

# Python-ohjelmointi

# Harjoitus 14

## TAVOITTEET

- Kerrataan konnagrafiikan perustoiminnot sekä merkkijonojen käsittely.
- Vahvistetaan aliohjelmien suunnittelua, tekemistä ja käyttämistä osana omaa ohjelmaa.
- Harjoitellaan koodin tulkintaa ja ohjelman toiminnan testausta ilman konetta.

## Matematiikan sisällöt

Tehtävissä harjoitellaan ja syvennetään seuraavia aiheita: riippuvuuden käsite, kuvaajan piirtäminen, suoran yhtälöön liittyvien käsitteiden hallinta ja suoran piirtäminen.

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

## KERTAUSTA HARJOITUKSESTA 13

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen merkkijonojen käsittelyä.

Run

Tee ohjelma, jolla voit erottaa omiksi lähtöarvoikseen käyttäjän syöttämästä koordinaattipisteestä  $x$ -koordinaatin ja  $y$ -koordinaatin. Käyttäjää pyydetään antamaan koordinaattipiste ilman sulkeita ja erottamaan koordinaatit toisistaan pilkulla.

Ohjelma tulostaa ”Antamasi pisteen  $x$ -koordinaatti on \_\_\_ ja  $y$ -koordinaatti on\_\_\_).

merkkijonon jakaminen osiin  
osat = merkkijono.split("merkki, jonka kohdalta jaetaan")

## KERTAUSTA MERKKIJONOISTA

### TEHTÄVÄ 2

Tavoitteena on palauttaa mieleen merkkijonojen käsittelyä.

Run

Tee ohjelma, jolla voit tutkia, ovatko annetut suorat yhdensuuntaisia.

- Tee aliohjelma, joka erottaa  $x$ :n kertoimen eli kulmakertoimen suoran yhtälöstä ja palauttaa sen.

- Anna aliohjelman nimeksi *kulmakerroin*.  
Aliohjelma saa parametrinaan suoran yhtälön.
- Etsi suorasta ”=”-merkin indeksi.
- Etsi suorasta ” $x$ ”-merkin indeksi.
- Kulmakerroin on näiden merkkien väli.
- Testaa aliohjelman toimintaa seuraavilla kutsuilla
  - `print kulmakerroin('y=2x + 1')`

tietyn merkin sijainti merkkijonossa  
`merkkijono.index("etsittävä_merkki")`

- print kulmakerroin( $y = -2x + 1$ )
  - Jos aliohjelma toimii, poista testausrivit.
- Tee aliohjelma, joka vertaa kahden suoran kulmakertoimia ja palauttaa tiedon, onko kyse leikkaavista suorista vai yhdensuuntaisista suorista. Tämä aliohjelma kutsuu ensimmäistä aliohjelmaa.
  - Anna aliohjelmalle nimi *yhdensuuntaisuus*.
  - Tee ehtorakenne if – else. Jos ensimmäisen suoran kulmakerroin on yhtä suuri kuin toisen suoran kulmakerroin, tulostetaan ”Suorat ovat yhdensuuntaiset.”. Muutoin tulostetaan ”Suorat ovat leikkaavat.”
- Pääohjelmassa
  - kysytään käyttäjältä kaksi suoraa muodossa  $y = ax + b$ .
    - Tee muuttujat *suora1* ja *suora2* joihin käyttäjän antamat suoran yhtälöt tallennetaan.
  - kutsutaan aliohjelmaa, jossa kulmakertoimia verrataan ja annetaan ohjelmalle syötteeksi käyttäjän antamat suorat.

### TEHTÄVÄ 3

Tavoitteena on vahvistaa merkkijonojen käsittelyä.

Run

- Tee ohjelma, jossa käyttäjä antaa suoran yhtälön muodossa  $y = ax + b$  eli ratkaistussa muodossa. Tee
- ohjelma, joka pilkkoo käyttäjän antaman suoran yhtälön erottaen siitä  $x$ :n kertoimen ja vakiotermin. Ohjelma tulostaa lopuksi alkuperäisen yhtälön,  $x$ :n kertoimen ja vakiotermin tyyliin yhtälö: ” $y = ax + b$ , jossa kulmakerroin =  $a$ , vakiotermi =  $b$ ”
  - hyödynnä kertoimen erottamisessa tehtävän 2 ohjelmaa ja tee siitä tämän ohjelman aliohjelma.
  - tee aliohjelma vakiotermin erottamiselle suoran yhtälöstä.

### TEHTÄVÄ 4

Tavoitteena on palauttaa mieleen ehtorakenne sekä vahvistaa aliohjelmien ja merkkijonojen käyttämistä.

Run

- Tee ohjelma, joka ilmoittaa annetun lämpötilalukeman kahden muun lämpötilasteikon lukemana.
- Celsiuksista Fahrenheiteiksi ja Kelvineiksi
  - Fahrenheiteista Celsiuksiksi ja Kelvineiksi
  - Kelvineistä Celsiuksiksi ja Fahrenheiteiksi

- Tee aliohjelma, joka erottaa vastauksesta luvun ja yksikön erikseen ja muuntaa luvun lukumuotoon.
- Tee omat aliohjelmat jokaiselle muunnokselle. Muunnosaliohjelma kutsuu luvun ja yksikön erottavaa aliohjelmaa. Aliohjelma tulostaa muunnetun arvon yksiköineen.
- Pääohjelmassa käyttäjältä pyydetään muunnettava lämpötila yksiköineen.
- Ehtorakenteen vertailussa tutkitaan, millä yksiköllä käyttäjä on antanut lukeman ja sen perusteella valitaan kutsuttava muunnosaliohjelma.

## KUVAAJIA KONNAGRAFIIKALLA

### TEHTÄVÄ 5

Tavoitteena on palauttaa mieleen piirtäminen konnagrafiikalla.

Run

Tee ohjelma, joka piirtää koordinaatiston.

- Piirrä akselit. Origo kannattaa olla samassa pisteessä, kun piirtoalueen origo.
- Leimausta ja konnan muotoa hyödyntäen lisää kasvavaan suuntaan  $y$ -akselille ja  $x$ -akselille nuolet akselien päihin.
- Piilota konna lopuksi.
- Lisää oikealle osoittavan nuolen pään viereen  $x$  ja ylöspäin osoittavan nuolen vasemmalle puolelle  $y$ .
- Nopeuta animointia halutessasi.

#### KONNAGRAFIIKAN KOMENTOJA

tekstiä kuvaan `write("teksti", font=("fontti", "koko", "tyyli"))`

taustaväri `bgcolor('väri')`

animoinnin nopeus `speed(luku 1 - 10)`

### TEHTÄVÄ 6

Tavoitteena on hyödyntää aliohjelmiä kuvaajan piirtämisessä.

Run

Tee ohjelma, jossa käyttäjä antaa suoran yhtälön ratkaistussa muodossa ja ohjelma piirtää suoran kuvaajan.

Tee ohjelma, jossa

- aliohjelma piirtää koordinaatiston (*tehtävä 4*)
- aliohjelma pilkkoo käyttäjän antaman yhtälön erottaen kertoimen ja vakiotermin (*tehtävä 3*)
- aliohjelma, joka laskee annetulle suoralle  $y$ :n arvon annetulla  $x$ :n arvolla. Aliohjelma saa kutsussa käytettävän  $x$ :n arvon ja palauttaa sillä lasketun  $y$ :n arvon.

- aliohjelma, joka piirtää suoran koordinaation alueelta valittujen alku- ja loppupisteen välille.
  - o ohjelma kutsuu aliohjelmaa, jolla lasketaan  $y$ :n arvo annetulla  $x$ :n arvolla. Käytä  $x$ :n arvona lukua -250.
    - nosta kynä ylös
    - konna siirtyy pisteparin  $(x,y)$ -osoittamana pisteeseen muodostaen suoran vasemman puoleisen pään.
    - konna siirtyy pisteparin  $(x,y)$ -osoittamana pisteeseen muodostaen suoran oikean puoleisen pään.
    - Siirtyessä ensimmäisestä pisteestä toiseen pisteeseen konna piirtää suoran kuvaajan.
    - piilota konna
- pääohjelmassa
  - o kysytään piirrettävä suora tai kirjoitetaan haluttu suoraan ruokakuppiin.
  - o kutsutaan koordinaation piirtävä aliohjelma.
  - o kutsutaan aliohjelmaa, joka piirtää suoran.
  - o kirjoitetaan suoran yhtälö piirtoikkunaan

## TEHTÄVÄ 7

Tavoitteena on harjoitella matemaattisen sisällön kirjoittamista ohjelmakoodiksi.

Run

- Tee ohjelma, jossa käyttäjä antaa kaksi suoralla olevaa pistettä ja ohjelma laskee näiden tietojen perusteella kulmakertoimen, sitten selvitetään vakiotermin ja lopulta muodostetaan ja tulostetaan suoran yhtälö.
- Aliohjelmat pilkkovat annetusta pisteestä erikseen  $x$ -koordinaatin ja  $y$ -koordinaatin
  - Aliohjelma laskee kulmakertoimen (mieti, mitä parametrejä aliohjelma tarvitsee)
    - o lasketaan kulmakerroin (mieti, miten laskukaavan muodostat)
    - o Aliohjelma palauttaa kulmakertoimen
  - Aliohjelma laskee vakiotermin (mieti, mitä parametrejä aliohjelma tarvitsee)
    - o lasketaan vakiotermin (mieti, miten laskukaavan muodostat)
    - o Aliohjelma palauttaa vakiotermin
  - Pääohjelmassa
    - o kysytään käyttäjältä koordinaattipiste ja aliohjelmien avulla tallennetaan koordinaattipisteet muuttujiin  $x_1$  ja  $y_1$
    - o kysytään käyttäjältä koordinaattipiste ja aliohjelmien avulla tallennetaan koordinaattipisteet muuttujiin  $x_2$  ja  $y_2$
    - o kutsutaan kulmakertoimen ja vakiotermin määrittäviä aliohjelmiä

**TEHTÄVÄ 8**

Tavoitteena on palauttaa mieleen ehtorakenne ja vertailuehtojen käyttäminen

**Run**

Tee ohjelma, joka kertoo kahden suoralla olevan pisteen perusteella, onko se laskeva vai nouseva vai jommankumman akselin suuntainen suora.

- Laskeva ja nouseva suora selviää kulmakertoimen avulla
- Akselin suuntaisuus näkyy koordinaattipisteen  $x$ - tai  $y$ -koordinaatin kummassakin pisteessä toistuvana arvona.
- Ohjelma tulostaa tiedon suoran luonteesta.

## PÖYTÄTESTAUS

Tehtyä ohjelmaa kannattaa aina testata, jotta voi varmistua sen oikeellisuudesta. Yksi tapa testaamiseen on suorittaa ohjelmaa eri lähtöarvoilla ja tutkia, toimiiko ohjelma toivotunlaisesti.

Ohjelman toimintaa voi myös testata ilman tietokonetta toteutettavana pöytätestauksena. Pöytätestauksessa käytetään työvälineinä paperia ja kynää ja ohjelmaa edetään sen suoritusrjestyksessä muuttujien arvoja tarkastellen. Pöytätestauksen tavoite on varmistaa, että ohjelma toimii suunnitellulla tavalla.

Pöytätestauksen voi tehdä myös viemällä ohjelman visualisointiohjelmaan, jossa tutkitaan koodin toimintaa ja ohjelman tulostetta.

### Esimerkki 1

On tehty ohjelma, joka tulostaa  $x$ :n arvon. Jokaisella toistokierroksella  $x$ :n arvo kasvaa yhdellä.

```
1 x = 1
2 for i in range (3):
3     print x
4     x = x + 1
```

- Tutki ohjelman toimintaa ja muuttujien arvoja pöytätestaamalla. Tee pöytätestaus käyttämällä taulukkoa muuttujien seurantaan.
- Tutki ohjelman toimintaa ja muuttujien arvoja käyttäen visualisointipalvelua osoitteessa <http://www.pythontutor.com/visualize.html>. Valitse visualisoi nyt ja seuraa ohjelman etenemistä vaihe vaiheelta.

### Ratkaisu

- Tehdään taulukko, johon kootaan tiedot ohjelman etenemisen vaiheista (ohjelman askel ja koodin rivi) sekä ohjelmassa esiintyvien muuttujien arvot eri vaiheissa.

Askel	Koodin rivi	Luku $x$	Kierrosmuuttuja $i$
1	1	1	
2	2	1	0
3	3	tulostuu 1	0
4	4	$1+1 = 2$	0
5	2	2	1
6	3	tulostuu 2	1

7	4	3	1
8	2	3	2
9	3	tulostuu 3	2
10	4	4	2
11	2	4	3

b) Visualisointisovellus näyttää alla olevan kaltaiselta.

```

Python 2.7
→ 1 x = 1
   2 for i in range (3):
   3     print x
   4     x = x + 1

```

[Edit this code](#)

Print output (drag lower right corner to resize)

Frames      Objects

Koodi kopioidaan ohjelmointialustalta visualisointisovellukseen. Ohjelmaa edetään suoritusvaihe kerrallaan ja sovellus näyttää muuttujien arvoja ohjelman eri vaiheissa. Alla olevaan taulukkoon on koottu visualisointiohjelman tiedot.

Koodirivi	Globaali muuttuja	Tuloste
x = 1	x = 1	
for i in range (3):	x = 1, i = 0	
print x	x = 2, i = 0	1
x = x + 1	x = 2, i = 0	
for i in range (3):	x = 2, i = 1	
print x	x = 2, i = 1	2
x = x + 1	x = 3, i = 1	
for i in range (3):	x = 3, i = 2	
print x	x = 3, i = 2	3
x = x + 1	x = 4, i = 2	
for i in range (3):	x = 4, i = 3	
Ohjelma päättyi.		

Visualisointiohjelmaa voi käyttää mm. selvittämään koodin toiminnassa oleva virhe. Visualisointiohjelmalla ei voi tutkia kirjastoja sisältäviä koodeja, sillä alusta ei tue kirjastoja.

## TEHTÄVÄ 9

Tavoitteena on harjoitella pöytätestausta ja visualisointiohjelman käyttöä yksinkertaisen ohjelman avulla.

Run

Tee pöytätestaus oheiselle koodille. Ohjelma tulostaa muuttujien  $a$  ja  $b$  summan  $a$ :n ja  $b$ :n arvojen muuttamisen jälkeen.

```
a = 2
b = 3
print a + b
a = a - 2
b = b + 1
print a + b
a = a - 2
b = b + 1
print a + b
```



## TEHTÄVÄ 10

Tavoitteena on vertailla pöytätestausta ja visualisointiohjelmaa.

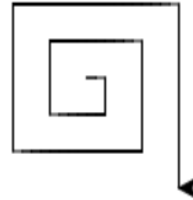
Run

```
from turtle import*
alue = Screen()
konna = Turtle()

a = 10

for i in range(10):
    konna.forward(a)
    konna.right(90)
    a = a + i
```

Tutki alla olevaa koodia. Koodin pitäisi muodostaa oheinen kuvio.



- Selitä ohjelman toimintaa eli kerro lyhyesti, mitä milläkin ohjelman rivillä tapahtuu?
- Tee ohjelmalle pöytätestaus. Mitä havaitset?
- Miten muuttaisit koodia?