

# Python-ohjelmointi

Kevät 2020

# Aloitus

- Kehitysympäristön lataaminen ja asentaminen omalle tietokoneelle <https://www.python.org/>
  - Downloads Python 3.8.2 (viimeisin versio)
- Käynnistetään Windowsissa esim. Käynnistä-valikon kautta IDLE (Python) -sovelluksesta
- Editori avataan File-valikosta New File
- Kun ohjelma on kirjoitettu, tallennetaan se omalle koneelle
- Käynnistetään editorissa valikosta Run | Run Module
  - Tai käynnistys suoraan F5-painikkeella

Kommenot

# print

- `print("tulostettavat tekstit")`
  - Huom! Word jne. -ohjelmat muuttavat lainausmerkit usein väärään muotoon:  
`print("tulostettavat tekstit")`
- Kirjoita ensimmäinen ohjelma
  - `print("Hello World!")`
  - Tallenna nimellä *Harjoitus1* ja suorita.

# print

- Täydennä ohjelmaa kommentilla ja toisella tulostusrivillä
  - #Ensimmäinen ohjelmani
  - print("Hello, world!")
  - print("Tämä oli ensimmäinen ohjelmani.")
- Tallenna ja suorita

# Numerot ja tekstit

- Lukutyypillä merkitys
- Kokonaisluvut: 56 tai -34
- Liukuluvut eli desimaaliluvut pisteellä: 23.4 tai -45.67
  - int = kokonaisluku (int = integer), float = desimaaliluku (liukuluku)
  
- Tekstejä sanotaan merkkijonoiksi
  - str = merkkijono (str = string)

# Muuttujat

- Muuttujiin tallennetaan tietoa
- Nimetään kuvaavalla tavalla
  - Esim. luku1 tai oppilaan\_nimi
- Muuttujan nimi ei saa alkaa numerolla
- Ei käytetä ä, ö ja å –kirjaimia
  - Ei kirjoitetta lukumäärä vaan lukumaara

# Muuttujien käyttö

- Muuttujiin sijoitetaan tietoa
  - luku1 = 27
  - luku2 = 13
  - des\_luku1 = 4.5
  
  - oppilas1 = "Aapeli"
  - oppilas2 = "Olivia Ohjelmoija"
- = on sijoitusoperaattori
  - == on vertailuoperaattori

# Muuttujien käyttö

```
luku1 = 7
```

```
luku2 = 15
```

```
luku1+luku2
```

```
tulos = luku1 - luku2
```

```
print(tulos)
```

```
oppilas = "Olli Ohjelmoija"
```

```
print(oppilas)
```

```
print(luku1+luku2)
```

# Esimerkkiohjelma

```
# Harjoitus2
# Muuttujien ja tulostuksen harjoittelua
luku1 = 5
luku2 = 4
nimi = "Oma nimi"

summa = luku1+luku2

print("Lukujen summa on", summa)
print("Nimeni on", nimi)
```

# input() eli syötteen lukeminen näppäimistöltä

- Näppäimistöltä luettu tieto tallennetaan aina muuttujaan
- Esim.
  - `oppilaan_nimi = input("Kirjoita oppilaan nimi: ")`
  - `luku1 = input("Anna jokin luku: ")`
    - `luku1`-muuttujaa ei voi käyttää laskuissa, koska se on merkkijono
- `input()`-komento lukee aina merkkijonoja (eli tekstiä)
  - Luettu tieto täytyy tarvittaessa muuttaa esim. kokonaisluvuksi

# Esimerkkiohjelma input()-komennosta

## Kirjoita ja testaa ohjelmaa Harjoitus3

```
# Harjoitus3
# Syötteen luku

oppilaan_nimi = input("Kirjoita oppilaan nimi: ")
arvosana = input("Anna kokeen arvosana: ")

print("Oppilas", oppilaan_nimi)
print("sai arvosanan", arvosana)
```

# Tyypimuunnokset input()-komennolla

## Testaa alla olevaa ohjelmaa Harjoitus4

```
#Harjoitus4
luku1 = float(input("Anna 1. luku"))    #Ohjelma muuttaa rivin desimaaliluvuksi
luku2 = int(input("Anna 2. luku"))      #Ohjelma muuttaa rivin kokonaisluvuksi

summa = luku1 + luku2

print("Lukujen summa on", summa)
```

# Ehtolause eli if-lause

- Ohjelman toiminto/komento suoritetaan vain, jos tietty ehto täyttyy

- Esim.

**if luku > 0:**

```
print("Luku on suurempi kuin nolla")
```

```
# Rivi tulostuu vain, jos luku-muuttujan arvo on suurempi  
# kuin nolla.
```

# Ehtolause eli if-lause

- Esim.

```
if luku == 0:
```

```
    print("Luku on nolla")
```

```
else:
```

```
    print("Luku on nollaa pienempi tai suurempi")
```

# Ehtolause eli if-lause

- Esim.

**if luku > 0:**

```
    print("Luku on nollaa suurempi")
```

**elif luku < 0:**

```
    print("Luku on nollaa pienempi")
```

**else:**

```
    print("Luku on nolla")
```

# Ehtolause eli if-lause

- Esim.

**if luku > 0:**

```
    print("Luku on nollaa suurempi")
```

**if luku < 0:**

```
    print("Luku on nollaa pienempi")
```

**if luku == 0:**

```
    print("Luku on nolla")
```

# Ehtolause eli if-lause

## Vertailuoperaattorit

$a == b$	a ja b ovat yhtä suuret
$a > b$	a on suurempi kuin b
$a < b$	a on pienempi kuin b
$a >= b$	a on suurempi tai yhtä suuri kuin b
$a <= b$	a on pienempi tai yhtä suuri kuin b
$a != b$	a ja b ovat erisuuria
$a <> b$	a ja b ovat erisuuria

# Ehtojen yhdistäminen

```
if luku > 2 and luku < 9:
```

```
    print("Luku on kahden ja yhdeksän välissä")
```

```
if luku > 10 or luku < -10 :
```

```
    print("Luku on yli kymmenen tai alle -10")
```

# Toistorakenne eli while-lause

```
luku = 1
```

```
while luku <= 10:
```

```
    print(luku)
```

```
    luku = luku+1
```

```
print("Silmukka on päättynyt")
```

# Toistorakenne eli while-lause

```
# Ohjelma tulostaa luvut 1, 2, 3, ..., 10
```

```
luku = 1
```

```
while luku <= 10:
```

```
    print(luku)
```

```
    luku = luku+1
```

```
print("Silmukka on päättynyt") # Sisennyksen muutos päättää rakenteen
```

# Listat

- Lista voidaan ajatella taulukkona, johon voidaan tallentaa tietoja
- Lista Oppilaat

Oona	Aada	Onni	Aapo	Kaapo	Siina
------	------	------	------	-------	-------

- Lista oppilaiden pisteet

15	19	10	21	26	8
----	----	----	----	----	---

- Kaikille oppilaiden nimille ja pisteille ei tarvita omia muuttujia

# Listat

- nimet = ["Aapo", "Bertta", "Cecilia", "Daavid", "Eetu", "Fanni"]
- koepisteet = [12, 16, 10, 15, 20, 4]
- Listan indeksi [0, 1, 2, ...]
- print("Oppilaan", nimet[0], "pisteet ovat", koepisteet[0])  
→ Tulostuu Oppilaan Aapo pisteet ovat 12

# Listan läpikäynti

```
indeksi = 0
```

```
while indeksi <= 5:
```

```
    print(koepisteet[indeksi])
```

```
    indeksi = indeksi+1
```

# Listaan tietojen syöttäminen

```
# Tallennetaan listaan luvut 1-100
```

```
lukulista = []
```

```
luku = 1
```

```
while luku <= 100:
```

```
    lukulista.append(luku)
```

```
    luku = luku+1
```

# Listaan tietojen syöttäminen

```
# Tallennetaan listaan käyttäjän kirjoittamat nimet
nimilista = []
lukumaara = int(input("Montako nimeä haluat kirjoittaa? "))
indeksi = 0
while indeksi < lukumaara:
    nimi = input("Kirjoita nimi: ")
    nimilista.append(nimi)
    indeksi = indeksi+1
```

# Listan tietojen lukeminen

```
# Tulostetaan listaan kirjoitetut nimet
```

```
indeksi = 0
```

```
while indeksi < lukumaara:
```

```
    print(nimilista[indeksi])
```

```
    indeksi = indeksi+1
```

# Toistolause - For-lause

```
for luku in range(5):  
    print("Hei")
```

# Tulostetaan Hei viisi kertaa. Muuttujan luku arvo  
# päivittyy automaattisesti

```
for luku in range(5):  
    print(luku)
```

# Tulostetaan luvut 0-4. Muuttujan luku arvo  
# päivittyy automaattisesti

```
for luku in range(1, 5):  
    print(luku)
```

# Käydään luvut 1-4 (eli  $1 \leq \text{luku} < 5$ )

# Toistolause - For-lause

```
for luku in range(10, 21):           # Käydään luvut 10-20
    print(luku)
```

```
for luku in range(10, 21, 2):       # Käydään luvut 10, 12, 14, ..., 20
    print(luku)                     # Askellus määritetään range-funktiolla
```

# Funktiot

- Ohjelma jaetaan osiin funktioilla.
- Funktio suorittaa aina jonkin oman toiminnon.
- Varsinkin ohjelmassa usein käytetyt toiminnot kannattaa kirjoittaa omiksi funktioikseen.

# Esimerkki funktioista

#Funktio

```
def laske_summa(eka, toka)
    summa = eka+toka
    return(summa)
```

#Nimet saisivat olla luku1 ja luku2

#Pääohjelma

```
luku1 = int(input("Anna ensimmäinen luku: "))
luku2 = int(input("toinen luku: "))
summa = laske_summa(luku1, luku2)
print("Lukujen summa on",summa)
```

# Funktio

- Funktio aloitetaan **def** *funktion\_nimi()*
  - Kirjoitetaan funktiot ohjelman alkuun
- Funktio voi ottaa vastaan tietoa parametreilla
  - `def funktion_nimi(parametri1, parametri2,...)`
  - Paramerejä ei ole pakko olla
- Pääohjelmassa funktion kutsussa tieto välitetään parametreilla
  - `funktion_nimi(parametri1, parametri2,...)`

# Funktion paluuarvo

- Funktio voi palauttaa lopuksi jonkin arvon

```
def laske_summa(eka, toka)
    summa = eka+toka
    return(summa)
```

- Kutsussa oltava muuttuja, johon arvo sijoitetaan  
summa = laske\_summa(luku1, luku2)

# Funktio ilman paluuarvoa

- Funktion ei tarvitse palauttaa mitään arvoa

```
def tulostus()
```

```
    print("Tulostava funktio.")
```

- Kutsu ilman parametreja

```
tulostus()
```

# Funktio ilman paluuarvoa

- Funktion ei tarvitse palauttaa mitään arvoa vaikka saisikin parametreja

```
def tulostus(nimi)
    print("Nimi on", nimi)
```

- Kutsu parametrilla  
tulostus(nimi)

# Esimerkki

- Kirjoita ohjelma, joka pyytää käyttäjältä kahta lukua
  - Jos molemmat luvut ovat alle 10, lasketaan niiden summa
  - Jos toinen on alle 10 ja toinen yli 10, lasketaan erotus
  - Jos molemmat luvut ovat yli 10, lasketaan niiden tulo
  - Jos molemmat luvut ovat arvoltaan 0, tulostetaan teksti "Nollatullos."
- Käytä if-lauseita ja funktioita