

Python-ohjelmointi

Harjoitus 7

TAVOITTEET

- Kerrataan keskeisiä komentoja ja rakenteita harjoituksista 1–6.
- Opitaan pyöristämään round()-toiminnolla.
- Opitaan silmukkarakenne while.

Matematiikan sisällöt

Tehtävissä harjoitellaan ja syvennetään prosenttilaskennan osaamista.

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

KERTAUSTA HARJOITUKSISTA 1–6

Ensimmäiset kolme tehtävää kertaavat Säde 1 -kirjan ohjelmointiharjoitusten eli harjoituskokoelmien 1–6 keskeisiä sisältöjä. Tehtävissä tarvittavia komentoja ja rakenteita on koottu tehtävänantojen rinnalle, mutta rakenteita ja sisältöjä voi kerrata myös aikaisemmista harjoituskokoelmista. Tehtävänantojen rinnalla kerrotaan, mistä harjoituskokoelmasta tehtävässä tarvittavat sisällöt löytyvät.

TEHTÄVÄ 1

Tavoitteena on suunnitella ja tehdä ohjelma, joka kysyy käyttäjältä desimaaliluvun, jonka hän haluaa muuntaa prosenttiluvuksi. Ohjelma tulostaa lopuksi käyttäjän antaman desimaaliluvun ja siitä muunnetun prosenttiluvun.

Run

- Kirjoita ohjelman suunnitelma kommentiksi ohjelman alkuun.
- Tee ohjelma muuttujien ja komentojen avulla.
- Testaa ohjelmaa ensin kokonaisluvuilla.
- Muunna ohjelmaa niin, että se toimii oikein myös desimaaliluvuilla.

Tarvittavia komentoja:

print()
input()
int()
str()
float()

Kertaa tarvittaessa harjoituskokoelmien 1 ja 2 sisältöjä.

TEHTÄVÄ 2

Tavoitteena on tehdä ohjelma, jossa harjoitellaan toistorakenteen ja silmukkalaskurin käyttöä.

Run

- Tulosta parittomat luvut väliltä 1–100.
- Tulosta joka neljäs luku väliltä 1–100 siten, että luku 4 on ensimmäinen tulostuva luku.
- Tulosta lähtölaskenta eli laskeva lukujono luvusta 20 alkaen.

Tarvittavia komentoja:

for
range()

Kertaa tarvittaessa harjoituskokoelmien 4 ja 5 sisältöjä.

TEHTÄVÄ 3

Tavoitteena on suunnitella ja tehdä ohjelma, jolla voidaan laskea

- haluttu prosenttiosuus alkuperäisestä arvosta
- alennettu hinta, kun alkuperäinen hinta ja alennusprosentti on annettu
- korotettu hinta, kun alkuperäinen hinta ja korotusprosentti on annettu.

Run

- Tee ohjelma, joka kysyy käyttäjältä alkuperäisen arvon ja prosenttiluvun.
- Tee ohjelma, joka muuntaa prosenttiluvun desimaaliluvuksi.
- Käyttäjän pitää valita, haluaako hän selvittää prosenttiosuuden, alennettun hinnan vai korotetun hinnan.
- Lopuksi tulostuu vastauslause, josta ilmenee alkuperäinen arvo, prosenttiluku sekä loppuvastaus.

Tarvittavia komentoja:

if – elif – else

Kertaa tarvittaessa harjoituskokoelmien 1 ja 6 sisältöjä.

PYÖRISTÄMINEN

Lukuja saadaan pyöristettyä **round()**-komennolla.

Komennolle annetaan määräyksinä pyöristettävä luku sekä pyöristystarkkuus.

Esimerkki 1

Tee ohjelma, joka laskee alla olevien laskujen vastaukset ja tulostaa sekä saadun tarkan vastauksen että pyöristetyn vastauksen. Vastauksen pyöristystarkkuus on merkitty sulkeisiin.

- a) $3,5 \cdot 4,9$ (yhden desimaalin tarkkuuteen)
- b) $58 \cdot 120$ (satojen tarkkuuteen)
- c) $7,2 \cdot 1,5$ (kokonaisluvuksi)

Ratkaisu

Pyöristämiseen käytetään round-komentoa. Kirjoitetaan tehtävänannossa annettu lasku muuttujaan *alkuperäinen* ja pyöristetty arvo muuttujaan *pyöristetty*. Kohdissa a ja b pyöristystarkkuus voidaan määrittää antamalla round-komennolle pyöristystarkkuusmäärityksenä lukuarvo kertomaan halutun tarkkuuden. Lukuarvo 1 kertoo, että vastaukseen halutaan yksi desimaali. Satojen tarkkuudelle pyöristettäessä taas annetaan pyöristystarkkuudeksi -2.

Koska c-kohdan laskussa luvut ovat liukulukuja, pyöristystarkkuus 0 ei anna kokonaislukua. Jotta vastaus saadaan pyöristettyä kokonaislukujen tarkkuuteen, täytyy muuttujatyypin muuntaa.

Pyöristystoiminnon vastaukselle täytyy siis tehdä muuttujatyypin muunnos liukuluvusta kokonaisluvuksi int()-komennolla.

Pyöristyskomento

`round(luku, pyöristystarkkuus)`

Esimerkkejä pyöristystarkkuudesta:

- 1 yhden desimaalin tarkkuus
- 2 kahden desimaalin tarkkuus
- 2 satojen tarkkuus

Esimerkin 1 ratkaisun koodi:

```
#kohta a
alkuperainen = 3.5 * 4.9
pyoristetty = round(alkuperainen, 1)
print str(alkuperainen)+ u'≈'+ str( pyoristetty)
# kohta b
alkuperainen = 58 * 120
pyoristetty = round(alkuperainen, -2)
print str(alkuperainen)+ u'≈'+ str( pyoristetty)
# kohta c
alkuperainen = 7.2 * 1.5
pyoristetty = int(round(alkuperainen))
print str(alkuperainen)+ u'≈'+ str( pyoristetty)
```

TEHTÄVÄ 4

Tavoitteena on tehdä ohjelma, jossa harjoitellaan pyöristyskomennon käyttöä.

Run

Pyöristä luku 1 569 823,2564

- yhden desimaalin tarkkuuteen
- kokonaisluvuksi
- satojen tarkkuuteen
- miljoonien tarkkuuteen.

TEHTÄVÄ 5

Tavoitteena on tehdä ohjelma, jossa harjoitellaan muutosprosentin laskemista ja pyöristämiskomennon käyttöä.

Run

Tee ohjelma, joka laskee muutosprosentin, kun ohjelmalle annetaan alkuperäinen arvo ja lopullinen arvo.

- Ohjelma pyytää käyttäjältä luvut ja muuttaa ne liukuluvuiksi.
- Ohjelma laskee muutosprosentin.
- Ohjelma tutkii, onko muutosprosentti positiivinen vai negatiivinen. Sen perusteella ohjelma tekee johtopäätöksen, onko kyseessä korotus tai alennus.
- Ohjelma tulostaa joko lauseen

Muutos on _____, kyseessä on korotus.

tai lauseen

Muutos on _____, kyseessä on alennus.

Muutosprosentti ilmoitetaan kokonaislukujen tarkkuudella.

TOISTORAKENNE WHILE

Aikaisemmin harjoituksissa on tutustuttu toistorakenteeseen for, jossa toistokierrosten määrä määritellään range()-komennon avulla. Python-kielen toinen toistorakenne on **while**-rakenne. While-silmukka toistuu niin kauan kuin silmukan silmukkaehto on tosi. Jotta silmukan suorittaminen alkaa, pitää ehdon olla alussa tosi.

Toistorakenne while

while *silmukkaehto*:
suoritettavat toiminnot

Esimerkki 2

Tee ohjelma, jolla tulostat kolmella jaolliset luvut välillä 0–30 käyttäen

- a) for-rakennetta
- b) while-rakennetta.

Nollaa ei tulosteta jonon alkuun.

Ratkaisu

Kolmella jaollisuuden voi tehdä joko hyppäämällä joka kolmanteen lukuun tai käyttämällä modulusta eli jakojäännöstä. Tässä esimerkissä käytetään joka kolmanteen lukuun hyppäämistä.

- a) For-rakenne tarvitsee toistokierrosten laskemiseen range()-komentoa. Range-komennolla määritetään tarkastelualue. Tarkasteluväli on 0–31 eli lopetuskohdaksi pitää määrittää *haluttu viimeinen luku* + 1. Jotta nolla ei tulostu lukujonon alkuun, pitää tarkastelualue aloittaa ensimmäisestä tulostuvasta luvusta eli luvusta 3. Tarkasteluvälin komennoksi tulee range (3, 31, 3).

```
for i in range (3, 31, 3):
    print i,
```

Luvut 3, 6, 9, 12, 15, 18, 21, 24, 27 ja 30 tulostuvat yhdelle riville, kun tulostuskomentorivin loppuun lisää pilkkun. Ilman pilkkua jokainen luku tulostuisi omalle rivilleen.

- b) While-rakenteessa määritetään luku alussa nollassi. Silmukkaa toistetaan niin kauan kuin luku on pienempi kuin 30. Silmukassa lukua kasvatetaan aluksi kolmella ja sitten vasta tulostetaan luku.

```
a = 0
while a < 30:
    a = a + 3
    print a,
```

Koska tulostuskomennon perässä on pilkku, luvut 3, 6, 9, 12, 15, 18, 21, 24, 27 ja 30 tulostuvat yhdelle riville.

TEHTÄVÄ 6

Tavoitteena on tehdä ohjelma, jolla harjoitellaan silmukkarakenteen käyttöä.

Run

Tee ohjelma, joka tulostaa lähtölaskennan luvusta 10 lukuun 0 käyttäen

a) for-rakennetta

b) while-rakennetta.

SILMUKKARAKENNE JA LOPPUTULOKSEN TULOSTAMINEN

Esimerkki 3

Suunnittele ja tee ohjelma, jolla saat laskettua vuokran suuruuden viiden vuoden kuluttua, jos siihen tulee joka vuosi 5,0 prosentin korotus. Vuokra on alussa 320 €. Ohjelma tulostaa vain lopputuloksen eli vuokran suuruuden viiden vuoden kuluttua, ja vuokra pyöristetään senttien tarkkuuteen.

Ratkaisu

Kirjoitetaan suunnitelma kommentteihin ja kommenttien väliin ohjelmakoodi.

```
# Määritellään muuttujat vuokra ja korotus
# Vuokra = a
# Korotusprosentti = p
a = 320
p = 5.0
# Käytetään while-silmukkaa lopullisen vuokran laskemiseen.
# Silmukkaa toistetaan viidesti, määritetään
laskurimuuttujaksi n ja sen arvoksi alussa 1.
# Silmukassa lasketaan korotuksen suuruus ja uusi vuokra.
# Laskurimuuttujaa kasvatetaan yhdellä.
n = 1
while n < 6:
    korotus = p/100 * a
    a = a + korotus
    n = n + 1

# Mikäli vuokran tulostus lisättäisiin silmukkaan,
tulostaisi se vuokran jokaisen silmukan päätteeksi.
# Kun halutaan tulostaa vain lopullinen tulos, lisätään
tulostusrivi ohjelman loppuun niin, ettei se ole silmukkaan
kuuluvassa sisennyksessä.

print round(a, 2)
```

Vuokran suuruus viiden vuoden kuluttua vuotuisella 5 prosentin korotuksella on 408,41 euroa.

TEHTÄVÄ 7

Tavoitteena on tehdä ohjelma, jolla harjoitellaan while-silmukan käyttöä.

Run

Tee arvuuttelupeli, jossa käyttäjä saa vihjeen, jonka perusteella hän yrittää keksiä oikean käsitteen. Silmukkaa toistetaan niin kauan kuin käyttäjän arvaus on eri suuri kuin oikea vastaus. Keksi itse kysymys ja vastaus.

!= = eri suuri kuin

TEHTÄVÄ 8

Tavoitteena on suunnitella ja tehdä ohjelma, jolla harjoitellaan silmukkarakenteen käyttöä.

Run

Laske, kuinka paljon tietokoneen arvo on pudonnut neljässä vuodessa, kun joka vuosi sen arvosta vähennetään 25 %. Tietokoneen ostohinta on 1 250 €. Ohjelma tulostaa tietokoneen arvon neljän vuoden kuluttua.

TEHTÄVÄ 9

Run

Tavoitteena on suunnitella ja tehdä ohjelma, jolla saadaan laskettua, kuinka paljon säästötilillä on rahaa kymmenen vuoden kuluttua, kun säästötilin vuotuinen korko on 0,25 %. Säästötilille laitetaan 4 000 €. Laskelmissa ei oteta huomioon säästötiliin kohdistuvaa lähdeveroa.

VEROTUS

Palkasta täytyy maksaa veroja. Jokainen palkansaaja saa verokortin, johon on merkitty ennakonpidätysprosentti eli kuinka paljon veroa palkasta maksetaan. Verokortissa on perusprosentti, jolla ennakonpidätys tehdään tiettyyn rahasummaan asti. Tämän jälkeen ennakonpidätys tehdään lisäprosenttia käyttäen.

Jesperin verokortissa on ilmoitettu, että 3 700 euroon saakka veroa maksetaan perusprosentin 27 % mukaan. Siitä ylimenevältä osalta ennakonpidätys tehdään lisäprosentilla 42 %. Jos Jesper saa palkkaa 4 200 €, niin 3 700 eurosta ennakonpidätys tehdään perusprosentilla 27 %. Loppuosasta eli $4\,200\text{ €} - 3\,700\text{ €} = 500\text{ €}$ ennakonpidätys tehdään lisäprosentilla 42 %.

Palkkalaskelmaan tulostuu kokonaispalkka, josta ennakonpidätys tehdään, ennakonpidätyksen suuruus ja maksettava palkka eli kokonaispalkka vähennettynä ennakonpidätyksellä. Näiden lisäksi palkasta pidätetään muitakin maksuja.

TEHTÄVÄ 10

Tavoitteena on suunnitella ja tehdä ohjelma palkkalaskelman muodostamiseksi.

Run

- Tee ohjelma, joka kysyy käyttäjältä perusprosentin, perusprosentin rajan, lisäprosentin ja kuukauden palkan.
- Ohjelman tulostaa käyttäjälle seuraavat tiedot: kokonaispalkka, ennakonpidätys ja maksettava palkka.
- Voit halutessasi selvittää internetistä, mitä muita kuluja palkasta pidätetään ja lisätä myös ne ohjelmaan. Tällöin saat selville tarkemman arvion tilille tulevan palkan suuruudesta.