

# Python-ohjelmointi

# Harjoitus 3

## TAVOITTEET

- Kerrataan vertailuoperaattorien käyttö sekä jos-ehdorakenne.
- Opitaan piirtämään kilpikonnagrafiikan avulla ja lisäämään valmiita kirjastoja.
- Opitaan laskemaan kääntymiskulmia ja käyttämään kilpikonnagrafiikan komentoja.

## Matematiikan sisällöt

Tehtävät tukevat harjoittelua koordinaatistossa ja geometrian peruskäsitteiden ymmärtämistä.

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

## KERTAUSTA HARJOITUKSESTA 2

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen vertailuoperaattorien käyttö ja jos-ehdorakenne.

Run

Tee ohjelma, jolla voit verrata käytettävissä olevaa rahamäärää konserttilipun hintaan. Jos käytettävissä olevat rahat riittävät, ohjelma tulostaa lauseen ”Voit ostaa lipun!”. Jos rahaa ei ole riittävästi, ohjelma tulostaa lauseen ”Valitettavasti rahasi eivät riitä.”

- Määritä muuttujat *rahaa\_kaytettavissa* ja *lipun\_hinta* ja niille arvot.
- Tee seuraavat ehdorakenteet:
  - Jos *rahaa\_kaytettavissa* on suurempi kuin *lipun\_hinta*, ohjelma tulostaa lauseen ”Voit ostaa lipun!”
  - Jos *rahaa\_kaytettavissa* on pienempi kuin *lipun\_hinta*, ohjelma tulostaa lauseen ”Valitettavasti rahasi eivät riitä.”
- Testaa ohjelmaa eri rahamäärillä.
- Toimiiko ohjelma, jos rahaa on yhtä paljon kuin lippu maksaa? Jos ei, muokkaa ohjelmaa niin, että se toimii tässäkin tapauksessa.

## PIIRTÄMINEN PYTHONILLA

Python-kielessä piirtämiseen käytetään niin sanottua Turtle-ohjelmointia. Piirroskuvat tulevat ohjelmointialustan valkoiselle alueelle eli kankaalle.

Python-kielelle on olemassa valmis kokoelma eli kirjasto konnaa ohjaavia komentoja. Jotta Turtle-komennot ovat käytettävissä, kyseinen kirjasto pitää tuoda omaan ohjelmaan. Ohjelman alussa luodaan myös piirtoalue sekä annetaan konnalle nimi.

Kilpikonnagrafiikka käyttävän ohjelman alkuun kirjoitetaan ainakin seuraavat rivit.

```
from turtle import * # Tuodaan turtle-komennot sisältävä kirjasto.
alue = Screen()     # Määritetään piirtoalue.
allu = Turtle()     # Määritetään konnalle nimeksi Allu.
allu.reset()        # Tyhjentää piirtoalueen ja palauttaa konnan origoon oletusasetuksilla.
```

## TEHTÄVÄ 2

Tavoitteena on oppia piirtämään kilpikonnagrafiikan avulla.

Run

Kirjoita alla oleva ohjelma ja suorita se.

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

konna.forward(50)
```

- Mitä ohjelma tekee?
- Vaihda luvun 50 tilalle jokin toinen luku, joka on pienempi kuin 200. Mihin luku vaikuttaa?
- Lisää ohjelmaan rivi `konna.right(90)` ja suorita ohjelma. Mitä rivi tekee?
- Lisää ohjelmaan rivi, jolla `konna` piirtää alaspäin 100 yksikköä pitkän viivan.

## KONNAN LIKUTTAMINEN JA KÄÄNTYMINEN

Konnan toimintoja ohjataan käskyjen avulla. Seuraavassa taulukossa on esimerkkejä komennosta, joita on kirjoitettu konnalle nimeltään Kilppari.

Komennot aloitetaan aina konnan nimellä ja pisteen jälkeen tulee haluttu komento. Mikäli komento vaatii, komennon perään sulkeisiin kirjoitetaan tarvittavat määritteet.

### Konnan komennon muoto

Konnan komento kirjoitetaan seuraavaan muotoon:

konnaNimi.komento(*määrite*)

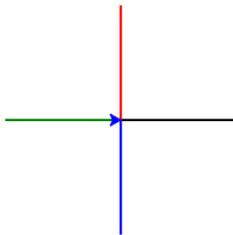
Komento	Toiminto
kilppari.forward(50)	Kilppari menee eteenpäin sijaintipaikastaan 50 yksikköä.
kilppari.backward(50)	Kilppari peruuttaa sijaintipaikastaan 50 yksikköä.
kilppari.right(90)	Kilppari kääntyy sijaintipaikassaan 90° oikealle.
kilppari.left(90)	Kilppari kääntyy sijaintipaikassaan 90° vasemmalle.
kilppari.color('red')	Kilpparin piirtämän viivan väri muuttuu punaiseksi.
kilppari.pensize(10)	Kilpparin piirtämän viivan paksuus muuttuu 10 yksikön paksuuteen.

### TEHTÄVÄ 3

Tavoitteena on oppia konnan liikuttamisen peruskomennot, suunnan määrääminen ja viivan värin vaihtaminen.

Run

Tee ohjelma, jolla saat piirrettyä alla olevan kuvion. Tuo ohjelmaan Turtle-kirjasto, ja määritä piirtoalue.

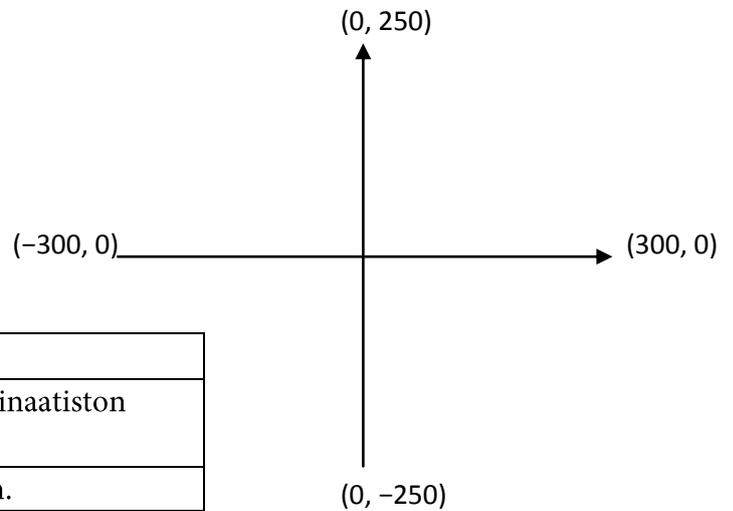


- Anna konnalle nimi.
- Kirjoita koodi, jolla konna piirtää 100 yksikköä pitkän mustan viivan ja peruuttaa takaisin 100 yksikköä.
- Kirjoita koodi, jolla konna kääntyy 90° vasempaan, piirtää 100 yksikköä pitkän punaisen viivan ja peruuttaa takaisin. Värit pitää kirjoittaa englanniksi.
- Jatka ohjelmaa, jotta saat kuvion piirrettyä.
- Muuta ohjelmaa siten, että saat jokaisen viivan piirrettyä eri paksuudella.
- Kokeile vaihtaa käytettyjä värejä.

## PIIRTOALUE

Piirtoalue on koordinaatisto. Konna aloittaa piirtämisen origosta eli pisteestä (0, 0) ellei ohjelmassa toisin määrätä.

Viereisestä kuvasta näkee koordinaatiston koon.  
Kuvioita voidaan piirtää liikuttamalla konnaa koordinaatiston pisteestä pisteeseen.



Komento	Toiminto
kilppari.goto(10, 20)	Kilppari menee koordinaatiston pisteeseen (10, 20).
kilppari.home()	Kilppari palaa origoon.
kilppari.penup()	Kilppari ei piirrä viivaa liikkueessaan.
kilppari.pendown()	Kilppari piirtää viivan liikkueessaan.

## TEHTÄVÄ 4

Tavoitteena on täydentää kuvio.

Run

- Kirjoita oheinen koodi ja testaa, millainen kuvion alku muodostuu.
- Miksi alussa on komento lopettaa viivan piirtäminen?
- Täydennä kuvio siten, että siitä muodostuu talo.
- Yritä piirtää taloon myös ovi ja ikkuna.

```
from turtle import *
alue = Screen()
allu = Turtle()
allu.reset()

allu.penup()
allu.goto(0, -80)
allu.pendown()

allu.goto(-100, -80)
allu.goto(-100, 100)
allu.goto(0, 100)
allu.goto(-100, 100)
allu.goto(0, 200)
allu.goto(-50, 150)
allu.goto(-50, 200)
allu.goto(-30, 200)
allu.goto(-30, 170)
allu.goto(0, 200)
```

## TEHTÄVÄ 5

Tavoitteena on piirtää kuvio liikuttamalla konnaa koordinaatiston pisteestä toiseen.

Run

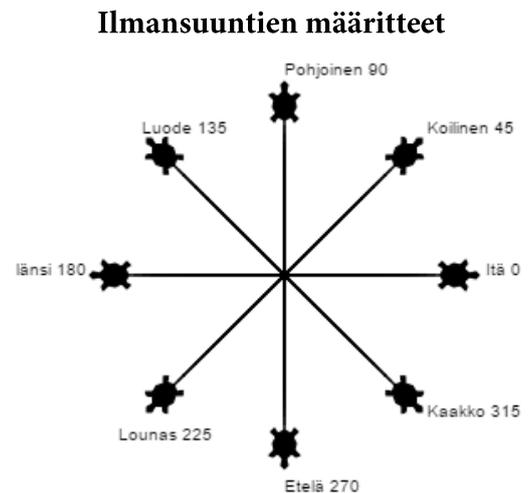
- Tuo Turtle-komennot ohjelmaasi, määritä piirtoalue, ja anna konnalle nimi.
- Nosta kynä ylös ja siirrä konna pisteeseen (-15, 45).
- Laske kynä alas ja liikuta konna pisteisiin (0, 15), (30, 15), (0, 0), (15, -30), (-15, -15), (-45, -30), (-30, 0), (-60, 15), (-30, 15) ja (-15, 45).
- Mikä kuvio muodostuu?

## ETENEMISSUUNTA, KÄÄNTYMISKULMAT JA MUODON VAIHTAMINEN

Hahmolle voidaan määrittää eri muotoja komennolla **shape()**. Muodoksi voidaan valita jokin seuraavista: 'arrow', 'circle', 'classic', 'square', 'triangle' tai 'turtle'.

Konnan etenemissuunta voidaan määrittää ilmansuuntien avulla. Tällöin käytetään joko komentoa **seth()** tai **setheading()**.

Komento	Toiminto
kilppari.shape('triangle')	Kilpparin muodoksi määritetään kolmio.
kilppari.seth(0)	Kilpparin suunta on itään eli 0 astetta.
kilppari.seth(90)	Kilpparin suunta on pohjoiseen eli 90 astetta.
kilppari.seth(180)	Kilpparin suunta on länteen eli 180 astetta.
kilppari.seth(270)	Kilpparin suunta on etelään eli 270 astetta.
kilppari.stamp()	Kilppari leimaa sijaintinsa.



### Esimerkki 1

Tee ohjelma, jossa konnan etenemissuunta määritellään ilmansuuntien mukaan ja jossa konna leimaa lopuksi sijaintipaikkansa etenemissuunnan määrittämisen mukaan. Etenemissuuntien määrittämisen välillä konna liikkuu 50 yksikköä määritettyyn etenemissuuntaan.

### Ratkaisu

Ohjelmaan alkuun tuodaan Turtle-kirjasto, määritetään piirtoalue ja annetaan konnalle nimi. Valitaan konnan muodoksi nuoli ja määritetään konnan etenemissuunta ilmansuuntien avulla.

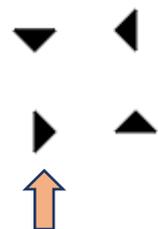
### KOODI

```
from turtle import * # Tuodaan turtle-komennot sisältävä kirjasto.

alue = Screen() # Määritetään piirtoalue.
allu = Turtle() # Määritetään konnalle nimeksi Allu.
allu.reset() # Tyhjennetään piirtoalue.

allu.shape('arrow')
allu.penup()
allu.stamp() # Leimataan konnan sijainti muodolla arrow eli nuoli.
allu.seth(0) # Määritetään etenemissuunnaksi itä eli 0.
allu.forward(50)
allu.seth(90) # Määritetään etenemissuunnaksi pohjoinen eli 90.
allu.stamp()
allu.forward(50)
allu.seth(180) # Määritetään etenemissuunnaksi länsi eli 180.
allu.stamp()
allu.forward(50)
allu.seth(270) # Määritetään etenemissuunnaksi etelä eli 270.
allu.stamp()
```

### PIIRROS



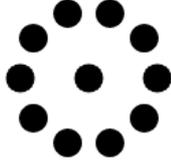
Kuvion aloituspiste

## TEHTÄVÄ 6

Tavoitteena on harjoitella leimaustoimintoa ja piirtää kuvio.

Run

Tee ohjelma, jolla voit piirtää alla olevan kuvion.



- Määritä konnan muodoksi ympyrä eli 'circle'.
- Konna ei piirrä viivaa eli nosta kynä ylös.
- Liikuta konnaa eteenpäin esimerkiksi 75 yksikköä ja tee leima. Peruuta konna takaisin 75 yksikköä.
- Täyden kulman asteluku on 360°. Kun konna tekee 10 leimaa, se jakaa ympyrän kymmeneen yhtä suureen osaan. Kuinka monta astetta konna kääntyy peräkkäisten leimausten välissä? Konna voi kääntyä joko vasempaan tai oikeaan.
- Jatka koodia, kunnes kuvio tulee valmiiksi.
- Viimeisen peruutuksen jälkeen tee leima kotipisteeseen eli keskipisteeseen.
- Miten ohjelmaa pitäisi muuttaa, että saisit kellotaulua vastaavat leimaukset eli 12 leimaa?

## TEHTÄVÄ 7

Tavoitteena on harjoitella hahmon muodon vaihtamista ja leimaamista.

Run

Tee ohjelma, jossa konna käy tekemässä leiman jokaisessa ilmansuunnassa 100 yksikön etäisyydellä kotipisteestä. Konna leimaa kotipisteen aina, kun siellä käy, mutta ei piirrä viivaa liikkeistään. Pääilmansuunnassa konnan muoto on 'turtle', väliilmansuunnassa konnan muoto on 'arrow'.

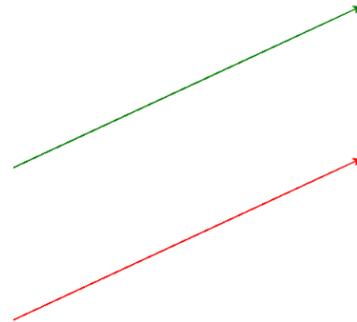
- Määritä konnan muoto `shape()`-komennolla, ja anna komento kynän nostamiseksi.
- Määritä konnan etenemissuunnat ilmansuuntien avulla. Aloita idästä (0).
- Liikuta konnaa `forward()`-komennolla 100 yksikköä, ja tee lopussa leima.
- Palauta konna kotiin `home()`-komennolla, ja tee kotipisteeseen leima.
- Määritä uusi etenemissuunta, sille etenemismatka ja leimaaminen. Palauta konna kotiin.
- Täydennä ohjelma niin, että konna käy leimaamassa jokaisen ilmansuunnan ja tekee leiman joka kerta myös kotipisteeseen.

## USEITA KONNIA

Ohjelmassa voi olla mukana useita konnia. Jokaiselle konnalle annetaan oma nimi. Komenttoon liitetty konnan nimi kertoo, mitä konna annettu komento ohjaa. Jokainen konna lähtee liikkeelle origosta.

### Esimerkki 2

Tee ohjelma, jonka avulla voit piirtää viereisen kuvion.



### Ratkaisu

Ohjelmaan tarvitaan kaksi konnaa, jotka lähtevät eri pisteitä mutta etenevät samaan suuntaan yhtä pitkän matkan.

Valitaan toisen konnan lähtöpisteeksi  $(-250, -150)$ . Jos suorien välinen etäisyys on 200 yksikköä, toisen konnan lähtöpisteeksi tulee  $(-250, 50)$ . Käännetään molemmat konnat samaan suuntaan ja määritetään matka, jonka ne kulkevat. Näin saadaan syntymään kuvio, jossa on yhdensuuntaiset suorat.

Ne arvot, jotka koskevat molempia konnia, ilmoitetaan muuttujien avulla.

```
from turtle import *
x = -250           # Määritetään x-koordinaatti, joka on molemmille konnille yhteinen.
kulma = 25        # Määritetään kulma, johon molemmat konnat kääntyvät.
matka = 500       # Määritetään matka, jonka molemmat konnat etenevät.

alue = Screen()
ansa = Turtle()   # Nimetään ensimmäinen konna.
onni = Turtle()   # Nimetään toinen konna.
ansa.color('red') # Määritetään molemmille konnille viivan väri.
onni.color('green')
ansa.penup()
ansa.goto(x, -150) # Ansa siirtyy valittuun aloituspisteeseen.
onni.penup()
onni.goto(x, 50)  # Onni siirtyy valittuun aloituspisteeseen.
ansa.pendown()
onni.pendown()
ansa.left(kulma)  # Konnat kääntyvät valitun kulman suuntaan.
onni.left(kulma)
ansa.forward(matka) # Konnat etenevät valitun matkan.
onni.forward(matka)
```

## TEHTÄVÄ 8

Tavoitteena on harjoitella useamman konnan käyttämistä ohjelmassa.

Run

Tee ohjelma, jolla voit piirtää kolme yhdensuuntaista suoraa. Ensimmäinen konna lähtee pisteestä (280, 120). Tee suorista mahdollisimman pitkät.

- Päätä, millä etäisyydellä suorat ovat toisistaan, ja valitse kahden muun konnan lähtöpisteet. Jos kaikkien lähtöpisteiden  $x$ -koordinaatit ovat samat, käytä muuttujaa apuna.
- Määritä muuttuja, jonka avulla ilmaiset konnien kulkusuunnan.
- Määritä muuttuja, jonka avulla ilmaiset konnien etenemän matkan.
- Määritä aloituspisteet ja konnien liikkuminen.
- Testaa ohjelma.

## TEHTÄVÄ 9

Tavoitteena on harjoitella konnan liikuttamista ja liikkumissuunnan määrittämistä.

Run

Tee ohjelma, jossa konnat piirtävät toisiaan leikkaavat suorat.

- Ensimmäinen konna lähtee pisteestä  $(-280, 200)$ .
- Toinen konna lähtee pisteestä  $(-50, -200)$ .
- Määritä konnille kulkusuunnat ja etenemismatkat siten, että suorat leikkaavat toisensa.
- Testaa ohjelma.
- Muuta kulkusuuntia siten, että suorat leikkaavat toisensa 90 asteen kulmassa.

## TEHTÄVÄ 10

Tavoitteena tutustua ympyrä-komentoon sekä oppia täyttämään muoto.

Run

- Piirrä ympyrä komennolla `circle()`. Sulkuihin kirjoitetaan ympyrän säde. Määritä säteeksi 50 yksikköä. Testaa ohjelmaa.
- Kun ympyrä halutaan värittää, pitää ennen ympyrän piirtämistä lisätä komento `begin_fill()` ja ympyrän piirtämisen jälkeen `end_fill()`. Testaa ohjelmaa.
- Jos reunaviiva halutaan piirtää eri värillä kuin täyttöväri, värikomentoon `color` lisätään ensin reunaviivan väri ja sitten täyttöväri. Esim. `color('green', 'red')`.
- Täydennä ohjelma niin, että ympyrän ympärille piirtyy neliö, johon ympyrä juuri ja juuri mahtuu. Ympyrä voi olla väritetty.

