

# Python-ohjelmointi

# Harjoitus 2

## TAVOITTEET

- Kerrataan tulostuskomento ja lukumuotoisen muuttujan muuttaminen merkkijonoksi.
- Opitaan jakojäännös eli modulus, vertailuoperaattorit, ehtorakenne jos, input-komento ja merkkijonon muuttaminen lukumuotoon.

## Matematiikan sisällöt

Tehtävissä harjoitellaan vertailun, jaollisuuden ja prosenttilaskennan sisältöjä.

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

## KERTAUSTA HARJOITUKSESTA 1

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen muuttujan määrittäminen, tulostuskomento ja muuttujatyypin vaihtaminen merkkijonoksi str-komennolla.

Run

- Määritä muuttujat muuttuja1 = 4 ja muuttuja2 = -6.
  - Tulosta lause: Lukujen \_\_\_\_ ja \_\_\_\_ summa on \_\_\_\_ ja tulo on \_\_\_\_.
- Viivojen kohdalle tulostuu koodisi arvot.

## MODULUS

Ohjelmointikielissä on jaollisuuden tutkimista varten modulus- eli jakojäännösoperaattori. Jakojäännös on kokonaisluku, joka jää yli, kun jakolaskun jako ei mene tasan.

Esimerkiksi  $\frac{17}{4} = 4$ , jää 1.

Kun jako menee tasan eli jaettava on jaollinen jakajalla, jakojäännös on nolla. Modulusta käytetään kuten muitakin laskutoimituksia.

### Modulus-operaattori

Modulus-operaattori kertoo kahden luvun jakojäännöksen.

Jakojäännöksen laskutoimitusmerkki on %.

### Esimerkki 1

Tulosta lukujen 6 ja 2 summa, erotus, tulo ja osamäärä. Tutki, mikä on lukujen 6 ja 2 jakojäännös.

### Ratkaisu

#### KOODI

```
muuttuja1 = 6
muuttuja2 = 2
print muuttuja1 + muuttuja2      # summa
print muuttuja1 - muuttuja2      # erotus
print muuttuja1 * muuttuja2      # tulo
print muuttuja1 / muuttuja2      # osamäärä
print muuttuja1 % muuttuja2      # jakojäännös
```

#### TULOSTE

```
8
4
12
3
0
```

**Esimerkki 2**

Tee ohjelma, jonka avulla voit tutkia, onko luku 28 jaollinen luvulla

a) 2                    b) 3.

**Ratkaisu**

a)

```
tutkittavaLuku = 28                    # Määritetään muuttuja tutkittavaLuku ja annetaan sille arvo.
jakaja = 2                            # Määritetään muuttuja jakaja ja annetaan sille arvo.
jakojaannos = tutkittavaLuku % jakaja    # Tutkitaan jaollisuutta jakojäännöksen avulla.
print jakojaannos                    # Tulostetaan jakojäännöksen arvo.
```

Ohjelman tulosteeksi saadaan **0**. Koska jakojäännös on nolla, luku 28 on jaollinen luvulla 2.

b) Vaihdetaan koodissa jakajan arvoksi 3 ja suoritetaan ohjelma.

Tulosteeksi saadaan **1**. Koska jakojäännös ei ole nolla, luku 28 ei ole jaollinen luvulla 3.

Jaollisuutta ei kannata tutkia jakolaskun avulla, kun käytetään Python-kielen versiota 2. Versio 2 antaa jakolaskun vastaukseksi sen, kuinka monta kokonaista kertaa jakaja mahtuu tutkittavaan lukuun. Esimerkiksi laskun  $28 / 3$  vastaukseksi tulostuisi 9. Yli jäävä osa eli 1 ei tulostuisi.

**TEHTÄVÄ 2**

Tavoitteena on harjoitella modulus-operaattorin käyttöä ja tutkia jaollisuutta.

Run

- Tee ohjelma, jolla voit tulostaa lukujen 48 ja 4 summan, erotuksen, tulon, osamäärän ja jakojäännöksen. Tulosta jokainen omalle rivilleen.
- Vaihda luvuiksi 52 ja 5. Mikä on jakojäännöksen arvo?
- Vaihda jakajan arvoksi nolla. Mitä huomaat? Miksi näin tapahtuu?

**TEHTÄVÄ 3**

Tavoitteena on pilkkoa ongelma osiin ja ratkaista se muuttujien ja apumuuttujien avulla.

Run

Tee ohjelma, jolla saat ratkaistua seuraavan tehtävän: Leikkiautoja on yhteensä 49 kappaletta. Kuusi lasta haluaa leikkiä autoilla yhdessä ja jokaiselle lapselle annetaan aluksi yhtä monta autoa. Kuinka monta autoa kukin lapsi saa? Kuinka monta autoa jää pois leikistä?

- Määritä muuttujat ja niiden arvot.
- Tulosta se autojen määrä, jonka jokainen lapsi saa.
- Tulosta se autojen määrä, joka jää pois leikistä.

## LOOGISET LAUSEKKEET

Loogisia lausekkeita eli totuusarvoja saadaan, kun tutkitaan, onko tietty ehto voimassa tarkasteluhetkellä.

Looginen lauseke tuottaa arvon tosi (*True*), jos vertailuehto toteutuu ja arvon epätosi (*False*), jos vertailuehto ei toteudu. Vertailuehdon kirjoittamiseen käytetään vertailuoperaattoreita.

### Vertailuoperaattorit Python-kielessä

|                             |    |
|-----------------------------|----|
| yhtäsuuri kuin              | == |
| suurempi kuin               | >  |
| suurempi tai yhtäsuuri kuin | >= |
| pienempi kuin               | <  |
| pienempi tai yhtäsuuri kuin | <= |
| erisuuri kuin               | != |

Loogisia lausekkeita voi yhdistää operaattoreilla

and ehdot yhtä aikaa voimassa

or toinen tai molemmat ehdot voimassa.

### Esimerkki 3

Tee ohjelma, jolla voit tutkia, ovatko väitteet tosia vai epätosia.

- Luku 3 on pienempi kuin luku 2.
- Luku 17 on suurempi tai yhtä suuri kuin luku 10.
- Kirjain a on suurempi kuin b.
- Luku -3 on pienempi kuin luku 0 ja suurempi kuin luku -9.

### Ratkaisu

| Looginen lauseke   | Koodi                                     | Loogisen lausekkeen arvo | Johtopäätös  |
|--|---|--------------------------|--|
| a) Luku 3 on pienempi kuin luku 2.                           | <pre>print 3 &lt; 2</pre>                 | False                    | Tutkitaan lukujen suuruusjärjestystä. Saadaan epätosi, koska 3 on suurempi kuin 2.   |
| b) Luku 17 on suurempi tai yhtä suuri kuin luku 10.          | <pre>print 17 &gt;= 10</pre>              | True                     | Saadaan tosi, koska 17 on suurempi kuin 10. Tai-sana ei vaadi yhtäsuuruuden voimassaoloa samanaikaisesti.                                |
| c) Kirjain a on suurempi kuin b.                             | <pre>print "a" &gt; "b"</pre>             | False                    | Tutkitaan kirjainten aakkosjärjestystä. Koska a tulee aakkosissa ensin, on se pienempi kuin b. Siksi vertailuehto tuottaa arvon epätosi. |
| d) Luku -3 on pienempi kuin luku 0 ja suurempi kuin luku -9. | <pre>print -3 &lt; 0 and -3 &gt; -9</pre> | True                     | Tarkastelu vastaa matematiikan tilannetta $-9 < -3 < 0$ eli antaa vertailuehdolle arvoksi tosi.  |

## TEHTÄVÄ 4

Tavoitteena on harjoitella vertailuoperaattoreiden käyttämistä ja ymmärtää, miten loogiset lausekkeet toimivat.

Run

Tutki, toteutuvatko seuraavat loogiset lausekkeet. Käytä tulostuskomentoa ja sopivaa vertailuoperaattoria.

- Onko luku 1 pienempi kuin luku 5?
- Onko  $5 \cdot 5$  suurempi kuin  $4 \cdot 6$ ?
- Onko luku  $-3$  yhtä suuri kuin luvun 3 vastaluku?
- Onko luku 10 pienempi tai yhtä suuri kuin 7?
- Onko luku 100 suurempi tai yhtä suuri kuin  $1\ 000 / 20$ ?

## TEHTÄVÄ 5

Tavoitteena on harjoitella vertailujen käyttämistä merkkijonoihin.

Run

Tutki, toteutuvatko seuraavat loogiset lausekkeet. Käytä tulostuskomentoa ja sopivaa vertailuoperaattoria. Muista kirjoittaa merkkijonot lainausmerkkeihin.

- Onko kirjain g pienempi kuin kirjain h?
- Onko kirjain s suurempi kuin kirjain t?
- Onko sana yhtä suuri kuin Sana?
- Onko HipHop eri suuri kuin Rap?
- Onko oma\_luokka yhtä suuri kuin oma\_luokka?
- Onko salAs1na yhtä suuri kuin salaS1na?

## EHTORAKENNE JOS

Yksinkertaisin ehtorakenne on jos. Tapahtuma tapahtuu, jos annettu ehto täyttyy.

Huomaa, että

- rivi, jossa on **if**, päättyy kaksoispisteeseen
- ehtoon liittyvä tapahtumarivi sisennetään.

### Ehtorakenne jos

if ehto tai vertailu:

*Tapahtuma, joka tapahtuu, kun vertailu on tosi.*

### Esimerkki 4

Tee ohjelma, jolla voit tutkia, onko tarkastettava luku parillinen vai pariton.

Jos luku on parillinen, ohjelma tulostaa lauseen: Luku \_\_\_ on parillinen.

Jos luku ei ole parillinen, ohjelma tulostaa lauseen: Luku \_\_\_ ei ole parillinen.

### Ratkaisu

Parilliset luvut ovat jaollisia kahdella, joten määritetään ohjelman jakajaksi 2.

Kommentit selittävät tehtyä ohjelmaa.

```
# Määritetään muuttujat tutkittavaLuku ja jakaja ja niille arvot.
# Valitaan tutkittavaksi luvuksi 256.
# Jakajan arvoksi määritetään 2.
tutkittavaLuku = 256
jakaja = 2
jakojaannos = tutkittavaLuku % jakaja
# Kirjoitetaan ehtorakenne tilanteelle, että luku on parillinen.
# Jakojäännös on tällöin yhtä suuri kuin nolla.
if jakojaannos == 0:
    print "Luku", tutkittavaLuku, "on parillinen."
# Kirjoitetaan ehtorakenne tilanteelle, että luku on pariton.
# Jakojäännös on tällöin eri suuri kuin nolla.
if jakojaannos != 0:
    print "Luku", tutkittavaLuku, "on pariton."
```

Muita lukuja voidaan tutkia, kun vaihdetaan tutkittavan luvun arvo.

## TEHTÄVÄ 6

Tavoitteena on harjoitella ehtolauseita ja käyttää vertailumerkkejä pienempi kuin ja suurempi kuin.

Run

- Määritä muuttuja ja sille arvoksi jokin kokonaisluku väliltä -10 ja -1 tai väliltä 1 ja 10.
- Tee ehtolause, joka tulostaa lauseen ”Luku on positiivinen.”, kun luku on suurempi kuin nolla.
- Tee ehtolause, joka tulostaa lauseen ”Luku on negatiivinen.”, kun luku on pienempi kuin nolla.
- Vaihda muuttujan arvoa ja tutki, toimiiko tekemäsi ohjelma.

## OHJELMALLE ANNETTAVAT SYÖTTEET

Tähän asti ohjelmalle annettavat syötteet eli erilaiset muuttujan arvot on kirjoitettu osaksi ohjelman koodia. Muuttujalle annettavat arvot voidaan kysyä myös käyttäjältä, jolloin ohjelmaa voi helpommin käyttää erilaisten tilanteiden tutkimiseen. Käyttäjältä kysytään tietoa **input()**-komennolla.

### Näytölle tuleva kysymys

input ( )

Sulkeiden sisään kirjoitetaan näytölle tuleva kysymys.

### Esimerkki 5

Tee ohjelma, jossa käyttäjältä kysytään jokin monivalintakysymys. Vastausvaihtoehtojen perusteella tulostuu joko lause ”Vastaus on väärin.” tai ”Hyvin tiedetty.”

### Ratkaisu

Määritetään muuttuja, joka saa arvon input()-komennolla. Annettua vastausta verrataan ehtolauseella. Saadun totuusarvon perusteella tulostuu joko ”Vastaus on väärin.” tai ”Hyvin tiedetty!”.

```
# Määritetään muuttuja vastaus, jonka arvoksi tulee käyttäjän kysymykseen antama vastaus.
vastaus = input("Valitse oikea vaihtoehto: Prosentti tarkoittaa a) tuhannesosaa, b) sadasosaa
vai c) kymmenesosaa.")
# Tehdään ehtolauseet jokaiselle vaihtoehtokirjaimelle. Vastauksena saadaan joko a, b, tai c.
# Annettua vastausta verrataan vastaavaan kirjaimeen ja oikea vastaus tulostaa "Hyvin
tiedetty!".
if vastaus == "a":
    print u"Vastaus on väärin."
if vastaus == "b":
    print u"Hyvin tiedetty!"
if vastaus == "c":
    print u"Vastaus on väärin."
```

Kun ohjelma suoritetaan, esitettävä kysymys tulee ponnahdusikkunaan. Käyttäjä kirjoittaa ponnahdusikkunaan pyydetyn tiedon. Ohjelma lukee käyttäjän syöttämän tiedon ja käyttää sitä kuten ohjelmaan kirjoitettua muuttujaa.

## TEHTÄVÄ 7

Tavoitteena on harjoitella input()-komennon käyttöä.

Run

Tee ohjelma, joka kysyy käyttäjältä nimen ja tulostaa tervehdyksen:

”Hei \_\_\_\_\_! Kaunis päivä tänään.”

- Määritä muuttuja nimi ja sille input()-komento. Input()-komennon sulkeiden sisään tulee käyttäjälle näkyvä pyyntö kertoa nimi.
- Tulosta tervehdyslause, joka sisältää käyttäjän antaman nimen.

## TEKSTIN MUUTTAMINEN LUKUMUOTOON

Input()-komennolla saadaan käyttäjältä merkkijono eli tekstimuodossa oleva tieto. Jos käyttäjältä halutaan kysyä lukua, pitää se vastauksen saamisen jälkeen muuttaa luvuksi **int()**-komennolla.

### Merkkijonon muuttaminen luvuksi

|           |                                      |
|-----------|--------------------------------------|
| int ( )   | Muuttaa merkkijonon kokonaisluvuksi. |
| float ( ) | Muuttaa merkkijonon liukuluvuksi.    |

### Esimerkki 6

Tee ohjelma, jolla voi tutkia, onko käyttäjän antaman luku jaollinen kolmella.

### Ratkaisu

Määritetään muuttuja `tutkittavaLuku`, joka saa arvon `input()`-komennolla.

Määritetään muuttuja `jakaja` ja sen arvoksi 3.

Kun lasketaan jakojäännös, saatu vastaus pitää muuttaa lukumuotoiseksi `int()`-komennolla.

```
tutkittavaLuku = input("Anna luku, jonka jaollisuutta tutkitaan.")
jakaja = 3
jakojaannos = int(tutkittavaLuku) % jakaja
print jakojaannos
```

## TEHTÄVÄ 8

Tavoitteena on harjoitella merkkijonon muuttamista desimaaliluvuksi `float()`-komennolla ja `input()`-komennon käyttöä.

Run

Tee ohjelma, jolla käyttäjän antama desimaaliluku saadaan muutettua prosenttiluvuksi.

- Määritä muuttuja *alkuperainen*. Pyydä käyttäjää antamaan desimaaliluku Python-kielen hyväksymässä muodossa eli käyttämään desimaalierottimena pistettä.
- Määritä muuttuja *prosenttiluvuksi* ja sille lausekkeeksi  $\text{alkuperainen} \cdot 100$ . Muuta käyttäjän antama tieto eli muuttuja *alkuperäinen* lukumuotoon `float()`-komennolla.
- Tulosta muuttuja *prosenttiluku* muodossa \_\_\_\_%. Viivan tilalle tulee saamasi prosenttiluku.
- Suorita ohjelma.
- Kokeile ohjelman toimintaa eri desimaaliluvuilla.

## TEHTÄVÄ 9

Tavoitteena on tehdä ohjelma, jota voi käyttää apuna Eratostheneen seulan tekemiseen. Ohjelmalla etsitään alkuluvut väliltä 1–100.

Run

Tee ohjelma, jolla voit tutkia, onko luku jaollinen luvuilla 2, 3, 5 ja 7. Tutki vain lukuja väliltä 1–100.

- Määritä muuttuja *tutkittavaLuku*. Pyydä käyttäjää antamaan kokonaisluku väliltä 1–100.
- Tee tarvittavat ehtolausekkeet jokaiselle jakajalle ja mieti, mitä ehtolauseke tulostaa, kun se on tosi. Muista muuttaa käyttäjän antama tieto lukumuotoon `int()`-komennolla.
- Suorita ohjelma.
- Vaihda tutkittavaa lukua.

## TEHTÄVÄ 10

Tavoitteena on tehdä ohjelma, jolla saadaan laskettua halutun tiedon prosenttiosuus koko joukosta.

Run

Tee ohjelma, jolla saat ratkaistua laatikon tehtävän.

- Määritä muuttuja *kysytty\_vari* ja pyydä käyttäjältä sille arvo.
- Määritä muuttuja *sukkiaYht* ja pyydä käyttäjältä sille arvo.
- Määritä muuttuja *prosenttia* ja sille lauseke `kysytty_väri / sukkia yhteensä · 100`.
- Tulosta ”Vastaus on \_\_\_ %.” Viivan tilalle tulee saamasi prosenttiluku.
- Testaa ohjelman toimintaa tehtävän eri arvoilla.

**Tehtävä:** Laatikossa on 4 punaista, 10 mustaa ja 6 sinistä sukkaa. Kuinka monta prosenttia kaikista sukista on  
a) punaisia  
b) mustia  
c) sinisiä?



## LISÄTEHTÄVIÄ

### LISÄTEHTÄVÄ 1

Tavoitteena on tehdä ohjelma, jolla voidaan laventaa ja supistaa murtolukuja.

- Suunnittele ohjelma, jolla voit supistaa ja laventaa murtolukuja. Huomaa, ettet voi käsitellä murtolukuja sellaisenaan, vaan joudut tarkastelemaan erikseen osoittajaa ja nimittäjää.
- Pyydä käyttäjää antamaan erikseen osoittaja ja erikseen nimittäjä.
- Kokeile tulostaa supistettu tai lavennettu murtoluku.