OHJELMOINTIA PYTHONILLA

MATEMATIIKKA 7-9 -LUOKKA

Juuso Linnusmäki, 2020

Sisällysluettelo

Juuso Linnusmäki, 2020

Päivitetty viimeksi: 26.4.2021

Ohjelmointitehtäviä 7. luokalle

Oppitunnit 1-2: Turtle-grafiikan perusteita

KILPIKONNAGRAFIIKKAA

Kilpikonnagrafiikka tarkoittaa grafiikan piirtoa näkymättömän kursorin, "kilpikonnan" avulla. Aloitettaessa kursori sijoitetaan jonnekin päin ruutua, minkä jälkeen sille annetaan komentoja, joiden avulla kursori liikkuu näytöllä. Ohjelmointiympäristö, jota käytämme, tukee vain osaa turtle-grafiikan komennoista. Tämän vuoksi teemme varsin yksinkertaisia tehtäviä turtle-grafiikkaa hyödyntäen.

KILPIKONNAGRAFIIKAN ALKUMÄÄRITTELYT

```
!pip3 install ColabTurtle
import ColabTurtle.Turtle
```

Nämä määrittelyt on oltava ohjelman alussa. Tehtäväpohjissa ne ovat jo valmiina.

```
alex = ColabTurtle.Turtle
alex.initializeTurtle()
```

Annetaan kilpikonnallemme nimi, esim. alex. Käytä haluamaasi nimeä. Voit nimetä myös useita eri kilpikonnia.

,

KILPIKONNAGRAFIIKAN YLEISIMPIÄ KOMENTOJA

Komento	Mitä tapahtuu?
alex.forward(100)	Alex-kilpikonna liikkuu eteenpäin 100 askelta.
alex.backward(50)	Alex-kilpikonna liikkuu taaksepäin 50 askelta.
jesse.right(90)	Jesse-kilpikonna kääntyy oikealle 90 astetta.
jesse.left(50)	Jesse-kilpikonna kääntyy vasemmalle 50 astetta.
mike.penup()	Mike-kilpikonna tekee toimintoja piirtämättä, kilpikonnan ”kynä nousee ylös paperista”.
mike.pendown()	Mike-kilpikonna jatkaa piirtämistä kilpikonnan, kilpikonnan ”kynä takaisin kiinni paperiin”.
john.setx(50)	John-kilpikonna asetetaan uuteen sijaintiin x-suunnassa. Y-koordinaatti pysyy samana.
john.sety(100)	John-kilpikonna asetetaan uuteen sijaintiin y-suunnassa. x-koordinaatti pysyy samana.
jessie.goto(50, 300)	Jessie-kilpikonna siirtyy sijaintiin (50, 300).
jessie.color(”green”)	Jessie-kilpikonnan piirtoväri vaihdetaan vihreäksi. Tuetut värit ovat valkoinen, keltainen, oranssi, punainen, vihreä, sininen, violetti, harmaa ja musta.
jessie.width(10)	Jessie-kilpikonnan piirtämän viivan leveys vaihdetaan arvoon 10.
bgcolor(”red”)	Piirtoalueen taustaväri vaihdetaan punaiseksi. Tuetut värit ovat valkoinen, keltainen, oranssi, punainen, vihreä, sininen, violetti, harmaa ja musta.

ESIMERKKI 1: Värikäs kahdeksankulmio

```
!pip3 install ColabTurtle
import ColabTurtle.Turtle
```

Alkumäärittelyt

```
alex = ColabTurtle.Turtle
alex.initializeTurtle()
```

Kilpikonnan määrittäminen Alex-nimiseksi.

```
bgcolor("white")
```

Piirtoalueen taustaväri valkoiseksi.

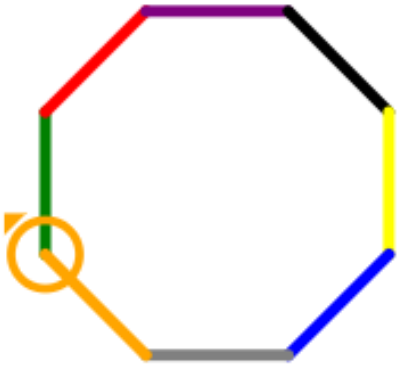
```
alex.color("green")
alex.forward(50)
alex.right(45)
```

piirtoväri vihreäksi, 50 askelta eteenpäin ja 45 astetta käännös oikealle.

```
alex.color("red")
alex.forward(50)
alex.right(45)
```

piirtoväri punaiseksi, 50 askelta eteenpäin ja 45 astetta käännös oikealle.

```
alex.color("purple")
alex.forward(50)
alex.right(45)
alex.color("black")
alex.forward(50)
alex.right(45)
alex.color("yellow")
alex.forward(50)
alex.right(45)
alex.color("blue")
alex.forward(50)
alex.right(45)
alex.color("grey")
alex.forward(50)
alex.right(45)
alex.color("orange")
alex.forward(50)
```



Tee tehtävät 1-10

Tehtäväosio 1: Turtlegrafiikkaa

1. Piirrä sininen neliö, jonka sivun pituus on 100 yksikköä.

---

2. Piirrä
   a. 120 asteen kulma
   b. 45 asteen kulma
   c. 250 asteen kulma

   vinkki: Mieti, kuinka monta astetta kilpikonnan on käännyttävä, jotta kilpikonna piirtää kyseisen kokoisen kulman.

---

3. Piirrä kilpikonnalla reitti, joka sisältää kaksi suoraa kulmaa, kolme terävän kulmaa, yhden kuperan kulman sekä kaksi tylppää kulmaa. Tee reitti siten, että se kiertää piirtoaluetta myötäpäivään ja pyydetyt kulmat jäävät piirtoalueen oikealle puolelle etenemissuunnasta katsottuna.

---

4. Piirrä haluamasi värinen ja kokoinen tasasivuinen kolmio.

   vinkki: Mieti, kuinka monta astetta kilpikonnan on käännyttävä, jotta tasasivuiseen kolmioon muodostuu 60 asteen kulmat.

---

5. Piirrä monivärinen suunnikas, jonka viereisten kulmien asteluvut ovat 50 ja 130 astetta.

---

6. Piirrä kaksi eriväristä yhdensuuntaista suoraa ja niille jokin normaali.

---

7. Piirrä tasakylkinen kolmio, jonka kyljen pituus on 73 yksikköä ja huippukulman suuruus 64 astetta.

---

8. Piirrä haluamasi värinen ja kokoinen säännöllinen kuusikulmio. Luo myös toinen kilpikonna, joka piirtää haluamasi värisen ja kokoisen säännöllisen 10-kulmion.

---

9. Piirrä oranssi suorakulmio ja jaa se lävistäjälle kahdeksi kolmioksi.

---

10. Suunnittele ja toteuta värikäs taideteos turtle-grafiikan avulla. Mitä matemaattisia muotoja ja käsitteitä taideteokseesi liittyy?

Oppitunnit 3-5: Tiedon tulostus, laskutoimitukset ja muuttujat

TIEDON TULOSTAMINEN NÄYTÖLLE, print()

Ohjelma tulostaa näytölle tietoa komennon **print()** avulla. Sulkeisiin tulee tulostettava merkkijono, muuttujan nimi tai laskutoimitus. Merkkijono on kirjoitettava lainausmerkkeihin. Laskutoimitusten tai muuttujien yhteydessä lainausmerkkejä ei käytetä.

ESIMERKKI 2

```
print(”Tämä on merkkijono ja siksi lainausmerkeissä”)
print(5+10)
summa=1+2+3+4+5
print(summa)
```

Tässä ohjelma tulostaa laskutoimituksen vastauksen.

Tässä on määritelty muuttuja nimeltä summa.

Tässä ohjelma tulostaa muuttujassa **summa** olevan tiedon.

Tee tehtävät 11-12

MUUTTUJAT

Ohjelmoinnissa on usein kätevää määritellä muuttujia. Niiden avulla koodista saa usein helpommin ymmärrettävän ja yksinkertaisemman. Muuttujan nimen käyttäjä voi valita vapaasti muutamaa sääntöä noudattaen:

- Ei ääkkösiä
- Ei tiettyjä erikoismerkkejä
- Ei välilyöntejä
- Ensimmäinen kirjain oltava pieni

Uuden muuttujan määritteleminen on helppoa. Se tehdään yhtäsuuruusmerkin **=** avulla.

ESIMERKKI 3

```
vanha_hinta=10.5
uusi_hinta=7.5
print(”Hinta on muuttunut”, vanha_hinta – uusi_hinta, ”euroa!”)
```

Tässä määritellään muuttujat. Desimaalierotin on piste!

Muuttujien välinen laskutoimitus

Merkkijonot ovat lainausmerkeissä, mutta muuttujat eivät ole. Merkkijonot ja muuttujat erotetaan toisistaan print()-komennon sisällä pilkuilla!

Tee tehtävät 13-16

LASKUTOIMITUKSET

Laskutoimitukset ovat samankaltaisia kuin laskimella laskettaessa:

- yhteenlasku, **+**
- vähennyslasku, **-**
- kertolasku, ***
- jakolasku, **/**
- potenssi, ****
- jakolaskun kokonaisosa, **//**
- jakolaskun jakojäännös, **%**

Laskutoimituksissa voi käyttää sulkeita normaaliin tapaan.

KOODIN KOMMENTOINTI

Tietokoneohjelmien lukemisen helpottamiseksi ja ohjelmoijan omien ajatusten jäsentämiseksi ohjelmiin lisätään usein kommenttirivejä. Kommentteihin kirjoitetaan lyhyitä kuvauksia siitä, mitä ohjelmassa tapahtuu. Kommenttirivi alkaa aina #-merkillä ja se ei vaikuta mitenkään varsinaisen ohjelman suoritukseen. Kommentteja voi kirjoittaa selittämään ohjelman osien tai yksittäisen ohjelmarivin toimintaa.

Kommentointia kannattaa käyttää, sillä se on osa hyvää ohjelmointikäytäntöä!

ESIMERKKI 4: Laskutoimituksia

```
# Määritellään aluksi muuttujia
```

Kommentointi hyvää ohjelmointikäytäntöä!

```
luku1=5.2
luku2=18.9
tulo=luku1*luku2
```

Huomaa, että desimaaliluvuissa käytetään pilkun sijaan pistettä!

Muuttujien avulla voidaan määritellä helposti myös uusia muuttujia.

```
# Tulostetaan lukujen summa ja tulo
print(”Lukujen summa on”, luku1+luku2)
print(”Lukujen tulo on”, tulo)
```

Tässä laskutoimituksia yllä määriteltyjen muuttujien avulla.

```
print(”Kaksi potenssiin kolme on”, 2**3)
print(”Lukujen 5 ja 7 erotus on”, 5-7)
```

Tässä laskutoimituksia luvuilla. Tässä ei ole käytetty yllä määriteltyjä muuttujia.

ESIMERKKI 5: Muuttolaatikoiden vuokra

```
laatikoiden_maara=9
# Ilmaistaan vuokran lauseke
vuokra=25+8*laatikoiden_maara
# tulostetaan vuokran suuruus
print(”Vuokra”, laatikoiden_maara, ”muuttolaatikolta on yhteensä”, vuokra, ”euroa!”)
```

Määritellään muuttuja **laatikoiden_lukumaara.**

Määritellään muuttuja **vuokra** laskulausekkeella.

Ohjelma tulostaa käyttäjälle tiedon vuokran suuruudesta. Print()-komennossa on käytetty yllä määriteltyjä muuttujia **laatikoiden_maara** ja **vuokra**.

Tee tehtävät 17-22

Tehtäväosio 2: Perustehtävät

11. Tee ohjelma, joka tulostaa näytölle tekstin *Olen 7. luokkalainen Pikkolan koulun opiskelija!*

---

12. Tee ohjelma, joka tulostaa näytölle seuraavien laskutoimituksien tulokset:

a. 9+10
b. 121-89+12+91
c. 19,5-21,2+1,05

---

13. Määrittele aluksi ohjelman alkuun muuttuja koulumatka ja tallenna muuttujaan koulumatkasi pituus kilometreinä.

vinkki: `koulumatka=2.6`

Tee sitten ohjelma, joka tulostaa näytölle tekstin *Koulumatkani pituus kilometreinä on…*
Käytä print()-komennossa ohjelman alussa määrittelemääsi muuttujaa **koulumatka**!

vinkki: `print(”koulumatkani pituus kilometreinä on”, koulumatka)`

Mitä tapahtuu, jos muutat ohjelman alussa määrittelemäsi muuttujan koulumatka lukuarvoa joksikin toiseksi?

---

14. Määrittele aluksi ohjelman alkuun muuttuja, jonka nimi on **opettaja** ja tallenna siihen opettajasi nimi. Opettajan nimi on laitettava lainausmerkkeihin, koska opettajan nimi on merkkijono!

vinkki: `opettaja=”Pekka Pikkola”`

Tee ohjelma, joka tulostaa näytölle tekstin *Minun matematiikan opettajani on…*

Käytä print()-komennossa ohjelman alussa määrittelemääsi muuttujaa **opettaja**!

---

15. Määrittele aluksi ohjelman alkuun muuttujat **koulumatka** ja **kaverin_koulumatka**. Tallenna muuttujiin sopivat luvut kuvaamaan koulumatkojen pituuksia. Tee sitten ohjelma, joka laskee sinun ja kaverisi koulumatkojen erotuksen ja antaa näytölle tekstin *Minun ja kaverini koulumatkojen erotus on …. kilometriä*

vinkki: `print(”Minun ja kaverini koulumatkojen erotus on”, koulumatka - kaverin_koulumatka, ”kilometriä!”)`

Kokeile muuttaa muuttujien lukuarvoja. Mitä huomaat ohjelman suorituksessa? Milloin vastaukseksi tulee positiivinen luku? Entä negatiivinen?

---

16. Määrittele aluksi ohjelman alkuun muuttujat **nimi, ika, kaverin_nimi ja kaverin_ika** ja tallenna niihin sopivat tiedot. Muista, että merkkijonot tallennetaan lainausmerkkien kera, numerot ilman lainausmerkkejä!

Tee ohjelma, joka tulostaa näytölle tekstin *Minun nimi on … ja olen … vuotta vanha. Kaverini nimi on …. ja hän on … vuotta vanha.*

Käytä print()-komennossa muuttujia, jotka olet määritellyt ohjelman alussa!

17. Määrittele ohjelman alkuun muuttujat **x** ja **y** ja tallenna niihin jotkin valitsemasi kokonaisluvut. Tee sitten ohjelma, joka tulostaa näytölle seuraavien laskutoimituksien tulokset:

d. $x + y$
e. $x - 5y$
f. $(x + y) \cdot 5$
g. $\frac{x+y}{3}$
h. $x + \frac{5y}{3}$

---

18. Lauseke 50-19x ilmaisee, kuinka monta kilometriä Pekan pyörälenkkiä on jäljellä, kun Pekka on pyöräillyt x tuntia. Kuinka monta kilometriä matkaa oli jäljellä, kun Pekka oli pyöräillyt a) 2 tuntia, b) 2,5 tuntia, c) 0,5 tuntia. Ratkaise tehtävä ohjelmoimalla!

vinkki:
x=2
print(”Pyörämatkaa on jäljellä”, 50-19***x**, ”kilometriä)

---

19. Rooman lomareissun hinta määräytyy lausekkeen 150+89x avulla, missä **x** on lomakohteessa vietettyjen hotelliöiden lukumäärä. Mallorcan lomareissun hinta on sen sijaan 230+84x, missä **x** on lomakohteessa vietettyjen hotelliöiden lukumäärä. Määrittele alkuun muuttuja **x** ja anna sille jokin arvo ja tulosta näytölle molempien reissujen kokonaishinta. Vaihda x:n arvoa ja etsi kokeilemalla x:n arvo, jolla molemmat reissut ovat yhtä kalliita.

---

20. Määrittele ohjelman alkuun muuttujat **pituus** ja **leveys** sekä tallenna niihin sopivat luvut. Tee sitten ohjelma, joka tulostaa näytölle suorakulmion piirin, kun sen leveys ja pituus ovat ohjelmassasi määritellyt luvut.

vinkki: print(”Suorakulmion, jonka pituus on”, **pituus**, ”ja leveys on”, **leveys**, ”piiri on”, …)

Kokeile muuttaa muuttujien pituus ja leveys arvoja. Mitä huomaat?

---

21. Määrittele ohjelman alkuun muuttujiin seuraavat kilohinnat:

kurkku=3.55
tomaatti=2.90
peruna=0.85
paprika=5.09

Tee vielä muuttuja **maara** (määrä) johon tallennat halutun kilomäärän. Tämän jälkeen ohjelma laskee kuinka paljon maksaa muuttujan **maara** verran erilaisia tuotteita.

vinkki: print(**maara**, ”kilogrammaa kurkkua maksaa”, **maara*kurkku**, ”euroa!”)

---

22. Suunnittele ja toteuta jokin oma ohjelmasi, jossa käytät mahdollisimman monipuolisesti eri muuttujia!

Ohjelmointitehtäviä 8. luokalle

Oppitunnit 1-3: Tiedon tulostus, laskutoimitukset, muuttujat, syötteet ja tietotyypit

Tee aluksi kertaavat tehtävät 23-27

KÄYTTÄJÄN ANTAMAT SYÖTTEET, input()

Kun ohjelman halutaan vuorovaikuttavan ohjelman käyttäjän kanssa, on käytettävä komentoa **input().** Tyypillisesti **input()**-komennon avulla käyttäjältä kysytään jotain tietoa, jota ohjelma tarvitsee ohjelman suorituksessa.

Jotta käyttäjän ohjelmalle syöttämää tietoa voidaan käyttää ohjelman eri kohdissa, on tieto tallennettava johonkin muuttujaan.

ESIMERKKI 6: Mikä sinun nimesi on?

```
nimi=input(”Mikä sinun nimesi on \n”)
print(”Hauska nähdä sinua”, nimi)
```

Käyttäjältä kysytty tieto tallennetaan tässä muuttujaan **nimi.** Komento **\n** tekee rivinvaihdon.

Tee tehtävä 28

HIEMAN TIETOTYYPEISTÄ

Tieto voi olla tyypiltään esimerkiksi tekstiä eli merkkijono, kokonaisluku tai desimaaliluku:

- *Matematiikka on kivaa* on merkkijono, **str()**
- *5* on puolestaan kokonaisluku, **int()**
- *1,24533* (ohjelmoinnissa 1.24533) on desimaaliluku, **float()**

Tietotyyppiä voi vaihtaa edellä mainittujen komentojen **str(), int()** ja **float()** avulla. Erityisen tärkeää tämä on käytettäessä **input()** -komentoa, koska tällä komennolla annetut syötteet ohjelma tulkitsee automaattisesti merkkijonoina.

Merkkijonojen laskutoimitukset eivät toimi samalla tavalla kuin kokonais- tai desimaaliluvuilla. Tämän vuoksi käyttäjän antamat luvut on muutettava aluksi kokonais- tai desimaaliluvuksi!

ESIMERKKI 7

```
# Ohjelma kysyy käyttäjän ikää ja pituutta
ika=int(input(”Minkä ikäinen olet? \n”))
pituus=float(input(”Kuinka pitkä olet? Anna vastaus metreinä! \n”))
```

int() muuttaa käyttäjän antaman syötteen kokonaisluvuksi!

float() muuttaa käyttäjän antaman syötteen desimaaliluvuksi!

```
# Ohjelma tulostaa käyttäjän iän kahden vuoden kuluttua
print(”Kahden vuoden kuluttua olet”, ika+2, ”-vuotias!”)
```

Muista erottaa teksti ja muuttujat toisistaan pilkuilla. Teksti tulee lainausmerkkeihin, muuttujat ja laskulausekkeet eivät! Erottimeksi käy pilkun lisäksi myös +, mutta tällöin muuttujan ja tekstin väliin ei tule automaattisesti välilyöntiä.

```
# Ohjelma laskee käyttäjän ja maailman pisimmän henkilön pituuksien
erotuksen
print(”Maailman pisin ihminen oli 2,72 m pitkä. Teidän pituuksilla on
eroa huimat”, 2.72-pituus, ”metriä!”)
```

Tee tehtävä 29

ESIMERKKI 8: Ravintolan tippilaskuri

```
print(”Tämä ohjelma laskee sinulle, kuinka paljon sinun on maksettava
ravintolalaskun lisäksi juomarahaa”)

# Ohjelma kysyy käyttäjältä laskun loppusummaa ja tippiprosenttia
lasku=float(input(”Kuinka paljon on ravintolalaskun loppusumma? \n”))

tippiprosentti=float(input(”Kuinka monta prosenttia haluat antaa
juomarahaa? \n”))

# Muunnos desimaaliluvuksi
tippi_desimaaliluku=tippiprosentti/100
```

Muutetaan tippiprosentti desimaaliluvuksi ja tallennetaan se uuteen muuttujaan.

```
# Laskun loppusumman lauseke
loppusumma=lasku+lasku*tippi_desimaaliluku
```

Lisätään alkuperäiseen laskuun prosenttiosuus laskusta. Tallennetaan tulos uuteen muuttujaan.

```
# Tulostetaan näytölle loppusumma
print(”Juomarahan kanssa ravintolalasku on yhteensä”, loppusumma,
”euroa!”)
```

Tee prosenttilaskentatehtävät 30-35

Tehtäväosio 3: Perustehtävät

23. Tee ohjelma, joka tulostaa tekstin *Hello World!* näytölle.

---

24. Tee ohjelma, joka tulostaa jonkin kivan tervehdyksen matematiikan opettajallesi. Lisää ohjelmaan myös jokin matematiikkaan/koodaamiseen liittyvä tervehdys, jonka ohjelma tulostaa näytölle.

---

25. Määrittele muuttujat **syntymavuosi** ja **vuosi_nyt** ja tallenna niihin sinuun itseesi sopivat oikeat vuosiluvut. Tee ohjelma, joka tulostaa näytölle tekstin *Olen syntynyt vuonna….* ja *Nyt on vuosi….*

vinkki: `print(”Olet syntynyt vuonna”, syntymavuosi)`

---

26. Muokkaa edellisen tehtävän ohjelmaa siten, että ohjelma laskee muuttujien avulla, kuinka monta vuotta täytät/täytit tänä vuonna ja tulostaa näytölle tekstin *Täytät tänä vuonna…*

---

27. Määrittele muuttujat **luku1** ja **luku2**. Tee sitten ohjelma, joka laskee näiden lukujen yhteen-, vähennys-, kerto ja jakolaskun tuloksen näytölle.

vinkki: `print(”Luvun”, luku1, ”ja luvun”, luku2, ”summa on”, luku1+luku2)`

---

28. Tee ohjelma, joka kysyy aluksi sinun nimeäsi ja tallentaa käyttäjän antaman vastauksen muuttujaan **nimi**. Tämän jälkeen ohjelma tulostaa näytölle tekstin *Onpa kivaa koodata kanssasi… (ohjelman käyttäjän nimi).*

---

29. Tee ohjelma, joka kysyy kuinka monta tuntia nukut vuorokaudessa ja tallentaa vastauksen muuttujaan **uni**. Tämän jälkeen ohjelma tulostaa näytölle tiedon, kuinka monta prosenttia vuorokaudessa nukut.

vinkki: `print(”Nukut vuorokaudesta”, (uni/24)*100, ”prosenttia!”)`

Olisi hyvä, että jokaisen ohjelman alussa ohjelma antaisi käyttäjälle tiedon, mitä kyseinen ohjelma tekee, esim.

```
print(”Tämä ohjelma laskee kuinka monta prosenttia jokin luku on
jostain toisesta luvusta”)
```

---

30. Kuinka monta prosenttia

Tee ohjelma, joka laskee, kuinka monta prosenttia jokin luku on jostain toisesta luvusta. Aluksi ohjelma kysyy lukua 1 ja tallentaa sen muuttujaan **luku1**. Tämän jälkeen ohjelma kysyy toista lukua ja tallentaa sen muuttujaan **luku2**. Ohjelma laskee kuinka paljon on (**luku1/luku2)*100** ja tulostaa kysytyn tiedon näytölle.

vinkki: `luku1=float(input(”Anna jokin luku \n”))`

```
print(luku1, ”on luvusta”, luku2, luku1/luku2*100, ”prosenttia”)
```

---

31. Alennusprosentti

Tee ohjelma, joka laskee, kuinka suuri jokin alennus on prosentteina. Aluksi ohjelma kysyy alkuperäistä hintaa ja tallentaa tiedon muuttujaan **alkuhinta**. Seuraavaksi ohjelma kysyy alennettua hintaa ja tallentaa sen muuttujaan **alehinta**. Tämän jälkeen ohjelma laskee, kuinka suuri alennus on prosentteina ja tulostaa tiedon näytölle.

vinkki: `muutos=alkuhinta-alehinta`

---

32. Prosenttiosuus

Tee ohjelma, jonka avulla saat laskettua prosenttiosuuden jostakin luvusta. Aluksi ohjelma kysyy lukua, josta prosenttiosuus halutaan laskea. Tämän jälkeen ohjelma kysyy, kuinka monta prosenttia luvusta halutaan laskea. Seuraavaksi ohjelma laskee kysytyn prosenttiosuuden ja tulostaa sen näytölle.

---

33. Alennettu hinta

Tee ohjelma, joka laskee tuotteen uuden hinnan, kun alkuperäinen hinta ja alennusprosentti tiedetään. Ohjelma kysyy aluksi tuotteen alkuperäistä hintaa ja tallentaa sen sopivasti nimettyyn muuttujaan. Tämän jälkeen ohjelma kysyy, kuinka monta prosenttia tuote on alennuksessa ja tallentaa myös tämän tiedon muuttujaan. Sen jälkeen ohjelma laskee alennetun hinnan ja tulostaa sen näytölle.

---

34. Arvonlisävero

Tee ohjelma, joka laskee tuotteen verottoman hinnan. Aluksi ohjelma kysyy tuotteen myyntihintaa ja arvonlisäveroprosenttia ja tallentaa tiedot sopiviin muuttujiin. Tämän jälkeen ohjelma tulostaa näytölle tuotteen arvonlisäverottoman hinnan.

---

35. Liuosten sekoitus

Tee ohjelma, joka kertoo syntyvän liuoksen pitoisuuden, kun sekoitetaan kahta eri vahvuista liuosta toisiinsa. Aluksi ohjelma kysyy liuoksen 1 massan ja pitoisuuden (%) ja tallentaa tiedot sopiviin muuttujiin. Tämän jälkeen ohjelma kysyy liuoksen 2 massan ja pitoisuuden (%) ja tallentaa myös nämä tiedot sopiviin muuttujiin. Tämän jälkeen ohjelma laskee syntyvän sekoituksen pitoisuuden ja tulostaa tiedon näytölle.

Oppitunnit 4-6: Ehtorakenne, vertailuoperaattorit ja totuusarvot

EHTORAKENNE

if-ehtolauseilla voidaan ohjata ohjelman toimintaa erilaisten ehtojen avulla. Vain ensimmäinen ehdot täyttävä osio suoritetaan, loput jätetään huomiotta. Ehtolauseessa voi olla (tai olla olematta) rajaton määrä **elif**-osia ja yksi **else**-osa.

RAKENNE:

```
If ehto kirjoitetaan tähän:
    tähän kirjoitetaan, mitä ohjelma tekee, jos ehto on TOSI (TRUE)
elif ehto kirjoitetaan tähän:
    tähän kirjoitetaan, mitä ohjelma tekee, jos ehto on TOSI (TRUE)
else:
    Mikäli yksikään yllä olleista ehdoista ei toteudu, ohjelman
    suoritus etenee tämän komennon mukaan.
```

Ehtorakenteeseen kuuluu ehdon jälkeinen sisennys. Ilman sisennystä rakenne ei toimi oikein. Älä unohda myöskään ehdon jälkeistä kaksoispistettä!

VERTAILUOPERAATTORIT JA TOTUUSARVOT

Vertailu	Milloin tosi?
X == Y	X on sama kuin Y
X != Y	X ei ole sama kuin Y
X < Y	X on pienempi kuin Y
X <= Y	X on pienempi tai sama kuin Y
X > Y	X on suurempi kuin Y
X >= Y	X on suurempi tai sama kuin Y

ESIMERKKI 9: Mopoikäinen?

```
ika=int(input(”Kuinka vanha olet? \n”))
#Iän vertailu ehtolauseen avulla
if ika<15:
    print(”Et saa vielä ajaa mopolla!”)
else:
    print(”Saat ajaa mopolla!”)
```

Jos ikä on pienempi kuin 15, niin:

Muussa tapauksessa:

Ohjelma kysyy käyttäjän ikää ja tallentaa sen muuttujaan **ika**. Koska ikä on kokonaisluku, on **input()**-komennon lisäksi käytettävä komentoa **int()**, joka muuttaa iän merkkijonon sijaan kokonaisluvuksi.

Tee tehtävät 36 ja 37

VERTAILUOPERAATTORIEN YHDISTELY

Monimutkaisempia ehtolauseita varten vertailuoperaattoreita on yhdisteltävä komentojen **and**, **or** ja **not** avulla.

Yhdistys	Milloin tosi?
A and B	A on tosi ja B on tosi
A or B	A on tosi tai B on tosi
not A	A ei ole tosi

ESIMERKKI 10: Ikäkysely

```
ika=int(input(”Kuinka vanha olet? \n”))
if ika < 0 or ika > 120:
    print(”Nyt huijaat!”)
else:
    print(”Selvä homma! Olet juuri sopivan ikäinen koodauksen
    opiskelija!”)
```

Jos ikä on pienempi kuin 0 tai suurempi kuin 120, niin:

Älä unohda ehdon jälkeistä sisennystä. Yleensä koodausohjelma tekee sen tosin automaattisesti!

ESIMERKKI 11: Salaista tietoa

```
password=”PikkolanKoulu”
```

Määritellään muuttuja **password** ja tallennetaan siihen merkkijono *PikkolanKoulu.*

```
print(”Tämä ohjelma sisältää salaista tietoa. Jotta pääset tietoon
käsiksi, sinun on tiedettävä järjestelmän salasana!”)
#Ohjelma kysyy käyttäjältä salasanaa
salasana=input(”Anna salasana salaisen tiedon näkemiseksi: ”)
#salasanan vertailu
if salasana==password:
    print(”Tämä on huippusalaista tietoa! Shhhh…”)
else:
    print(”Antamasi salasana on väärin. Salainen tieto ei kuulu
    sinulle!”)
```

Jos käyttäjän antama **salasana** on täsmälleen sama kuin muuttujan **password** sisältämä merkkijono, tulostuu salainen tieto näkyviin näytölle.

Muussa tapauksessa käyttäjälle tulostuu tieto, että **salasana** oli väärä.

Tee tehtävä 38

36. Tee ohjelma, joka kysyy käyttäjältä tämän ikää (kokonaisluku) ja iän perusteella ohjelma kertoo käyttäjälle, onko hän alaikäinen vai täysi-ikäinen.

---

37. Tee ohjelma, joka kysyy käyttäjältä lukua 1 ja lukua 2 ja tallentaa ne sopiviin muuttujiin. Tämän jälkeen ohjelma tekee lukujen suuruusvertailun ja tulostaa näytölle tekstin, joka kertoo kumpi luvuista on suurempi. Ohjelma huomioi, myös mikäli luvut ovat yhtä suuria.

vinkki: `if luku1 < luku2:`
`print(luku2, ”on suurempi kuin”, luku1)`

---

38. Tee ohjelma, joka kysyy käyttäjältä mitä koulua hän käy. Jos käyttäjä vastaa *Pikkola*, tulostaa ohjelma näytölle tekstin *Käyt Suomen parasta yläkoulua!* Jos käyttäjä vastaa *Pitkäjärvi* tai *Sariola* tai *Vatiala*, tulostaa ohjelma näytölle tekstin *Käyt hyvää yläkoulua!* Kaikissa muissa tapauksissa ohjelma tulostaa näytölle tekstin *Harmi, että et opiskele Kangasalan yläkouluissa!*

LISÄÄ MATEMAATIIKKAA OHJELMIIN

Lisää uusia laskutoimituksia (esimerkiksi neliöjuuri) saa käyttöön lisäämällä ohjelman alkuun komennon **import math**.

Tämän jälkeen ohjelmassa voi käyttää monia uusia laskutoimituksia, joiden käyttö ei ole mahdollista ilman edellä mainittua komentoa.

Lisää käyttökelpoista matematiikkaa:

- neliöjuuri (esim. luvusta 9), **math.sqrt(9)**
- pii, **math.pi**
- paljon muita, joita löytää tarvittaessa internetistä.

ESIMERKKI 12: Neliöjuurilaskuri

```
import math
```

Neliöjuuriominaisuutta varten tarvitaan ohjelmaan matematiikka-kirjasto käyttöön. Se saadaan laittamalla ohjelman alkuun komento **import math**.

```
print(”Tämä ohjelma tulostaa neliöjuuren halutusta luvusta!”)
luku=float(”Anna luku, josta haluat laskea neliöjuuren: ”)
print(”Luvun”, luku, ”neliöjuuri on”, math.sqrt(luku))
```

Neliöjuuri saadaan komennolla **maths.sqrt()**, jossa sulkuihin tulee juurrettava luku.

Tee tehtävät 39-44

- Olisi hyvä, että jokaisen ohjelman alussa ohjelma antaisi käyttäjälle tiedon, mitä kyseinen ohjelma tekee, esim.

```
print(”Tämä ohjelma laskee suuri kolmion kolmas kulma on kun kaksi muuta kulmaa tiedetään.”)
```

---

39. Kulmalaskuri

Tee ohjelma, joka kysyy ensin käyttäjältä kolmion kahden kulman suuruudet ja tallentaa ne sitten sopiviin muuttujiin. Tämän jälkeen ohjelma laskee kolmion kolmannen kulman suuruuden ja tulostaa tiedon näytölle. Mikäli kolmannen kulman suuruudeksi tulisi kuitenkin negatiivinen luku, antaa ohjelma käyttäjälle virheilmoituksen virheellisesti syötetyistä kulmista.

---

40. Kolmioluokittelija

Tee ohjelma, joka kysyy ensin käyttäjältä kolmion suurimman kulman suuruutta ja tallentaa sen sitten sopivaan muuttujaan. Ohjelma antaa tämän jälkeen käyttäjälle tiedon, onko kyseinen kolmio teräväkulmainen, suorakulmainen vai tylppäkulmainen kolmio.

---

41. Pythagoraan lause

Tee ohjelma, joka kysyy käyttäjältä, haluaako käyttäjä laskea hypotenuusan pituuden, kun molempien kateettien pituudet tiedetään (vaihtoehto 1) vai haluaako käyttäjä laskea suorakulmaisen kolmion kateetin pituuden, kun toisen kateetin ja hypotenuusan pituus tiedetään (vaihtoehto 2).

Tämän jälkeen ohjelma kysyy vaihtoehtoon perustuen tarvittavat tiedot ja tulostaa näytölle kysytyn pituuden (hypotenuusa tai kateetti).

Lisätehtävä: Osaatko muokata tehtävää siten, että ohjelma antaa virheilmoituksen, mikäli käyttäjän antamat syötteet ovat virheellisiä?

---

42. Ympyrä

Tee ohjelma, joka kysyy käyttäjältä, haluaako käyttäjä laskea ympyrän pinta-alan vai ympyrän kehän pituuden. Tämän jälkeen ohjelma kysyy käyttäjältä ympyrän sädettä (yksikössä cm) ja laskee käyttäjälle kysytyn suureen sekä tulostaa tiedon näytölle.

Lisätehtävä: Osaatko muokata tehtävää siten, että ohjelma antaa virheilmoituksen, mikäli käyttäjän antamat syötteet ovat virheellisiä?

---

43. Ympyrä 2

Täydennä edellistä ohjelmaa siten, että se laskee käyttäjälle myös ympyrän kaaren pituuden tai sektorin pinta-alan, kun käyttäjä antaa ohjelmalle syötteenä kaarta vastaavan keskuskulman suuruuden.

---

44. Lisätehtävä

Taksimatkan hinta määräytyy aloitusmaksun, matkan pituuden ja matkustajamäärän mukaan. Tee laskuri, joka laskee taksimatkan hinnan. Käytä apuna seuraavaa Tampereen taksin hinnastoa:

- Matkataksa 1-4 matkustajaa 1,84 €/km TAI 5-8 matkustajaa 2,25€/km
- Aloitusmaksu 5,3€ (arkisin kello 06:00-20:00, sekä lauantaisin ja juhlapyhien aattoina kello 06:00-16:00) TAI 8,08€ (muina aikoina).

Ohjelmointitehtäviä 9. luokalle

Oppitunnit 1-3: Kertausta

Tee aluksi 8. luokan ohjelmointiasioita syventävät tehtävät 45-49

Tehtäväosio 7: Avaruusgeometriaa

Olisi hyvä, että jokaisen ohjelman alussa ohjelma antaisi käyttäjälle tiedon, mitä kyseinen ohjelma tekee, esim.

```
print(”Tämä ohjelma laskee suorakulmaisen särmiön tilavuuden.”)
```

---

45. Suorakulmaisen särmiön tilavuus

Tee ohjelma, joka laskee suorakulmaisen särmiön pohjan pinta-alan ja edelleen myös kyseisen kappaleen tilavuuden yksikössä cm^3. Aluksi ohjelma kysyy käyttäjältä suorakulmaisen särmiön pituutta ja leveyttä senttimetreinä ja tallentaa kysytyt tiedot muuttujiin **pituus** ja **leveys**. Tämän jälkeen ohjelma kysyy käyttäjältä suorakulmaisen särmiön korkeutta senttimetreinä ja tallentaa kysytyn tiedon muuttujaan **korkeus**. Lopuksi ohjelma tulostaa näytölle tiedon suorakulmaisen särmiön pohjan pinta-alasta ja tilavuudesta.

vinkki:

```
pituus=float(input(”Anna suorakulmaisen särmiön pituus senttimetreinä: ”))

print(”Pohjan pinta-ala on”, pituus*leveys, ”cm2”)
```

---

46. Yksikkömuunnos

Tee ohjelma, joka kysyy käyttäjältä tilavuutta yksikössä cm^3 ja tämän jälkeen ohjelma tulostaa näytölle, kuinka paljon kyseinen tilavuus on yksiköissä mm^3, cm^3, dm^3 ja m^3.

Osaatko muuttaa ohjelmaa ehtorakenteen avulla siten, että se kysyisikin aluksi missä yksikössä haluat antaa ohjelmalle muutettavan tilavuuden ja sen jälkeen ohjelma tulostaisi näytölle kyseisen tilavuuden muissa yksiköissä (mm^3, cm^3, dm^3 ja m^3)

vinkki:

```
tilavuus=float(input(”Anna tilavuuden lukuarvo, jonka haluat muuttaa muihin yksiköihin. \n”))

yksikko=input(”Anna myos kyseisen tilavuuden yksikko (mm3, cm3, dm3 vai m3?): \n”)

if yksikko==”mm3”:

    print(tilavuus, ”on yksikössä mm3”, tilavuus)

    print(tilavuus, ”on yksikössä cm3”, tilavuus/1000)

    print(tilavuus, ”on yksikössä dm3”, tilavuus/1000000)
```

```
    print(tilavuus, ”on yksikössä m3”, tilavuus/1000000000)
elif yksikko ==”cm3”:
    print(tilavuus, ”on yksikössä mm3”, tilavuus*1000)
    print(tilavuus, ”on yksikössä cm3”, tilavuus)
    JNE…
    JNE…
```

---

47. Pallo

Tee ohjelma, joka kysyy aluksi käyttäjältä, haluaako tämä laskea pallon tilavuuden vai pinta-alan. Tämän jälkeen ohjelma kysyy käyttäjältä pallon sädettä ja laskee kyseisen pallon tilavuuden tai pinta-alan, riippuen kumman laskun käyttäjä on ohjelmalta pyytänyt.

---

48. Kartion tilavuus

Tee ohjelma, joka kysyy aluksi käyttäjältä, haluaako käyttäjä laskea suoran ympyräkartion vai säännöllisen nelisivuisen pyramidin tilavuuden. Tämän jälkeen ohjelma kysyy laskuun tarvittavat tiedot (pohjan säde + kartion korkeus TAI pohjan sivun pituus + pyramidin korkeus) ja tulostaa näytölle kyseisen kartion tilavuuden.

vinkki:

```
import math

kartio_tyyppi=input(”Haluatko laskea ympyräkartion (A) vai nelisivuisen
pyramidin (B) tilavuuden? \n”)

if kartio_tyyppi==”A”:
      r=float(input(”Anna pohjaympyrän säde (cm): ”))
      h=float(input(”Anna kartion korkeus (cm):  ”))
      pohja=math.pi*r**2
      tilavuus=(pohja*h)/3
      print(”Ympyräkartion tilavuus on”, tilavuus, ”cm3”)

 elif kartio_tyyppi==”B”:

      …

 else:

     print(”Et antanut oikeaa vastausta A tai B!”)
```

---

49. Lieriön pinta-ala

Tee ohjelma, joka laskee ympyrälieriön tai suorakulmaisen särmiön kokonaispinta-alan. Aluksi ohjelma kysyy käyttäjältä, onko lieriön pohja ympyrä vai suorakulmio. Mikäli käyttäjä vastaa ympyrä, kysyy ohjelma käyttäjältä pohjaympyrän sädettä ja lieriön korkeutta ja tekee sen jälkeen laskutoimitukset ja tulostaa kokonaispinta-alan näytölle. Mikäli käyttäjä vastaa suorakulmio, kysyy ohjelma käyttäjältä pohjan pituuden ja leveyden sekä korkeuden ja laskee näiden tietojen avulla tarvittavat laskutoimitukset sekä tulostaa kokonaispinta-alan näytölle.

Oppitunnit 4-6: Silmukkarakenteet

WHILE-SILMUKKA

while-silmukkarakenteen avulla voidaan toistaa ohjelmaa niin kauan kuin silmukan jatkamisehto on voimassa. Jatkamisehtoa tarkistetaan jokaista - myös ensimmäistä - kierrosta ennen. Välttämättä silmukkaa ei siis suoriteta kertaakaan!

RAKENNE:

```
while silmukan jatkamisehto kirjoitetaan tähän:
    tähän kirjoitetaan, mitä ohjelma tekee, jos ehto on TOSI (TRUE)
```

Silmukkarakenne päättyy, kun sisennys päättyy. Tämän jälkeen sisennetty koodinpätkä toistetaan niin monta kertaa kuin silmukan jatkamisehto on voimassa.

Ilman sisennystä rakenne ei toimi oikein. Älä unohda myöskään jatkamisehdon jälkeistä kaksoispistettä!

ESIMERKKI 13: Salasanakysely

```
salasana=””
```

Alustetaan muuttuja **salasana** tyhjäksi merkkijonoksi.

```
#Toistorakenteen alku: verrataan, onko salasana koodaaminen vai ei
while salasana != ”koodaaminen”:
```

Jos **salasana** ei ole koodaaminen, etenee ohjelma silmukkarakenteeseen.

```
    #Ohjelma kysyy käyttäjältä joka kierroksella salasanaa
    salasana=input(”Anna oikea salasana, jos haluat eteenpäin! \n”)
```

Silmukkarakenteen ainoa sisältö on salasanan kysyminen, jonka jälkeen silmukan jatkamisehtoa päädytään tarkastelemaan uudelleen.

```
print(”Annoit oikean salasanana, joten pääsit eteenpäin!”)
```

Silmukkarakenteen toistaminen jatkuu niin kauan, kunne silmukan jatkamisehtoa ei enää täytetä. Jos siis muuttujassa **salasana** on merkkijono *koodaaminen*, ohjelma ei etene silmukkaan vaan tulostaa käyttäjälle tiedon oikeasta salasanasta.

ESIMERKKI 14: Positiivisten kokonaislukujen laskuri

```
#Alustetaan laskurin arvo nollaksi
laskuri=0
```

Muuttuja **laskuri** laskee, kuinka monta positiivista kokonaislukua käyttäjä syöttää peräkkäin. Aluksi muuttuja on alustettu nollaksi.

```
#Ohjelma kysyy käyttäjältä ensimmäistä kokonaislukua
luku=int(input(”Anna ensimmäinen kokonaisluku: ”))
#Toistorakenteen alku ja toiston ehto
while luku>0:
    laskuri=+1
```

Jos **luku** on suurempi kuin nolla, suorittaa ohjelma silmukkarakenteen, jossa laskuriin lisätään 1 ja kysytään uudelleen lukua. Jos **luku** on edelleen suurempi kuin nolla, toistetaan silmukkarakenne ja laskuri kasvaa yhdellä. Näin jatkuu, kunnes käyttäjän antama **luku** ei ole suurempi kuin nolla.

```
    #Ohjelma kysyy käyttäjältä joka kierroksella uutta kokonaislukua
    luku=int(input(”Anna seuraava kokonaisluku: ”))
print(”Annoit yhteensä”, laskuri, ”positiivista kokonaislukua!”)
```

Kun silmukan jatkamisehto (luku > 0) ei enää toteudu, ohjelma poistuu silmukkarakenteesta ja tulostaa näytölle tiedon, kuinka monta positiivista kokonaislukua käyttäjä syötti ohjelmalle peräkkäin (muuttujassa **laskuri** oleva luku).

LOPUTON SILMUKKA

while-silmukkarakenteen avulla voidaan tehdä myös loputon silmukkarakenne määrittämällä silmukan toistoehdoksi **true.** Tällöin silmukka keskeytyy ainoastaan törmätessään komentoon **break**.

ESIMERKKI 15: Suosikki kouluaine?

```
while True:
    aine=input("Mikä on sinun lempiaineesi koulussa? \n")
    if aine == ”matematiikka”:
        break
print("Olet oppinut läksysi!")
```

Silmukkaa toistetaan niin pitkään, kunnes ohjelma kohtaa komennon **break.**

Komentoon **break** törmätään vain, mikäli käyttäjä vastaa lempiaineekseen *matematiikka*.

Silmukan toisto lakkaa **break**-komennosta ja ohjelma antaa käyttäjälle viestin.

Tee tehtävät 50-52

FOR-SILMUKKA

for-silmukkarakenteen avulla voidaan toistaa ohjelmaa ennalta määrätty toistomäärä.

RAKENNE:

```
for i in range(…):
    tähän kirjoitetaan, mitä ohjelma tekee
```

Silmukkarakenne päättyy, kun sisennys päättyy. Tämän jälkeen sisennetty koodinpätkä toistetaan niin monta kertaa kuin toistomääräksi on asetettu.

Ilman sisennystä rakenne ei toimi oikein. Älä unohda myöskään jatkamisehdon jälkeistä kaksoispistettä!

ESIMERKKI 16: Luvun neliöt eli toinen potenssi

```
for i in range(10):
    print(”Luvun”, i, ”neliö eli toinen potenssi on ”, i * i)
print(”Ohjelman suoritus päättyy, silmukka on suoritettu 10 kertaa”)
```

Silmukka toistetaan 10 kertaa. Muuttuja i numerointi alkaa nollasta.

FOR-SILMUKAN TOISTOVÄLI

Silmukkarakenteen toistuminen määritellään **range**-komennon avulla. Lukuväliin liittyy kolme parametria: mistä luvusta silmukan toisto alkaa, minkä luvun kohdalla silmukka päättyy ja kuinka paljon luku kasvaa silmukan kierroksen jälkeen. Oletuksena silmukka alkaa nollasta ja luku kasvaa yhdellä kierroksen jälkeen.

Merkintä	Lukuväli
range(10)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
range(5, 15)	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
range(-6, 3)	-6, -5, -4, -3, -2, -1, 0, 1, 2
range(24, 16, -1)	24, 23, 22, 21, 20, 19, 18, 17
range(3, 20, 2)	3, 5, 7, 9, 11, 13, 15, 17, 19
range(35, 4, -4)	35, 31, 27, 23, 19, 15, 11, 7

ESIMERKKI 17: Lukujen 100…0 neliöjuuret

```
import math
for i in range(100,0):
    print(”Luvun”, i, ”neliöjuuri eli toinen potenssi on ”, sqrt(i))
```

Tee tehtävät 53-57

Tehtäväosio 8: Perustehtäviä

50. Tee ohjelma, joka kysyy Suomen parhaan peruskoulun nimeä niin pitkään, kunnes käyttäjä vastaa kysymykseen Pikkola. Tämän jälkeen käyttäjä saa viestin: *Hienoa, tiesit oikean vastauksen!*

---

51. Tee ohjelma, joka pyytää käyttäjältä lukuja ja laskee niiden summan niin kauan, kunnes käyttäjän antama luku on 0.

---

52. Muokkaa jokin tehtävistä 45-49 while-rakenteen avulla siten, että ohjelma kysyy ohjelman päätyttyä haluaako käyttäjä suorittaa ohjelman uudelleen ja mikäli käyttäjä vastaa *kyllä*, ohjelma toistuu uudelleen. Muussa tapauksessa ohjelma ei toistu uudelleen.

---

53. Kirjoita ohjelma, joka tulostaa kokonaisluvut, alkaen luvusta 9, korkeintaan lukuun 200 saakka, viiden välein.

---

54. Tulosta kaikki kaksinumeroiset seitsemällä jaolliset luvut. Käytä apuna for-silmukkaa ja range-komentoa.

---

55. Laske lukujen 5, 10, 15, 20, … summa. Lukuja on summassa kaikkiaan 120 kappaletta.

---

56. Tee ohjelma, joka pyytää käyttäjältä syötteeksi positiivisen kokonaisluvun ja tulostaa peräkkäisten kokonaislukujen summan kyseiseen lukuun saakka

---

57. Laske kaksinumeroisten kolmella jaollisten lukujen tulo käyttäen apuna for- ja range-komentoja.

Oppitunnit 7-9: Aliohjelmat ja funktiot

ALIOHJELMAT, def()

Jakamalla ohjelmat pienempiin osiin, saadaan ohjelmista helppolukuisempia, yleiskäyttöisempiä ja helpommin ylläpidettäviä. Tällöin ohjelmassa on pääohjelman lisäksi pienempiä aliohjelmia. Pääohjelmaan sisältyy tällöin aliohjelmakutsuja eli pääohjelma käyttää hyödyksi valmiiksi määriteltyjä pienempiä aliohjelmia.

Aliohjelmat kirjoitetaan mahdollisimman yleiskäyttöisiksi, jotta samaa aliohjelmaa voidaan hyödyntää kaikkiin samantyyppisiin tehtäviin ohjelman sisällä. Tällöin samanlaista algoritmia ei tarvitse kirjoittaa montaa kertaa uudelleen, vaan voidaan käyttää samaa algoritmia useassa eri kohdassa. Aliohjelmista saadaan yleiskäyttöisiä parametrien eli muuttujien avulla.

Yleensä aliohjelmat on tapana kirjoittaa ohjelman alkuun eli koodin yläosaan.

RAKENNE:

```
def aliohjelman nimi (mahdolliset muuttujat tulevat pilkuilla
eroteltuna sulkeisiin):
    tähän kirjoitetaan, mitä aliohjelma tekee
```

ESIMERKKI 18: Tarinantoistaja

```
def tarinankertoja():
    print("Tämä tarina on todella pitkä ja toistuu ohjelmassa monta eri
    kertaa eri tilanteissa. Siksi tämä tarina kannattaa kirjoittaa
    aliohjelmaksi, jotta ohjelmakoodi on helpommin luettavissa ja aina,
    kun tämä tarina halutaan näkyville, riittää, että kutsutaan tätä
    tarinaa pääohjelmassa.")

#Tästä alkaa pääohjelma. Aliohjelma on määritelty ohjelman alkuun.


tarinankertoja()

print("Hassu tarina, ja nyt se tulee uudelleen! \n")

tarinankertoja()

print("Ja vielä kerta kiellonpäälle! \n")

tarinankertoja()
```

Määritellään aliohjelma, jonka nimi on **tarinankertoja**

Kutsutaan aliohjelmaa. Tällöin ohjelma liittää tähän kohtaa koodin, mitä aliohjelmassa **tarinankertoja** on määritelty.

Kutsutaan aliohjelmaa uudelleen. Koodi pysyy lyhyempänä, kun samaa asiaa ei tarvitse kirjoittaa montaa kertaa.

ALIOHJELMAT, def()

Mikäli halutaan, että aliohjelma toistuu aina hivenen eri tavalla, tulee sille määrittää parametrit eli muuttujat. Parametrit lisätään sulkeisiin pilkulla eroteltuina. Niitä voi olla kuinka monta tahansa. On kuitenkin tärkeää huomata, että aliohjelmaa kutsuttaessa on parametrien määrän oltava sama kuin määrittelyvaiheessa on määritelty.

Aliohjelmissa käytettävät muuttujat ovat paikallisia eli ne ovat käytössä vain aliohjelman suorituksen ajan. Kun aliohjelmaa ei suoriteta, päättyy aliohjelmassa olevien muuttujien olemassaolo.

ESIMERKKI 19: Kehittynyt tarinankertoja

```
def tarinankertoja(nimi1, nimi2):
```

Määritellään aliohjelmalle kaksi parametria, **nimi1** ja **nimi2.**

```
print("Elipä kerran", nimi1 + ", jonka paras kaveri oli nimeltään", nimi2
+ ". Yhdessä he tekivät kaikkea mukavaa yhdessä. Eräänä päivänä", nimi1,
"joutui kuitenkin muuttamaan toiseen maahan.", nimi2,"tuli tästä kovin
surulliseksi.")
```

+ liittää merkkijonot yhteen ilman välilyöntiä. Pilkku lisää välilyönnin.

```
#Varsinainen pääohjelma

tarinankertoja("Kalle", "Petri")

print("Hassu tarina, vielä uudelleen! \n")

tarinankertoja("Viivi", "Toni")

print("Ja vielä kerta kiellonpäälle \n")

tarinankertoja("Julia", "Jenni")
```

Nyt tarina toistuu jokaisella kerralla hivenen eri tavalla, koska jokaisessa tarinassa **nimi1** ja **nimi2** muuttujiin on tallennettu eri nimet.

ESIMERKKI 20: Kolmen luvun keskiarvo

```
def kolmenluvunkeskiarvo (a=1, b=2, c=3):
     keskiarvo=(a+b+c)/3
     print(keskiarvo)
```

Määritellään ohjelma, joka laskee kolmen luvun keskiarvon. Parametreille voi halutessaan antaa oletusarvot, joita käytetään, jos muita arvoja ei käyttäjältä saada. Tässä oletusarvot **a**=1, **b**=2 ja **c**=3.

```
#Varsinainen pääohjelma

kolmenluvunkeskiarvo(5, 6, 7)
```

Kutsutaan aliohjelmaa, jolloin ohjelma laskee lukujen 5, 6, 7 keskiarvon.

```
kolmenluvunkeskiarvo (22, 95, 19)
```

Kutsutaan aliohjelmaa, jolloin ohjelma laskee lukujen 22, 95, 19 keskiarvon.

ESIMERKKI 21: Kehystäjä

```
def kehys(nimi):
     pituus=len(nimi)
     print("-" * pituus)
     print(nimi)
     print("-" * pituus)
```

Määritellään aliohjelma, joka tulostaa annetun nimen ylä- ja alapuolelle rivin viivoja. Viivojen lukumäärä on sama kuin nimen pituus.

```
#Alustetaan muuttuja tyhjäksi
nimi = ""

#Silmukkarakenne
while nimi != "lopeta":
```

Jos nimi ei ole ”*lopeta*”, jatkaa ohjelma nimen kyselyä uudelleen ja uudelleen.

```
     nimi=input("Minkä nimen haluat tulostaa kehystettynä?")
     kehys(nimi)
```

Kutsutaan aliohjelmaa.

Tee tehtävät 58-60

FUNKTIOT, def()

Aliohjelmaa, joka palauttaa jonkin arvon aliohjelman suorituksen tuloksena, kutsutaan funktioksi. Funktion arvo palautetaan return-komennolla. Funktio palauttaa arvon pääohjelmaan funktiokutsun paikalle.

RAKENNE:

```
def aliohjelman nimi (mahdolliset muuttujat tulevat pilkuilla
eroteltuna sulkeisiin):
    tähän kirjoitetaan, mitä funktio tekee
    return tähän kirjoitetaan, mitä funktio palauttaa pääohjelmaan
```

ESIMERKKI 22: Tiheyslaskuri

```
def tiheyslaskuri (tilavuus, massa):
    tiheys=massa/tilavuus
    return tiheys
```

Määritellään funktio. Kun funktiota kutsutaan, palauttaa se kutsun paikalle vain **return**-komennon perässä olevan arvon eli tässä esimerkissä tiheyden.

```
#Pääohjelma

massa=float(input("Anna kappaleen massa kilogrammoina: "))
tilavuus=float(input("Anna kappaleen tiheys yksikössä dm3: "))

if tiheyslaskuri (tilavuus, massa) >0:
    print("Tiheys on", tiheyslaskuri(tilavuus, massa), "kg/dm3")
else:
    print("virheelliset arvot, tiheys ei voi olla negatiivinen.")
```

Funktio palauttaa kutsun paikalle tiheyden arvon.

Tee tehtävät 61-69

58. Tee aliohjelman avulla ohjelma, joka tervehtii käyttäjää. Määritä aluksi aliohjelma, jonka nimi on **tervehtija** ja anna sille parametriksi eli muuttujaksi **nimi**.

Tee sitten ohjelma, joka tervehtii ainakin viittä eri henkilöä, kun kutsut aliohjelmaa ja annat sille parametrina aina eri henkilön nimen.

vinkki:

```
def tervehtija(nimi):
    print("……", nimi)

tervehtija("Kalle")
tervehtija("Henna")

jne…
```

---

59. Määrittele aliohjelma **matematiikankeskiarvo**, joka laskee todistuksen matematiikan arvosanojen keskiarvon yläkoulun ajalta (7. syksy, 7. kevät, 8. syksy, 8. kevät, 9. syksy ja 9. kevät). Laske aliohjelman avulla ainakin neljän eri henkilön matematiikan todistusarvosanojen keskiarvo.

---

60. Tee aliohjelma, joka tulostaa näytölle jonkinlaisen kehyksen, jonka olet laatinut näppäimistön erikoismerkkejä laatien. Kehyksen keskelle aliohjelma tulostaa myös sille parametrina annetun tekstin.

Tulosta aliohjelman avulla ainakin viisi erilaista kehystettyä tervehdystä. Käytä silmukkarakennetta esimerkin 19 mukaisesti

---

61. Tee funktio, joka laskee kuntosalijäsenyyden hinnan, kun sille annetaan parametreina jäsenyyden aloitusmaksun suuruus, kuukausimaksun suuruus ja jäsenyyden pituus kuukausina.

Tee ohjelmaan silmukkarakenne siten, että ohjelma laskee kuntosalijäsenyyden hinnan niin pitkään, kunnes käyttäjä antaa kaikkiin edellä mainittuihin muuttujiin arvon 0.

vinkki:

```
def kuntosali(aloitusmaksu, kuukausimaara, kuukausimaksu):
    hinta = aloitusmaksu + kuukausimaara*kuukausimaksu
    return hinta
```

62. Määrittele funktiot $f(x) = 5x + 4$, $g(x) = 3x + 19$ ja $h(x) = -0{,}3x - 4$. Tee ohjelma, joka laskee käyttäjän valitseman funktion arvon käyttäjän antamalla muuttujan **x** arvolla. Hyödynnä tehtävässä ehtorakennetta.

vinkki:

```
def f(x):
    arvo = 5*x+4
    return arvo
```

---

63. Myyntifirma maksaa työntekijälleen peruspalkkaa 8,5 €/h ja lisäksi 10 % toteutuneiden kauppojen myynnin arvosta. Muodosta funktio, joka laskee työntekijän palkan suuruuden, kun funktiolle annetaan parametreina työpäivän pituus tunteina ja päivän myynnin arvo.

---

64. Poimit herkkutatteja ja myyt ne sienitukkurille. Tukkuri maksaa sienistä 4€/kg. Sienimatkasi kulut ovat 12 €. Muodosta funktio, joka kuvaa montako euroa ansaitset sienien poiminnalla. Tee ohjelma, joka laskee sienimyynnin tuottosi, kun ohjelmalle annetaan parametrina poimimasi sienimäärä kilogrammoina.

Tee ohjelmaan myös virheentarkistus eli jos annat ohjelmalle negatiivisen kilogrammamäärän, ilmoittaa ohjelma virheellisestä syötteestä.

Osaatko tehdä silmukkarakenteen avulla ohjelmasta toistuvan?

---

65. Tee tehtävä 47, 48 tai 49 uudelleen funktioita apuna hyödyntäen.

---

66. Funktio $f(x) = 12x + 16$ ilmaisee veden lämpötilan kattilassa, kun x on lämmitysaika minuutteina. Muodosta ohjelma, jonka avulla voit laskea veden lämpötilan kattilassa. Mikäli veden lämpötila olisi 100 astetta tai enemmän, antaa ohjelma tulokseksi ”*vesi kiehuu*”.

---

67. Laske funktion $f(x) = 3x - 2$ arvot muuttujan **x** arvoilla 0…100. Hyödynnä ohjelmassa funktiota ja silmukkarakennetta.

---

68. Tee ohjelma, joka muuttaa celsiusasteet fahrenheiteiksi ja päinvastoin. Etsi aluksi netistä laskukaava muutokselle.

---

69. Ratkaise jokin oppikirjan funktio-osa-alueen tehtävistä ohjelmoimalla.

Harjoitustyö

Suunnittele ja toteuta jokin ohjelma, jossa käytät koko peruskoulun aikana oppimiasi ohjelmointitaitoja mahdollisimman kattavasti. Aihemahdollisuuksia esimerkiksi:

a) taksimatkan hinta -laskuri
b) postipaketin hinta -laskuri
c) tietokilpailupeli
d) Keksi itse!