pyöristetään vastaus kahteen desimaaliin.
```

Järjestä alla olevat koodit oikeaan järjestykseen, jotta ohjelma toimii suunnitelman mukaisesti.

A	<code>jakajat = float(input('Keikalla olleiden bändin jäsenten lukumäärä.'))</code>
B	<code>palkkio = palkkio - 75</code>
C	<code>print 'Jokainen saa ' + str(palkkio_per_jasen) + ' e.'</code>
D	<code>palkkio = int(input('Anna keikkapalkkio.'))</code>
E	<code>palkkio_per_jasen = round(palkkio/jakajat,2)</code>

### TEHTÄVÄ 4

Tavoitteena on harjoitella ongelman kirjoittamista pseudokielelle ja siitä Python-koodiksi.

Run

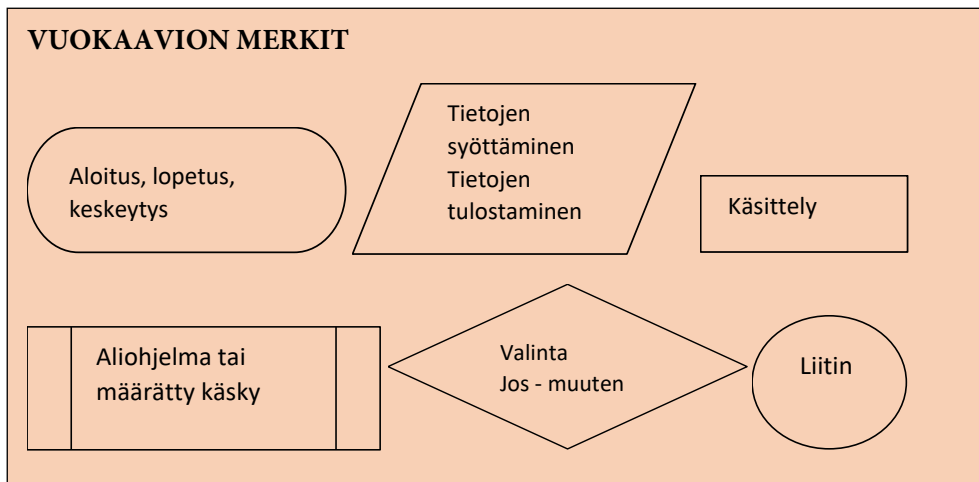
Tee ohjelma, jolla ratkaiset seuraavan Säde 2 -kirjan sivun 145 tehtävän 14.

14. Hillosipulipurkin sisältö painaa 320 g, josta vettä ja mausteita on yhteensä 130 g. Kuinka paljon yksi hillosipuli painaa, jos sipuleita on purkissa 170 kappaletta?

- Tee ohjelmasuunnitelma. Mieti tarvittavat muuttujat ja apumuuttujat.
- Tee ohjelmakoodi.

## VUOKAAVIO OHJELMAN SUUNNITTELUN VÄLINEENÄ

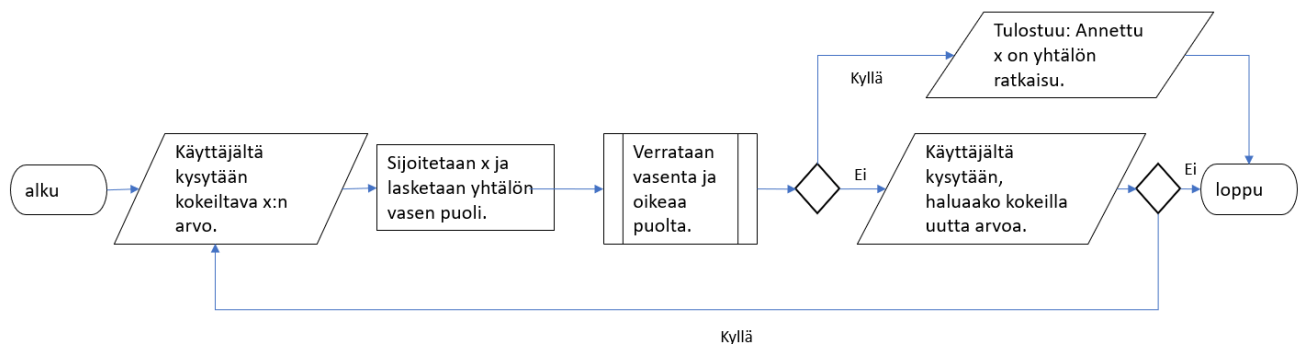
Vuokaavion avulla voidaan kuvata ohjelman toimintojen järjestystä, ohjelman etenemistä tai tietyn algoritmin eli toimintakomentojen etenemistä. Kun ohjelman idea on keksitty, voidaan muodostaa vuokaavio, josta laaditaan pseudokielinen suunnitelma varsinaisen koodin kirjoittamisen pohjaksi.



Vuokaavio voidaan piirtää piirto-ohjelmalla tai jollain toimisto-ohjelmalla.

### Esimerkki 1

Ohjelmassa tutkitaan, onko käyttäjältä kysytty luku yhtälön  $\frac{3x}{4} = -6$  ratkaisu. Ohjelman toimintaa kuvataan seuraavalla vuokaaviolla.



- Kirjoita ohjelmasuunnitelma.
- Kirjoita ohjelman koodi.



## Ratkaisu

a)

```
# Määritetään muuttuja x ja sille saadaan arvo kysymällä se
käyttäjältä.
# Vasen = sijoitetaan x:n arvo ja lasketaan vasen puoli.
# Oikea = - 6
# Verrataan, onko vasen == oikea.
# Jos puolet ovat yhtä suuria, tulostetaan "Annettu x on yhtälön
ratkaisu." ja ohjelma loppuu. Muuten tulostuu: Haluatko kokeilla
toisella x:n arvolla.
# Jos käyttäjä vastaa kyllä, palataan takaisin kysymään
käyttäjältä uusi arvo. Jos vastaa ei, ohjelma loppuu.
```

b)

```
vastaus = "K"
while vastaus == "K" or vastaus == "k":
    x = int(input('Anna x:n arvo.'))
    vasen = 3 * x / 4
    oikea = -6
    if vasen == oikea:
        print u'Annettu x on yhtälön ratkaisu.'
    else:
        vastaus = input('Haluatko kokeilla toista x:n arvoa? K/E')
```

Jotta ohjelma palaa viimeisen kysymyksen kyllä-vastauksen jälkeen takaisin arvon kysymiseen, täytyy koko ohjelma tehdä while-silmukan sisään. Toistorakenne while toteutetaan aina, kun viimeiseen kysymykseen vastataan kyllä.

## TEHTÄVÄ 5

Tavoitteena on harjoitella vuokaavion tekemistä.

Run

Tee vuokaavio alla olevan ohjelman toiminnasta. Ohjelma laskee käyttäjän syöttämien lukujen keskiarvon. Vuokaavion tekoa voi helpottaa, jos teet ensin pseudokielisen ohjelmasuunnitelman.

```
numero = 0
i = 0
vastaus = "k"
while vastaus == "k":
    a = float(input(' Anna arvosana.'))
    numero = numero + a
    i = i + 1
    vastaus = input(u'Haluatko antaa lisää numeroita? k/ e?')
keskiarvo = numero / i
print keskiarvo
```

R

## ALIOHJELMAT

Tähän asti olemme käyttäneet Python-kielen sisältämiä valmiita komentoja. Mikäli ohjelmia tehdään vain näitä käyttäen, tulee niistä helposti pitkiä ja välillä koodia on vaikea seurata. Ohjelman toiminnan seuraamista helpottaa se, että toimintoja jaetaan pienempiin osiohjelmiin. Näitä pienempiä ohjelmia kutsutaan aliohjelmiksi. Aliohjelma on itsenäinen ohjelman osa, joka suorittaa jonkin yksittäisen, yleensä pienimuotoisen, toiminnallisuuden. Aliohjelmalla voidaan myös korvata ohjelmassa toistuvasti esiintyvät samat toiminnallisuudet.

Ohjelmat muodostuvat siis pääohjelmasta ja aliohjelmista. Pääohjelmassa aliohjelmia kutsutaan nimellä suorittamaan aliohjelman sisältämä toimenpide. Ohjelmassa pääohjelma on yleensä viimeisenä.

Aliohjelmaa määriteltäessä sille annetaan nimi ja määritetään aliohjelman sisältämä toimenpide. Aliohjelmalle välitettäviä tietoja kutsutaan nimellä parametrit.

Aliohjelman määrittely aloitetaan *def*-sanalla, jolloin Python-kääntäjä ymmärtää, että tämä osa ohjelmaa suoritetaan vasta, kun sitä kutsutaan pääohjelmassa.

Kun aliohjelma on suoritettu, se palauttaa viimeisimmät arvot pääohjelman käyttöön, jossa ne voidaan esimerkiksi tulostaa tai käyttää toisen aliohjelman parametrinä.

### Esimerkki 2

Tee ohjelma, jossa aliohjelma laskee ja tulostaa sille annettujen kahden luvun summan. Pääohjelma kutsuu tätä aliohjelmaa.

### Ratkaisu

```
# Tehdään aliohjelma, jonka nimi on summa.
# Aliohjelman parametrit ovat kaksi lukua a ja b.
def summa(a, b):
    summa = a + b
    print summa

#Pääohjelma kutsuu aliohjelmaa summa ja vie aliohjelmalle tietoina
kaksi lukua.
summa(7, 9)
summa(10, -15)
```

Run

Ohjelman toimimista kannattaa testata muutamalla aliohjelman kutsulla.

## TEHTÄVÄ 6

Tavoitteena on harjoitella aliohjelman tekemistä ja kutsumista.

Run

Tee ohjelma, jossa lasketaan kartasta mitatun matkan pituus luonnossa metreinä. Kartan mittakaava on 1 : 10 000 eli 1 cm kartalla vastaa 10 000 cm:iä luonnossa.

- Tee aliohjelma, joka muuntaa kartalta mitatun matkan todelliseen matkaan luonnossa.
- Lisää aliohjelmaan muunnos senttimetreistä metreihin ja vastauksen tulostaminen metreinä. Tulosteessa olisi hyvä olla myös yksikkö mukana.
- Lisää ohjelmaan aliohjelman kutsu ja selvitä sen avulla, kuinka pitkää matka luonnossa vastaa
  - a) 3 cm
  - b) 8 cm
  - c) 12 cmkartalla.

## TEHTÄVÄ 7

Tavoitteena on harjoitella aliohjelman käyttöä.

Run

Tee ohjelma, joka muuntaa reseptissä olevat valmistusaineiden määrät vastaamaan haluttua annosmäärää.

- Tee aliohjelma, joka saa parametriksi haluttujen annosten lukumäärän.
  - Tee laskukaava, joka laskee määrien laskemiseen tarvittavan kertoimen (*halutut annokset/ohjeesta tulevat annokset*).
  - Laske ja tulosta tarvittavat valmistusainemäärät. Tulosteessa pitäisi näkyä myös yksikkö ja mikä valmistusaine on kyseessä.
- Selvitä aliohjelmaa käyttäen, kuinka paljon aineita tarvitaan
  - a) 15 annokseen
  - b) 10 annokseen
  - c) yhteen annokseen.

### Mustikkarahkavälipala

3 dl luonnonjugurttia

200 g rahkaa

2 dl mustikkakeittoa

1 tl sokeria

1 tl vanilliinisokeria

Sekoita ainekset juomaksi. Ohjeella valmistuu kolme annosta.

### Esimerkki 3

Tee ohjelma, jossa aliohjelma laskee käyttäjän ilmoittamien työtuntien perusteella hänen palkkansa. Tuntipalkka on 11,45 €. Palkka tulostetaan pääohjelmassa.

### Ratkaisu

Aliohjelma saa parametrina käyttäjältä kysytyt työtunnit. Käyttäjältä kysytyt tunnit on tallennettu apumuuttujaan, joka vieään parametrin arvona aliohjelman kutsuun. Aliohjelman määrittämisessä parametrin nimenä ei voi olla samaa nimeä kuin pääohjelmassa käytettävässä muuttujassa. Käyttäjältä kysytyt tunnit tallennetaan muuttujaan *tunnit*. Aliohjelman parametriksi laitetaan tunnin tunnus *h*.

Jotta aliohjelman tieto välittyy pääohjelman tulostukseen, täytyy aliohjelman lopuksi kertoa, mitä aliohjelmasta palautetaan pääohjelman käyttöön. Tämä tehdään *return*-komennolla.

```
# Määritellään aliohjelma.
def palkka(h):
    tili = h * 11.45

    return tili

# Määritellään muuttujan tunnit arvoksi käyttäjältä kysyttävät työtunnit.
tunnit = int(input( u'Ilmoita tekemiesi tuntien lukumäärä.'))

# Kutsutaan aliohjelmaa.
print 'Palkkasi on ' + str(palkka(tunnit)) + 'e'
```

### TEHTÄVÄ 8

Tavoitteena on tehdä ohjelma, jolla harjoitellaan aliohjelmien tekemistä ja kutsumista.

Run

Tee ohjelma, jossa lasketaan nopeus, kun matka ja aika on annettu.

- Tee aliohjelma *nopeus*, joka saa parametrin *matka* ja *aika*. Koska muuttujia *matka* ja *aika* käytetään pääohjelmassa, parametreina voidaan käyttää näiden suureiden tunnuksia eli  $matka = s$  ja  $aika = t$ .
  - Kirjoita nopeuden laskukaava
  - Tulosta saatu nopeus.
- Pääohjelman alussa käyttäjältä kysytään arvot matkalle ja ajalle. Muunna saadut arvot liukuluvuiksi.
- Kutsutaan aliohjelmaa *nopeus*, jonka parametrien arvoiksi annetaan muuttujat *matka* ja *aika*.

## TEHTÄVÄ 9

Tavoitteena on tehdä aliohjelmaa hyödyntävä ohjelma, jossa kerrataan prosenttilaskentaa ja pyöristämistä.

Run

Tee ohjelma, joka ilmoittaa osuuden prosentteina.

- Käyttäjältä kysytään arvot, jotka tallennetaan muuttujiin.
- Muuttujat ovat parametrien arvoja aliohjelman kutsussa.
- Aliohjelmassa lasketaan osuus desimaalilukuna ja muunnetaan desimaaliluku prosenttiluvuksi siten, että vastaus pyöristyy yhteen desimaaliin.

## TEHTÄVÄ 10

Tavoitteena on tehdä ohjelma, jossa aliohjelmien tuloksia käytetään pääohjelmassa.

Run

Tee ohjelma, joka kertoo käyttäjälle, kuinka paljon hänen ilmoittamansa euromäärä on muissa valuutoissa. Tee erikseen aliohjelmat esimerkiksi eurojen muuttamisesta punniksi, dollareiksi ja kruunuiksi. Etsi vaihtokurssien ajantasaiset arvot netistä.

Ohjelma kysyy käyttäjältä, minkä euromäärän hän haluaa tietää muissa valuutoissa. Pääohjelman lopuksi tulostetaan tieto, kuinka paljon ilmoitettu euromäärä on muissa valuutoissa.