

Ohjelmoinnin peruskurssi Y1

CS-A1111

20.9.2023

Oppimistavoitteet: tämän luennon jälkeen

- ▶ Osaat kirjoittaa for-käskyn avulla ohjelman, joka toistaa haluttua käskyä tai käskyjonoa niin, että käskyjono tarvitsee kirjoittaa vain yhteen kertaan.
- ▶ Tiedät, miten voit muotoilla ohjelman tulostetta, esim. säätää tulostettavissa desimaaliluvuissa desimaalien määrää.
- ▶ Olet saanut ajatuksia siitä, miten annetun ongelman perusteella voi kirjoittaa ongelman ratkaisevan tietokoneohjelman.
- ▶ Voit luennon aikana lähettää kysymyksiä ja kommentteja sivulla <http://presemu.aalto.fi/y1syksy2023>

Kertausta: toistokäsky while

```
def main():
    print("Anna lampotiloja, lopeta -300:lla.")
    lkm = 0
    summa = 0.0
    rivi = input("Anna ensimmäinen lampotila.\n")
    lampotila = float(rivi)
    while lampotila > -300.0:
        summa = summa + lampotila
        lkm = lkm + 1
        rivi = input("Anna seuraava lampotila.\n")
        lampotila = float(rivi)
    if lkm > 0:
        keskiarvo = summa / lkm
        print("Lampotilojen keskiarvo on", keskiarvo)
    else:
        print("Et antanut yhtään lampotilaa.")
```

```
main()
```

If-käsky toistokäskyn sisällä

```
def main():
    print("Anna lampotiloja, lopeta -300:lla.")
    lkm = 0
    summa = 0.0
    hellepaivat = 0
    rivi = input("Anna ensimmäinen lampotila.\n")
    lampotila = float(rivi)
    while lampotila > -300.0:
        summa = summa + lampotila
        lkm = lkm + 1
        if lampotila >= 25.0:
            hellepaivat += 1
        rivi = input("Anna seuraava lampotila.\n")
        lampotila = float(rivi)
    print("Hellepaivia on", hellepaivat, "kpl.")
    if lkm > 0:
        print("Lampotilojen keskiarvo on", summa / lkm)
```

main()

Toistokäsky for

- ▶ For-käskyn avulla voidaan käydä jonkin rakenteen alkiot järjestyksessä.
- ▶ Funktion `range` avulla voidaan generoida lukujono (ei täsmällisesti oikea termi), jonka kaikki alkiot voidaan käydä läpi for-käskyn avulla.
- ▶ Esimerkiksi `range(11)` generoi lukujonon, joka sisältää kokonaisluvut 0:sta 10:een asti.
- ▶ Voidaan kirjoittaa toistokäsky

```
for i in range(11):  
    tee jotain arvolle i
```

jossa `i` saa vuorotellen arvot 0–10.

Esimerkki: kertotaulu for-käskyn avulla

```
def main():  
    for i in range(11):  
        tulos = i * 6  
        print(i, "* 6 =", tulos)  
  
main()
```

Lisää range-funktiosta

- ▶ Annetaan myös jonon ensimmäisen alkio:
`range(1, 11)`.
- ▶ Määrätään jonon kahden peräkkäisen alkion väli:
`range(1, 11, 2)`.
- ▶ Väli voi olla myös negatiivinen:
`range(10, 1, -2)`.

Kertotaulu uudelleen

```
def main():  
    for i in range(1, 11):  
        tulos = i * 6  
        print(i, "* 6 =", tulos)
```

```
main()
```


Hellepäivien lukumäärä for-käskyn avulla

```
def main():
    HELLERAJA = 25.0
    rivi = input("Montako lampotilaa annat?\n")
    lkm = int(rivi)
    hellepaivien_lkm = 0
    for i in range(lkm):
        rivi = input("Anna seuraava lampotila.\n")
        lampotila = float(rivi)
        if lampotila >= HELLERAJA:
            hellepaivien_lkm += 1
    print("Hellepäivia oli", hellepaivien_lkm, "kpl.")
```

```
main()
```

Sisäkkäiset toistokäskyt

- ▶ Toistokäskyn sisällä voi olla toinen toistokäsky.
- ▶ Sisempi toistokäsky suoritetaan kokonaan eli sen kaikki kierrokset suoritetaan jokaisella ulomman toistokäskyn kierroksella.
- ▶ Seuraavien kalvojen esimerkkiohjelmat tulostavat kellonaikoja (tunnit ja minuutit) yhden minuutin välein. Minuutteja muutetaan sisemmässä toistokäskyssä ja tunteja ulommassa toistokäskyssä.

Kello for-käskyn avulla

```
def main():  
    for tunnit in range(24):  
        for minuutit in range(60):  
            print(tunnit, ":", minuutit)  
  
main()
```

Kello while-käskyn avulla

```
def main():
    tunnit = 0
    while tunnit < 24:
        minuutit = 0
        while minuutit < 60:
            print(tunnit, ":", minuutit)
            minuutit += 1
        tunnit += 1

main()
```

Välitehtävä

- ▶ Montako kertaa alla oleva ohjelma tulostaa Moi!

```
def main():
    isokierros = 0
    while isokierros < 4:
        pikkukierros = isokierros
        while pikkukierros < 4:
            print("Moi!")
            pikkukierros += 1
        isokierros += 1

main()
```

- ▶ Vastaa sivulla <http://presemo.aalto.fi/y1syksy2023>

Kumpaa käyttää: while vai for?

- ▶ while kätevämpi
 - ▶ kun toistojen lukumäärään saatetaan vaikuttaa toistokäskyn sisällä.
 - ▶ "kunnes toisin sanotaan" eli esim. käyttäjä antaa lopetuskäskyn tai jokin ehto toteutuu.
- ▶ for kätevämpi
 - ▶ kun käydään läpi jonkin tietorakenteen alkioita, esim. listassa olevia arvoja tai tiedostossa olevia rivejä.
 - ▶ kun toistojen lukumäärä on ennakolta tiedossa.
- ▶ Kummallakin käskyllä voi tehdä samat asiat eivätkä yllä olevat nyrkkisäännöt ole ehdottomia.

Tulostuksen muotoilu

- ▶ Tulostuksen muotoilun avulla voi esim.
 - ▶ määrätä desimaaliluvuista tulostettavien desimaalien määrän
 - ▶ määrätä, kuinka leveä kenttä tulostettavalle arvolle varataan
 - ▶ määrätä, mihin osaan käytettävissä olevassa tilassa tulostettava arvo sijoitetaan
 - ▶ päästä eroon ylimääräisistä välilyönneistä tulostuksessa.
- ▶ Oikeasti kysymys on tulostettavan merkkijonon muotoilusta.

Tulostuksen muotoilu f-stringien avulla

- ▶ Käytössä Pythonin versiosta 3.6 lähtien
- ▶ `print`-käsytössä tulostettavan merkkijonon eteen kirjoitetaan ennen lainausmerkkiä `f`-kirjain
- ▶ Tekstin keskelle lisätään aaltosulut ja niiden sisään tulostettavan muuttujan nimi tai muu lauseke siihen kohtaan, johon tulostuksessa halutaan tulevan tämän muotoiltava arvo.
- ▶ Muuttujan nimen tai muun lausekkeen jälkeen kirjoitetaan kaksoispiste ja *muotoilumääre*, joka kertoo, miten arvo muotoillaan.
- ▶ Esimerkki:

```
luku = 4.579875
tulo = luku * luku
print(f"Luku on {luku:7.3f}")
print(f"{luku:5.2f} toiseen on {tulo:6.2f}")
```


Muotoilumääre

- ▶ Muotoilukoodi alkaa kaksoispisteellä.
- ▶ Kaksoispisteen jälkeen voidaan antaa kokonaislukuna arvolle varattavan kentän leveys (voidaan myös jättää pois).
- ▶ Tämän jälkeen desimaaliluvuilla voidaan antaa piste ja sen jälkeen käytettävä tarkkuus (desimaalien määrä) kokonaislukuna.
- ▶ Lopuksi tulee määrittelykirjain, joka kertoo, millaista arvoa ollaan tulostamassa:
 - d kokonaisluku
 - f desimaaliluku ilman eksponenttiosaa
 - e desimaaliluku eksponenttiosan kanssa
 - E desimaaliluku eksponenttiosan kanssa
 - g desimaaliluku
 - s merkkijono

Esimerkkejä

- ▶ Muuttujien arvot tulostetaan kahden desimaalin tarkkuudella:

```
print(f"{luku:.2f} toiseen on {tulo:.2f}")
```

- ▶ Tulostettaville arvoille varataan vakiolevyiset kentät, desimaaliluvut tulostetaan kolmen desimaalin tarkkuudella:

```
print(f"{vuosi:3d} {korko:7.3f} {arvo:9.3f}")
```

- ▶ Kaksoispiste tulee suoraan merkkijonon jälkeen ja piste suoraan kokonaisluvun jälkeen ilman välilyöntejä välissä:

```
print(f"Opiskelija {nimi:s}: {pisteet:d}.")
```

tai

```
print(f"Opiskelija {nimi}: {pisteet}.")
```

Tasaus sarakkeessa

- ▶ Oletuksena merkkijonoja sisältävät sarakkeet tasataan vasempaan reunaan ja lukuja sisältävät oikeaan reunaan.
- ▶ Sarakkeen tasausta voi muokata haluamallaan tavalla seuraavien merkkien avulla:

<, ^ >

- ▶ Esimerkki:

```
print(f"{suku1:>5s} {etu1:>5s} {palkka1:<5d} eur")
Puro  Esa 2800  eur
print(f"{suku2:>5s} {etu2:>5s} {palkka2:<5d} eur")
Pomo  Tiina 11000 eur
```

Vanhempi tapa: format

- ▶ `print`-käskyssä kirjoitetaan lainausmerkkien sisälle tulostettava vakioteksti.
- ▶ Tekstin keskelle lisätään aaltosulut ja niiden sisään muotoilumääre sellaiseen paikkaan, johon tulostuksessa halutaan tulevan jokin muotoiltava arvo, esimerkiksi muuttujan arvo.
- ▶ Lainausmerkin jälkeen tulostettavat arvot annetaan `format`-käskyllä.
- ▶ Esimerkki:

```
luku = 4.579875
tulo = luku * luku
print("Luku on {:.3f}".format(luku))
print("{:5.2f} toiseen on {:6.2f}".format(luku, tulo))
```

Ongelmasta sen ratkaisevaksi ohjelmaksi

Opiskelijan palautetta harjoitustehtävästä:

Tässä tehtävässä oli paljon vaikeampi miettiä, miten ratkaisen ongelman kuin itse ohjelmointi.

- ▶ Ongelman ratkaisumenetelmän miettiminen on olennainen osa ohjelmointia!

Miten lähteä etsimään ratkaisumenetelmää

- ▶ Mieti, miten ratkaisisit saman ongelman, jos sinulla olisi käytössäsi vain kynä ja paperia, ehkä lisäksi taskulaskin.
- ▶ Kirjoita ohjelma, joka tekee samat asiat — vain automaattisesti ja paljon nopeammin.

Esimerkki

- ▶ Saat paperin, johon on kirjoitettu 30 päivän lämpötilat. Sinun pitää laskea hellepäivien ja pakkaspäivien lukumäärät. Miten teet sen?
- ▶ Entä, jos lämpötilat eivät olekaan paperilla, vaan saat ne tietokoneen kuvaruudulla yhden kerrallaan? Sinulla ei ole mahdollisuutta laittaa muistiin kaikkia näkemiäsi lämpötiloja.
- ▶ Entä, jos sinulta kysytäänkin pienintä lämpötilaa?

Toinen esimerkki

- ▶ Kirjoita ohjelma, joka laskee, kuinka monta kilometriä sähkömoottoriautolla pitää ajaa, jotta se tulisi edullisemmaksi kuin bensiinipolttomoottorilla toimiva auto.
 - ▶ Ohjelma kysyy käyttäjältä autojen hankintahinnat ja bensiiniauton polttoaineen kulutuksen.
 - ▶ Oletetaan bensiinin hinnaksi 2.07 eur / l ja sähköauton kuluttaman sähkön hinnaksi 8 snt / km.
- ▶ Miten ratkaisisit saman ongelman kynällä ja paperilla?
- ▶ Tee tästä yleinen lauseke ja kirjoita se ohjelmaan.