

# Python-ohjelmointi

# Harjoitus 4

## TAVOITTEET

- Kerrataan kilpikonnagrafiikalla piirtäminen
- Kerrataan muuttujien käyttäminen.
- Opitaan for-silmukkarakenne

## Matematiikan sisällöt

Tehtävät tukevat kulmien nimitysten ja laskemisten kertaamista, monikulmioiden luokittelun ja ominaisuuksien harjoittamista sekä nelikulmioiden piirin ja pinta-alan laskemista.

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

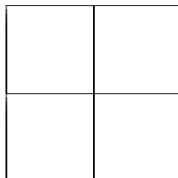
## KERTAUSTA HARJOITUKSESTA 3

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen piirtämiseen, kääntymiseen ja suunnan valintaan liittyvät komennot.

Run

Tee ohjelma, jolla saat piirrettyä alla olevan kuvion.  
Voit valita itse neliön sivun pituuden.



### TEHTÄVÄ 2

Tavoitteena on palauttaa mieleen useamman konnan komentaminen.

Run

Tee ohjelma, jossa kaksi konnaa piirtää oman suorakulmion, kumpikin omalla värillään. Suorakulmioiden etäisyys toisistaan on 100 yksikköä.

## MONIKULMIOIDEN PIIRTÄMINEN

Monikulmioiden piirtämisessä konna liikkuu joko eteenpäin tai tekee käännöksiä. Kääntymiskulma määritetään konnan etenemissuunnan suhteen.

### Esimerkki 1

Tee ohjelma, jolla voit piirtää suunnikkaan. Tee ohjelma siten, että sitä voi helposti hyödyntää erilaisten suunnikkaiden piirtämiseen. Tutki ohjelman toimintaa suunnikkaalla, jonka sivujen pituudet ovat 85 ja 46 yksikköä. Suunnikkaan terävä kulma on  $37^\circ$ .

### Ratkaisu

Tehdään ohjelmaan muuttujat *sivu1*, *sivu2* ja *kulma*, jotta ohjelmaa voi helposti testata erilaisilla suunnikkailla.

Hahmo tekee käännöksen etenemissuuntansa suhteen. Komento `left(37)` tuottaa suunnikkaan tylpän kulman, jonka suuruus on  $(180^\circ - 37^\circ) = 143^\circ$ . Jotta saadaan määritettyä oikea kääntymiskulma, haluttu kulma täytyy vähentää oikokulmasta eli 180 asteesta.



### KOODI

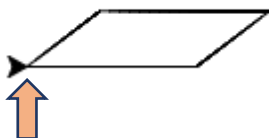
```
from turtle import*

alue = Screen()
ukko = Turtle()
ukko.reset()          # Tyhjentää piirtoalueen.

sivu1 = 85            # Määritetään muuttujat ja niille arvot.
sivu2 = 46            # Arvoja vaihtamalla ohjelma tekee erilaisen suunnikkaan.
kulma = 37

ukko.forward(sivu1)
ukko.left(kulma)     # Tämä rivi muodostaa suunnikkaan tylpän kulman.
ukko.forward(sivu2)
ukko.left(180-kulma) # Tämä rivi muodostaa suunnikkaan terävän kulman.
ukko.forward(sivu1)
ukko.left(kulma)
ukko.forward(sivu2)
ukko.left(180-kulma)
```

### PIIRROS



Kuvion aloituspiste

Ohjelma piirtää erilaisen suunnikkaan muuttujien arvoja muuttamalla.

### TEHTÄVÄ 3

Tavoitteena on harjoitella kääntymiskulmien laskemista.

Run

Tee ohjelma, jolla voit piirtää tasasivuisen kolmion, jonka sivun pituus on 200 yksikköä.

### TEHTÄVÄ 4

Tavoitteena on harjoitella kuvion täydentämistä.

Run

Tee ohjelma, jolla voit piirtää puolisuunnikkaan.

- Määritä koordinaatiston piste, josta aloitat piirtämisen. Oletuksena on, että kuvion piirtäminen alkaa pisteestä (0, 0).
- Liikuta konnaa niin, että vaakasuuntaisen sivun pituudeksi saadaan 45 yksikköä. Sivun jälkeen konnan täytyy kääntyä 90 astetta.
- Liikuta konnaa niin, että pystysuuntaisen sivun pituudeksi tulee 30 yksikköä. Sivun jälkeen konnan täytyy kääntyä 90 astetta.
- Liikuta konnaa niin, että pidemmän vaakasuuntaisen sivun pituudeksi tulee 65 yksikköä.
- Liikuta konna lähtöpisteeseen, jolloin kuvion täydentyy puolisuunnikkaaksi.

### TEHTÄVÄ 5

Tavoitteena on piirtää kuvio ja laskea sen piiri ja pinta-ala.

Run

Tee ohjelma, joka piirtää suorakulmion. Suorakulmion sivujen pituudet kysytään käyttäjältä. Piirtämisen jälkeen ohjelma laskee ja tulostaa kuvion piirin sekä pinta-alan. Sivujen pituuksille ei anneta tässä tehtävässä pituusyksikköä eikä vastaustulosteisiin tule yksiköitä.

## SILMUKKARAKENNE FOR

Jos tiettyä komentoa halutaan toistaa useamman kerran, se voidaan sijoittaa silmukkarakenteen sisään. Silmukkarakennetta toistetaan ohjelmaan kirjoitettujen toistojen verran.

Silmukkarakenteeseen kuuluvat komennot sisennetään.

**For**-silmukkarakenteessa tarvitaan laskuri, joka laskee toistokertojen lukumäärän. Tätä kutsutaan silmukkamuuttujaksi. Silmukkamuuttujaa voi nimetä myös jollakin kirjaimella. Toistojen lukumäärä ilmoitetaan komennolla **range()**. Kierrosten laskeminen alkaa aina nolasta ja päättyy yhtä ennen viimeistä arvoa, ellei range-komennossa muuta määrätä.

#### Silmukka eli toistorakenne for

Luku, josta laskeminen alkaa.      Luku, jota ennen laskeminen päättyy.      Siirtymä lukujen välillä.

```
for laskuri in range(alku, loppu, askel):
    Toistettavaksi haluttavat tapahtumat.
```

**Esimerkki 2**

Tee ohjelma, jolla voit tutkia, mitä range()-komento tekee, kun sulkeiden sisään kirjoitetaan

- a) 3
- b) 1, 10
- c) 0, 11, 2.

**Ratkaisu**

Ohjelma	Tuloste	Selitys
<pre>for laskuri in range (3):     print laskuri</pre>	0 1 2	Silmukkaa toistetaan kolme kertaa.
<pre>for laskuri in range (1, 10):     print laskuri</pre>	1 2 3 4 5 6 7 8 9	Silmukkamuuttuja alkaa luvusta 1 ja päättyy yhden luvun ennen lukua 10.
<pre>for laskuri in range (0, 11, 2):     print laskuri</pre>	0 2 4 6 8 10	Silmukkamuuttuja alkaa luvusta 0 ja päättyy yhden luvun ennen lukua 11. Siirtymä lukujen välillä on 2 eli silmukassa hypätään parittomien lukujen yli.

### Esimerkki 3

Tee ohjelma, jolla voit piirtää suunnikkaan for-silmukkaa hyödyntäen. Suunnikkaan sivujen pituudet sekä yksi sivujen välinen kulma pyydetään käyttäjältä.

### Ratkaisu

Suunnikkaan vastakkaiset kulmat ovat yhtä suuret ja vastakkaiset sivut yhtä pitkät ja yhdensuuntaiset. Tehdään ohjelma, joka piirtää vaakasuoran sivun, kääntyy, piirtää vinon sivun ja kääntyy. Suunnikas saadaan piirrettyä toistamalla edellä kuvatut vaiheet. Suunnikkaan piirtämiseksi tarvitaan siis kaksi silmukkakierrosta.

### KOODI

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

# Kysytään sivujen pituudet ja muutetaan samantien vastaus kokonaisluvuksi.
# Kokonaisluvuksi muuntaminen voidaan tehdä myös myöhemmin ohjelmassa.
sivu1 = int(input(u'Anna suunnikkaan sivujen pituudet kokonaislukuna. Anna
ensimmäinen suunnikkaan sivuista.'))
sivu2 = int(input('Anna toinen suunnikkaan sivuista.))

# Kysytään terävän kulman suuruus ja muutetaan saatu vastaus heti
kokonaisluvuksi.
kulma = int(input(u'Anna sivujen välinen terävä kulma.))

# Tehdään silmukkarakenne suunnikkaan piirtämiseksi.
for laskuri in range(2):
    ukko.forward(sivu1)
    ukko.left(180-kulma) # Tekee tylpän kulman.
    ukko.forward(sivu2)
    ukko.left(kulma) # Tekee terävän kulman.
```

### PIIROS

Esimerkkituloste suunnikkaasta, jonka sivut ovat 50 ja 60 yksikköä ja sivujen välinen terävä kulma on 30°.



### TEHTÄVÄ 6

Tavoitteena on harjoitella for-silmukkarakennetta.

Run

Tee ohjelma, joka käyttää for-silmukkarakennetta ja piirtää neliön, jonka sivun pituus on 200 yksikköä.

## TEHTÄVÄ 7

Tavoitteena on hyödyntää for-silmukkarakennetta.

Run

Tee ohjelma, joka hyödyntää for-silmukkarakennetta ja piirtää halutun säännöllisen monikulmion. Piirrä ensin tasasivuinen kolmio.

- Pyydä käyttäjältä sivun pituus.
- Pyydä käyttäjältä tieto, kuinka monta kärkipistettä monikulmiossa on. Esim. kolmiossa on kolme kärkipistettä ja neliössä neljä kärkipistettä.
- Muista muuntaa käyttäjän antamat vastaukset kokonaislukumuuttujiksi (int).
- Tee for-silmukka, jota toistetaan kärkipisteiden lukumäärän verran.
- Määritä eteenpäin liikuttava matka.
- Määritä kääntymiskulma. Se saadaan jakamalla 360° kärkipisteiden määrällä.
- Testaa ohjelma.
- Kokeile ohjelman toiminta viisikulmiolla ja kahdeksankulmiolla.

## TEHTÄVÄ 8

Tavoitteena on harjoitella for-silmukan silmukkamuuttujan hyödyntämistä tulosteissa.

Run

Tee ohjelma, joka tulostaa kahden kertotaulun vastaukset.

- Silmukkamuuttujaa voi käyttää laskulausekkeissa kuten mitä tahansa muuttujaa.
- Ohjelma tulostaa luvun 2 ja laskurimuuttujan tulon jokaisella silmukkakierroksella.

## SISÄKKÄISET SILMUKAT

Silmukkarakenteita voi kirjoittaa sisäkkäin. Ulompi silmukkarakenne etenee vasta sitten, kun sisempää silmukkarakennetta on toistettu haluttu määrä. Molempiin silmukoihin tulee oma silmukkamuuttuja.

### Sisäkkäiset silmukat

```
for laskuri1 in range(alku, loppu, askel):
```

```
    for laskuri2 in range(alku, loppu, askel):
```

```
        Toistettavaksi haluttavat tapahtumat.
```

**Esimerkki 4**

Tee ohjelma, joka laskee ja tulostaa oheisen tulosteen mukaiset laskut vastauksineen.

**Ratkaisu**

Ohjelman tulee tulostaa kokonaislukujen 1–4 tulot lukujen 5 ja 10 kanssa.

Tarvitaan kaksi silmukkarakennetta. Ulompi silmukkarakenne muodostaa luvut väliltä 1–4. Määritetään uloimman silmukan silmukkamuuttujaksi *laskuri1*.

Sisempi silmukkarakenne huolehtii viidellä jaollisista luvuista välillä 1–10. Määritetään sisemmän silmukkarakenteen silmukkamuuttujaksi *laskuri2*.

Tulosteessa tulostetaan näkyviin kertolasku ja sen vastaus.

**KOODI**

```
for laskuri1 in range(1, 5):
    for laskuri2 in range(5, 11, 5):
        print str(laskuri1) + '*' + str(laskuri2) + '=' + str(laskuri1 * laskuri2)
```

```
1 * 5 = 5
1 * 10 = 10
2 * 5 = 10
2 * 10 = 20
3 * 5 = 15
3 * 10 = 30
4 * 5 = 20
4 * 10 = 40
```

**TEHTÄVÄ 9**

Tavoitteena on harjoitella sisäkkäisten for-silmukkarakenteiden käyttämistä.

Run

Tee ohjelma, jolla voit tulostaa kahden luvun summan, kun molemmat luvut ovat väliltä 1–5.

- Määritä silmukkamuuttuja uloimmalle silmukalle ja sille alue.
- Määritä silmukkamuuttuja sisemmälle silmukalle ja sille alue.
- Tulosta summa.

**TEHTÄVÄ 10**

Tavoitteena on harjoitella sisäkkäisten for-silmukoiden käyttämistä.

Run

Tee ohjelma, jolla voit tulostaa kertotaulut 1–10. Tulosteessa täytyy näkyä, mitkä luvut kerrotaan ja mikä on niiden vastaus.

Esimerkiksi tulosteessa näkyy  $7 * 8 = 56$ .

## LISÄTEHTÄVIÄ

### LISÄTEHTÄVÄ 1

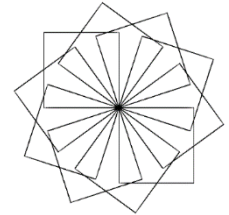
Tavoitteena on syventää silmukkarakenteen käyttämistä.

- Tee ohjelma, joka laskee ja tulostaa neliön piirin ja pinta-alan sivun pituuksille 1–10.

### LISÄTEHTÄVÄ 2

Tavoitteena on hyödyntää sisäkkäisiä silmukoita ja tehdä kuviotaidetta.

- Tee ohjelma, joka tulostaa viereisen kuvion.
- Kuvion perusmuoto on neliö, jonka sivun pituus on 150 yksikköä. Neliöitä on kuviossa kymmenen. Kun yksi neliö on valmis, hahmo kääntyy  $36^\circ$  ja piirtää uudelleen neliön.



Vihje: hahmon kääntyminen on osa ulompaa silmukkarakennetta.

### LISÄTEHTÄVÄ 3

Tavoitteena on tehdä yksikönmuunnoksissa hyödynnettävä muunnin.

- Tee ohjelma, joka muuntaa ja tulostaa metreinä annetun pituuden eri pituusyksiköissä.
- Tee vastaava muunninohjelma myös pinta-alayksiköille siten, että neliömetreinä annettu pinta-ala ilmoitetaan muissa pinta-alayksiköissä.