

Ohjelmoinnin peruskurssi Y1

CS-A1111

6.9.2023

Oppimistavoitteet: tämän luennon jälkeen

- ▶ osaat kirjoittaa Python-ohjelman, joka pyytää käyttäjältä lukuja, suorittaa niille laskutoimituksia ja kertoo niiden tuloksen käyttäjälle
- ▶ tiedät, minkä tyyppisiä arvoja Python-ohjelmissa voi käsitellä ja tunnet monia niille mahdollisia operaatioita
- ▶ tiedät, mitä tarkoitetaan pääohjelmalla
- ▶ Voit luennon aikana lähettää kysymyksiä ja kommentteja sivulla <http://presemo.aalto.fi/y1syksy2023>

Kertausta: muuttujat

- ▶ Muuttujia käytetään ohjelman käsittelemien arvojen tallentamiseen ja käsittelyyn.
- ▶ Muuttujalle voi antaa arvon sijoituskäskyllä =.

```
nimi = "Matti"  
print("Hei,", nimi)  
print("Tervetuloa Python-kurssille!")
```

Esimerkkiajo

```
Hei, Matti  
Tervetuloa Python-kurssille!
```

Kertausta: arvon pyytäminen käyttäjältä

- ▶ Käyttäjän antaman arvon voi lukea `input`-käskyllä.
- ▶ Käskyn sulkujen sisään kirjoitetaan käyttäjälle annettava kehote.
- ▶ Luetun arvon voi antaa arvoksi muuttujalle sijoituskäskyllä.

```
nimi = input("Kerro nimesi: ")  
print("Hei,", nimi)  
print("Tervetuloa Python-kurssille!")
```

Ajoesimerkki

```
Kerro nimesi: Minna  
Hei, Minna  
Tervetuloa Python-kurssille!
```

Lukuarvon lukeminen käyttäjältä

- ▶ Käsky `input` välittää aina luetun arvon tekstinä, merkkijonona.
- ▶ Jotta luettua arvoa voitaisiin käsitellä lukuna, se pitää ensin muuttaa kokonais- tai desimaaliluvuksi.

- ▶ Luettua merkkijonoa vastaava kokonaisluku:

```
luku1 = int(rivi)
```

- ▶ Luettua merkkijonoa vastaava desimaaliluku:

```
luku2 = float(rivi)
```

Esimerkki: kruunulaskuri

```
print("Muutan kruunut euroiksi.")
rivi = input("Anna rahasumma kruunuina: ")
kruunut = float(rivi)
eurot = kruunut / 11.88
print("Summa on", eurot, "euroa.")
```

Esimerkkiajo

```
Muutan kruunut euroiksi.
Anna rahasumma kruunuina: 543.50
Summa on 45.74915824915825 euroa.
```

Tyypeistä

- ▶ Monissa muissa ohjelmointikielissä (esim. Java ja C) muuttujat on määriteltävä ennen käyttöä ja kerrottava, minkä tyyppisiä arvoja muuttujalle voi antaa.
- ▶ Python-ohjelmissa muuttujia ei määritellä, mutta arvolla on kuitenkin aina tyyppi.
- ▶ Kokonaisluvut: tyyppi `int`.
- ▶ Desimaaliluvut: tyyppi `float`.
- ▶ Desimaalilukuja eksponenttimerkinnän avulla: `2.22E12` tai `1.57E-31`
- ▶ `float`-tyyppiset arvot eivät ole reaalilukuja matemaattisessa mielessä.

Tyypeistä – jatkoa

- ▶ Merkkijonojen tyyppi on `str`.
 - ▶ Voidaan käyttää yksin- tai kaksinkertaisia lainausmerkkejä, esim. `"Matti Virtanen"`, `'Maija Mäkinen'`
- ▶ Tyyppi `bool` totuusarvojen esittämiseen.
 - ▶ Mahdollisia arvoja `True` ja `False`.

Sijoituskäsky

- ▶ Muuttujalle voi antaa arvon sijoituskäskyllä:

```
leveys = 5
```

```
korkeus = 6
```

```
pinta_ala = leveys * korkeus
```

- ▶ Muuttujan vanhaa arvoa voi käyttää hyväksi uutta arvoa laskettaessa:

```
luku = 5
```

```
luku = luku + 2
```

```
korkeus = 7.0
```

```
korkeus = korkeus / 2
```

Sijoituskäsky – jatkoa

- ▶ Lyhennysmerkintä muuttujan vanhan arvon käyttämiselle sijoituskäskyssä:

```
muuttuja = muuttuja + jotain
```

voidaan kirjoittaa

```
muuttuja += jotain
```

- ▶ Esimerkiksi:

```
luku = 5
```

```
luku += 1
```

- ▶ Vastaavasti toimivat -=, *= ja /=.

Välitehtävä

- ▶ Mitä seuraava ohjelma tulostaa?

```
brutto = 5000
verot = 2000
netto = brutto - verot
verot = 1000
print(netto)
```

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

Aritmeettisia laskutoimituksia

- ▶ Yleisimmin käytetyt aritmeettiset operaattorit kokonais- ja desimaaliluvuille $+$, $-$, $*$, $/$, $//$, $\%$ ja $**$
- ▶ Jakolasku
 - ▶ $/$ perinteinen
 - ▶ $//$ katkaiseva jakolasku
- ▶ Muuta
 - ▶ $\%$ jakojäännös
 - ▶ $**$ potenssiin korotus.

Funktiot ja pääohjelma

- ▶ Tyypillisesti ohjelman rakennetta selkiytetään jakamalla ohjelma funktioihin.
- ▶ Funktio on ohjelman osa, jolle on annettu oma nimi.
- ▶ Tässä vaiheessa opetellaan määrittelemään yksi erityinen funktio, `main` eli pääohjelma.

Pääohjelman määrittely

- ▶ Otsikko

```
def main():
```

- ▶ Otsikon alle kirjoitetaan pääohjelmaan kuuluvat käskyt sisennettynä, esimerkiksi

```
def main():  
    print("Muutan kruunut euroiksi.")  
    rivi = input("Anna rahasumma kruunuina: ")  
    kruunut = float(rivi)  
    eurot = kruunut / 11.88  
    print("Summa on", eurot, "euroa.")
```

- ▶ Pääohjelmaa pitää kutsua sen ulkopuolella.

```
main()
```

Esimerkki: alennettu hinta

```
def main():
    rivi = input("Anna tuotteen normaalihinta (eur): ")
    hinta = float(rivi)
    rivi = input("Anna alennus prosentteina: ")
    alennus = float(rivi)
    alennettu = (100.0 - alennus) / 100.0 * hinta
    print("Alennettu hinta on", alennettu, "euroa.")

main()
```

Rivinvaihto tulosteen perään

- ▶ Jos käyttäjälle annettavan kehoitteen perään lisätään merkki `\n`, kehote ja käyttäjän syöte tulevat eri riveille.
- ▶ Käsky `print` lisää rivinvaihdon automaattisesti.

```
def main():  
    rivi = input("Anna tuotteen normaalihinta (eur): \n")  
    hinta = float(rivi)  
    rivi = input("Anna alennus prosentteina: \n")  
    alennus = float(rivi)  
    alennettu = (100.0 - alennus) / 100.0 * hinta  
    print("Alennettu hinta on", alennettu, "euroa.")  
  
main()
```


Esimerkki: tuntinopeus

- ▶ Kirjoitetaan ohjelma, joka pyytää käyttäjältä kuljetun matkan ja siihen käytetyn ajan. Ohjelma tulostaa tuntinopeuden.

```
def main():  
    rivi = input("Anna kuljettu matka (km)\n")  
    matka = float(rivi)  
    rivi = input("Anna matkaan kulunut aika (min)\n")  
    aika = float(rivi)  
    nopeus = 60.0 / aika * matka  
    print("Nopeus oli", nopeus, "km / h")
```

```
main()
```

Toinen versio: kaksi henkilöä, jotka käyttivät eri ajan

```
def main():
    rivi = input("Anna kuljettu matka (km)\n")
    matka = float(rivi)
    rivi = input("Anna ensimmäisen käyttämä aika (min)\n")
    aika1 = float(rivi)
    nopeus1 = 60.0 / aika1 * matka
    print("Ensimmäisen nopeus oli", nopeus1, "km / h")
    rivi = input("Anna toisen käyttämä aika (min)\n")
    aika2 = float(rivi)
    nopeus2 = 60.0 / aika2 * matka
    print("Toisen nopeus oli", nopeus1, "km / h")
```

```
main()
```

Kolmas versio: käytetään samaa muuttujaa useaan kertaan

```
def main():
    rivi = input("Anna kuljettu matka (km)\n")
    matka = float(rivi)
    rivi = input("Anna ensimmäisen käyttämä aika (min)\n")
    aika = float(rivi)
    nopeus = 60.0 / aika * matka
    print("Ensimmäisen nopeus oli", nopeus, "km / h")
    rivi = input("Anna toisen käyttämä aika (min)\n")
    aika = float(rivi)
    nopeus = 60.0 / aika * matka
    print("Toisen nopeus oli", nopeus, "km / h")
```

```
main()
```

Kommentit

- ▶ Kommentti aloitetaan #-merkillä.

```
# Ohjelma ilmoittaa sekunteina annetun ajan tunteina,  
# minuutteina ja sekunteina.
```

```
def main():  
    rivi = input("Anna aikajakson pituus sekunteina.\n")  
    pituus_sekunteina = int(rivi)  
    tunnit = pituus_sekunteina // 3600  
    jaannossekunnit = pituus_sekunteina % 3600  
    minuutit = jaannossekunnit // 60  
    sekunnit = jaannossekunnit % 60  
    print("Aikajakson pituus on", tunnit, "h", minuutit,  
          "min", sekunnit, "s.")
```

```
main()
```

Toinen dokumentointimahdollisuus

- ▶ #-merkillä aloitettavien kommenttien lisäksi Pythonissa on myös toinen mahdollisuus kommentoida ohjelman kokonaisuuksia, dokumentointimerkkijono (documentation string, docstring).
- ▶ Se on lainausmerkkien sisään pantu kommentti, jota voidaan käyttää joko ohjelmatiedoston alussa tai heti funktion tai luokan otsikkoa seuraavalla rivillä.
- ▶ Python-tulkki pystyy käyttämään hyväksi dokumentointimerkkijonoja ja niiden avulla voidaan myös generoida automaattisesti ohjelman dokumentteja.