

Python-ohjelmointi

Harjoitus 5

TAVOITTEET

- Kerrataan silmukkarakenteen käyttäminen.
- Kerrataan jos-ehtorakenteen käyttäminen.
- Opitaan if-else- ja if-elif-else-ehtorakenteet.

Matematiikan sisällöt

Tehtävät tukevat luku- ja kuviojonojen opettelua sekä muuttujakirjaimen tuomista osaksi laskutoimituksia. Tehtävät tukevat lisäksi yleisesti kirjainlaskennan harjoittelua.

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

KERTAUSTA HARJOITUKSESTA 4

TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen for-silmukkarakenne ja range()-komento.

Run

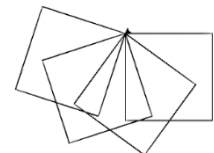
Tee ohjelma, joka tulostaa for-silmukkarakennetta hyödyntäen luvut 1–20.

TEHTÄVÄ 2

Tavoitteena on palauttaa mieleen sisäkkäiset silmukkarakenteet.

Run

Tee ohjelma, joka piirtää sisäistä silmukkarakennetta hyödyntäen oheisen kuvion.



- Kuviossa on neljä kappaletta neliöitä eli ulompaa silmukkarakennetta toistetaan neljä kertaa.
- Sisempi silmukkarakenne piirtää neliön. Neliön sivun pituuden voi päättää itse tai se voidaan kysyä käyttäjältä.
- Kuvioden välissä tehdään 36 asteen käännös. Mieti tämän komentorivin sisennys, jotta se kuuluu oikeaan silmukkarakenteeseen.

LUKU- JA KUVIOJONOT

Esimerkki 1

Tee ohjelma, joka tulostaa lukujonon 19, 17, 15, 13, 11, 9, 7, 5, 3, 1.

Ratkaisu

Lukujono muodostuu parittomista luvuista välillä 1–19 laskevassa järjestyksessä. Range()-komentossa alue määritetään ensimmäisestä luvusta eli luvusta 19 alkaen. Lukualue päättyy yhden luvun halutun päättymisluvun jälkeen. Koska viimeiseksi halutaan tulostaa luku 1, päättyy alue lukuun nolla. Parittomat luvut saadaan laskevassa järjestyksessä, kun määritetään askeleeksi -2. Laskuria kuvaamaan käytetään muuttujakirjainta *n*.

KOODI

```
for n in range (19, 0, -2):  
    print n
```

TULOSTE

```
19  
17  
15  
13  
11  
9  
7  
5  
3  
1
```

Jos lukujono halutaan tulostaa yhdelle riville, tulostusrivi kirjoitetaan seuraavasti:

```
print n,
```

Numerot tulostuvat tällöin yhdelle riville välilyönneillä erotettuina.

Jos lukujono halutaan tulostaa yhdelle riville luvut pilkulla erotettuna, kirjoitetaan tulostusrivi seuraavasti:

```
print (str(n)+' ', ),
```

Numerot tulostuvat yhdelle riville välilyönneillä erotettuna ja myös viimeisen luvun perän tulee pilkku.

Esimerkki 2

Tee ohjelma, joka tulostaa kuviojonon jäseniä käyttäjän haluaman määrän.

Ratkaisu

Kuviojono saadaan tulostettua leimaustoiminnon avulla.

Käyttäjän ilmoittamaa kuviojonon jäsenten määrää käytetään silmukkamuuttujan alueen päättymislukuun. Käyttäjältä pyydetty lukumäärä, muuttuja *lkm*, pitää muuttaa kokonaisluvuksi, jotta sitä voidaan käyttää `range()`-komennossa.

Ulommalla silmukkarakenteella määrätään kuviodien määrä. Silmukkamuuttujan lopetusluvuksi käytetään käyttäjän ilmoittamaa määrää lisättynä yhdellä. Ulomman silmukan silmukkamuuttujaa merkitään muuttujakirjaimella *n*. Kuviojonon jäsenten välinen etäisyys määritetään ulommassa silmukassa.

Sisempi silmukkarakenne toteuttaa halutun kuviojonon muodostumisen. Käytetään kuviojonon tekemiseen leimaustoimintoa. Sisemmän silmukan silmukkamuuttujaa merkitään muuttujakirjaimella *m*.

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

# Kysytään käyttäjältä, kuinka monta jäsentä lukujonoon tulostetaan.
# Vastaus pitää muuttaa kokonaisluvuksi, jotta sitä voidaan käyttää
# silmukkamuuttujan lopetusluvuksi.

lkm = int(input('Kuinka monta jäsentä kuviojonoon tulostetaan? (maks. 6)'))

ukko.shape('circle')
ukko.penup()

# Määritetään kuviojonon aloituspiste.
ukko.goto(-300, 0)

# Koska alue pitää olla yhden pidempi kuin haluttu kuviojono, pitää muuttujaan
# lkm lisätä 1.
for n in range(1, lkm + 1):

# Sisempää silmukkaa toistetaan ulomman silmukan silmukkamuuttujan verran.
    for m in range(n):
        ukko.stamp()
        ukko.forward(20)

# Määritetään kuviodien etäisyys toisistaan uloimmassa silmukkarakenteessa.
# Huomaa, että sisennys on vain neljä välilyöntiä.
    ukko.forward(55)
```

Kun käyttäjä haluaa tulostaa neljä kuviojonon jäsentä, on tuloste seuraava:



TEHTÄVÄ 3

Tavoitteena on muodostaa lukujono.

Run

Tee ohjelma, joka tulostaa lukujonon väliltä 2–20. Lukujonon ensimmäinen jäsen on 2 ja seuraava jäsen on aina kolme edellistä suurempi.

TEHTÄVÄ 4

Tavoitteena on muodostaa lukujono, jonka seuraava jäsen on aina edellistä pienempi.

Run

Tee ohjelma, joka tulostaa lähtölaskennan luvusta 10 lukuun 0. Seuraava jäsen on aina yhden edellistä pienempi.

TEHTÄVÄ 5

Tavoitteena harjoitella kuviojonon tulostamista leimaustoiminnolla.

Run

Tee ohjelma, joka tulostaa lukujonoa 1, 2, 3, 4, 5 vastaavan kuviojonon. Alla on mallikuva tällaista lukujonoa vastaavasta kuviojonosta. Käytä haluamaasi hahmoa kuviojonon muodostamiseen.



TEHTÄVÄ 6

Tavoitteena on tulostaa kuviojono piirtämällä.

Run

Tee ohjelma, joka tulostaa lukujonoa 2, 4, 6 vastaavan kuviojonon. Käytä kuviona piirrettyä ympyrää.

- Ympyrä saadaan piirrettyä circle-komennolla.
- Ympyrän säteeksi kannattaa valita 10 tai 20.
- Määritä kuvioiden välinen siirtymä sopivaksi.
- Määritä jäsenien välinen etäisyys.
- Muista nostaa ja laskea kynää tarvittaessa, muutoin siirtymiset piirtyvät viivana.

EHTORAKENNE VAIHTOEHTOISILLE TAPAHTUMILLE

Jos-ehtorakenteessa tapahtuma tapahtuu, jos annettu ehto täyttyy. Usein tarkasteltavia ehtoja on useampia kuin yksi. Ohjelma käy läpi kaikki jos-lauseet riippumatta siitä, kohdataanko ehdon täyttävä tapahtuma jossain vaiheessa. Erillisiä jos-ehtolauseita voidaan korvata käyttämällä **jos-muuten-rakennetta**, jossa ohjelma suorittaa tietyn tapahtuman, kun vertailu on tosi ja toisen tapahtuman vertailun ollessa epätosi.

Ehtorakenne mahdollistaa sen, että eri valinnat tuottavat erilaisen lopputuloksen.

Ehtorakenne if-else

if ehto tai vertailu:

Tapahtuma, joka tapahtuu, kun vertailu on tosi.

else:

Tapahtuma, joka tapahtuu, kun vertailu on epätosi.

Esimerkki 3

Testissä on oikein-väärin-väittämiä. Tee ohjelma, jossa käyttäjältä kysytään seuraavasti:

”Oikein vai väärin: $3x + x = 4x$. k vai e?”

Jos vastaus on oikein, ohjelma onnittelee oikeasta vastauksesta. Jos käyttäjän antama vastaus on väärin, ohjelma suosittelee käyttäjää kertaamaan asian uudelleen.

Ratkaisu

Vaihtoehtoisia tapahtumia on kaksi, joten ohjelma voidaan toteuttaa joko kahdella jos-lauseella tai käyttämällä jos-muuten-ehtorakennetta.

Tapa 1: Jos-ehtorakenteella toteutettuna

```
vastaus=input(u'Oikein vai väärin: 3x + x = 4x. k vai e?')
if vastaus == 'k':
    print('Onneksi olkoon oikeasta vastauksesta!')
if vastaus == 'e':
    print(u'Kertaa asia vielä uudelleen.')
```

Tapa 2: Jos-muuten-ehtorakenteella toteutettuna

```
vastaus=input(u'Oikein vai väärin: 3x + x = 4x. k vai e?')
if vastaus == 'k':
    print('Onneksi olkoon oikeasta vastauksesta!')
else:
    print(u'Kertaa asia vielä uudelleen.')
```


Esimerkki 4

Tee ohjelma, joka piirtää kuviojonon käyttäjän haluamalla hahmolla. Käyttäjä voi valita hahmoksi joko ympyrän, neliön tai kolmion. Käyttäjältä myös kysytään, kuinka monta jäsentä kuviojonoon tulostetaan.

Ratkaisu

Valittavana on kolme eri vaihtoehtoa, joista jokainen tekee eri tapahtuman. Käytetään rakennetta if-elif-else.

KOODI

```
from turtle import*
alue = Screen()
ukko = Turtle()
ukko.penup()
ukko.goto(-300, 50)

lkm = int(input(u'Kuinka monta jäsentä kuviojonoon piirretään? Anna luku väliltä 1 - 7.))
hahmo = input(u'Valitse käytettävä hahmo: ympyrä (y), neliö (n) vai kolmio (k).')

if hahmo == 'y':
    # Ensimmäinen vaihtoehto
    ukko.shape('circle')
    for n in range(1, lkm + 1):
        for m in range(n):
            ukko.stamp()
            ukko.forward(20)
        ukko.forward(50)
elif hahmo == 'n':
    # Toinen vaihtoehto
    ukko.shape('square')
    for n in range(1, lkm + 1):
        for m in range(n):
            ukko.stamp()
            ukko.forward(20)
        ukko.forward(50)
else:
    # Kolmas vaihtoehto
    ukko.shape('triangle')
    for n in range(1, lkm + 1):
        for m in range(n):
            ukko.stamp()
            ukko.forward(20)
        ukko.forward(50)
```

Tuloste, kun hahmoksi valitaan ympyrä ja kuviojonoon halutaan kolme jäsentä.



Toistuvasti esiintyvä osa koodista kannattaa tehdä kerran. Testaamisen jälkeen koodia voi kopioida muihin kohtiin ja tehdä niihin kohtiin tarvittavat muutokset.

TEHTÄVÄ 7

Tavoitteena on harjoitella if-else-rakenteen käyttämistä.

Run

Tee ohjelma, jossa käyttäjältä kysytään, onko hän kissa- vai koiraihminen. Kissaihmisille ohjelma tulostaa lauseen ”Kissoja onkin mukava silitellä ja kuunnella niiden kehräystä.”. Koiraihmisille ohjelma tulostaa ”Koirien kanssa onkin mukava ulkoilla ja harrastaa.” Käytä ohjelmassa if-else-rakennetta.

TEHTÄVÄ 8

Tavoitteena on harjoitella if-elif-else rakenteen käyttämistä.

Run

Tee ohjelma, joka toimii yksinkertaisena nelilaskimena. Käyttäjältä kysytään kaksi lukua ja minkä laskutoimituksen, yhteen-, vähennys-, kerto- vai jakolaskun, hän haluaa niillä tehdä.

- Määritä muuttujat ja niiden arvoksi käyttäjältä kysyttävät luvut. Vastaus on muutettava kokonaisluvuksi, jotta sitä voidaan käyttää myöhemmin laskutoimituksissa.
- Määritä muuttuja, jonka arvoksi tulee käyttäjän valitsema laskutoimitus. Mieti, miten haluat, että käyttäjä ilmaisee valitsemansa laskutoimituksen, jotta siitä voi tehdä ehtorakenteen.
- Ohjelma tulostaa valitun laskutoimituksen ja laskun vastauksen.
- Koska vaihtoehtoja on neljä, tarvittava ehtorakenne on if-elif-elif-else.

TEHTÄVÄ 9

Tavoitteena on syventää jos-muuten-rakenteen käyttämistä.

Run

Tee ohjelma, jossa käyttäjälle esitetään kysymyksiä lausekkeen ja muuttujan arvosta ja hänen on valittava kahdesta vaihtoehdosta oikea.

Käyttäjälle esitettävät kysymykset ja vastausvaihtoehdot:

- Mikä on lausekkeen $x + 1$ arvo, kun x saa arvon 5? a) 5 b) 6
- Mikä on lausekkeen $2n + 1$ arvo, kun $n = 10$? a) 21 b) 211
- Mikä on lausekkeen $3a$ arvo, kun $a = -4$? a) 12 b) -12
- Millä x :n arvolla $x - 4 = 9$? a) $x = 5$ b) $x = 13$

Käyttäjä valitsee joko a tai b. Oikeasta vastauksesta käyttäjä saa kannustuksena ”Hienosti tiedetty!”. Väärästä vastauksesta käyttäjä saa palautteena ”Kaipaavat vielä hieman harjoitusta tämän tyyppisestä tehtävästä.”

Jokaiseen kysymykseen tulee tehdä erillinen jos-muuten-rakenne.

TEHTÄVÄ 10

Tavoitteena on palauttaa mieleen merkkijonojen yhtenevyyden vertailu.

Run

Tee ohjelma, jossa käyttäjän pitää arvata jotain. Käyttäjä voi esimerkiksi

- arvata tunnussanan
- keksiä vastauksen arvoitukseen
- keksiä, mitä sanaa yritetään selittää
- keksiä, mitä vieraskieliset sanat tarkoittavat.

Päätä itse, mitä ohjelman antaa palautteena oikeasta ratkaisusta ja mitä väärästä.

Pohdi, onko ratkaisun kirjoitusasulla väliä eli pitääkö isot kirjaimet olla isoja vai onko itse sana tärkein kirjoitusmuodosta huolimatta.

LISÄTEHTÄVIÄ

LISÄTEHTÄVÄ 1

Tavoitteena on syventää jos-muuten-rakenteen käyttämistä.

- Tee ohjelma, jossa käyttäjältä kysytään pituus senttimetreinä ja haluaako hän muuttaa sen millimetreiksi vai metreiksi.