

# Ohjelmoinnin peruskurssi Y1

CS-A1111

# Ohjelmointityyli

## Oppimistavoitteet: tämän videon jälkeen

- ▶ Sinulla on peruskäsitys siitä, mitä hyvällä ohjelmointityylillä tarkoitetaan ja miksi sen käyttäminen on tärkeää.

# Hyvä ohjelmointityyli

- ▶ Miksi ihmisen pitää pystyä lukemaan valmista ohjelmakoodia?
  - ▶ virheiden korjaus
  - ▶ ohjelman laajentaminen tai muuttaminen
  - ▶ pohja uutta ohjelmaa tehdessä.
- ▶ Hyvä ohjelmointityyli auttaa ohjelman lukemisessa ja ymmärtämisessä.

## Mitä seuraava funktio tekee?

```
def f(t1,t2):
    i=0 #nollataan i
    x=0 #nollataan x
    y=len(t1) #y saa arvokseen t1:n pituuden
    while i<y: #jatketaan niin kauan, että i on y
        z=t1[i]-t2[i] #sijoitetaan erotus z:aan
        x+=z #lisataan z x:aan
        i+=1 #kasvatetaan i:ta yhdellä
    if y>0:
        k=x/y #k:n arvoksi tulee x jaettuna y
    else:
        k=0.0 #nollataan k
    return k #palautetaan k:n arvo
```

## Sama funktio kirjoitettuna toisin

```
def laske_erojen_keskiarvo(lista1, lista2):
    i = 0
    erojen_summa = 0
    alkioiden_lkm = len(lista1)
    while i < alkioiden_lkm:
        alkioiden_ero = lista1[i] - lista2[i]
        erojen_summa += alkioiden_ero
        i += 1
    if alkioiden_lkm > 0:
        keskiarvo = erojen_summa / alkioiden_lkm
    else:
        keskiarvo = 0.0
    return keskiarvo
```

- ▶ Kumpaa funktiota on helpompi ymmärtää?

# Hyvään ohjelmointityyliin kuuluvia asioita

- ▶ Ohjelman järkevä jako funktioihin.
- ▶ Kuvaavat nimet muuttujilla, parametreilla, funktioilla jne.
- ▶ Järkevä kommentointi.
- ▶ Sisennykset.
- ▶ Tyhjien tilan järkevä käyttö.

# Nimistä

- ▶ Muuttujat, parametrit ja funktiot on syytä nimetä niin, että ne kuvaavat mahdollisimman hyvin muuttujan tms. tarkoitusta.
- ▶ Esimerkiksi korkeus vs. luku.
- ▶ Yksikirjaimiset nimet ovat suositeltavia vain erikoistapauksissa:
  - ▶ Toistokäskyn kierroslaskureina.
  - ▶ Muuttujan nimi on sama kuin esim. matematiikan tai fysiikan kaavassa yleisesti käytetty symboli.
- ▶ Muuttujien nimet on syytä kirjoittaa pienellä kirjaimella ja vakioiden nimet isolla kirjaimella.



# Kommentit

- ▶ Selvittävät koodin lukijalle sen rakennetta ja toimintaa.
- ▶ Ohjelmatiedoston alkuun yleiskommentti, jossa kerrotaan ohjelman kirjoittaja ja se, mitä tiedostossa oleva ohjelma tekee.
- ▶ Jokaisen funktion otsikon yhteyteen kommentti, joka kertoo funktion tarkoituksen, parametrien merkityksen sekä funktion palauttaman arvon.
- ▶ Funktioiden sisällä on syytä kommentoida vain suurempia kokonaisuuksia ja epätavallisia ratkaisuja.
- ▶ Älä kommentoi asioita, jotka ovat itsestään selviä jokaiselle Python-kieltä osaavalle, esim.

```
i += 1    #i:ta kasvatetaan yhdellä
```

# Tyhjän tilan käyttö

- ▶ Tyhjät rivit helpottavat ohjelman rakenteen näkemistä.
- ▶ Kahden funktion väliin on aina syytä jättää 1–2 tyhjää riviä.
- ▶ Myös pidemmän funktion sisällä voi erottaa kokonaisuuksia tyhjillä riveillä.
- ▶ Älä kirjoita koodia, jossa joka toinen rivi on tyhjä — se vain hankaloittaa ohjelman lukemista.
- ▶ Erotta operaattori ja operandit toisistaan välilyönnillä.
- ▶ Myös pilkun jälkeen kannattaa yleensä kirjoittaa välilyönti.