Ohjelmointi lukion matematiikassa, mitä opettaa opintojaksossa MAA11?

Perusteiden jatko

osa 1.

Kertaus

OPETUSHALLITUS RAHOITTAA HANKETTA

Hello world!

Teppo Harju, fysiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio

Ville Tilvis, matematiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio

Mika Setälä, matematiikan opettaja, Lempäälän lukio

Antti Laaksonen, yliopistonlehtori, Helsingin yliopisto, tietojenkäsittelytieteen osasto

Päivän aikataulu ma 7.2.2022

12.30 - Hei kaikki, esittelyt, tämä dia

12.30 - Osa 1. Kertaus

- Eri ympäristöt
- Perusteiden kertaus

13.00

- Harjoitustehtäviä

14.00 - Osa 2. Uutta asiaa

- funktiot ja listat
- Esimerkkejä ja harjoituksia

14.30 Harjoituksia pienryhmissä

15.30 - Lopetus

Koulutuksen materiaalikansio

bit.ly/ohjelmointimaol

Hei maailma!

Kerro lyhyesti:

Kuka olet ja mistä tulet?

Koulutuksen tarkoitus ja päivän tavoite

- Harjaantua ohjelmoimaan yksinkertaisia ohjelmia Pythonilla
- Ideoita ja esimerkkejä moduulin maa11 opettamiseen

Päivän päätteeksi:

- Olen kerrannut Pythonin perusteet
- Olen oppinut lisää ohjelmointia

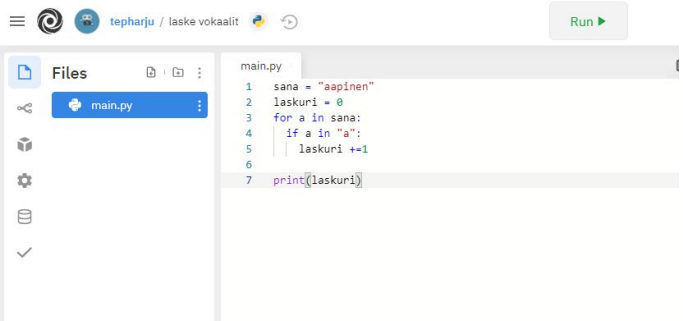
Valitse jokin ympäristö Pythonin kirjoittamiseen

Helpoin vaihtoehto:

- https://cheat.abitti.fi/

Muita vaihtoehtoja

- https://replit.com
- Python asennettuna omalle koneelle
- Ti-nspire CX CAS



Abitin oma Python-tulkki

FYSIIKKA KEMIA MAANTIEDE MATEMATIIKKA MUSIIKKI NÄPPÄIMISTÖ **OHJELMOINTI** YLEISOHJEET PÅ SVENSKA **SUOMEKSI**

Hae/Sök

Ohjelmointi

- Voit selata ohjelmoinnin lähdemateriaalia (Python 3).
- Voit kopioida lähdemateriaalissa olevia ohjelmakoodeja leikepöydälle klikkaamalla haluamaasi esimerkkikoodia.
- Liitä leikepöydälle kopioimasi koodi klikkaamalla ohjelmakoodi-ikkunaa ja painamalla ctrl-v.
- Suorita ohjelmakoodi-ikkunassa oleva koodi klikkaamalla Suorita ohjelma -painiketta.
- Ohjelmakoodi-ikkunaa ei varmuuskopioida. Voit kopioida ohjelmakoodisi koesuoritukseen, jossa se samalla varmuuskopioidaan.

Tekstin tulostaminen

```
print("Hello world!")
```

```
Hello world!
```

Kommentointi

Kommenttirivi alkaa `#`-merkillä. Kun koodi suoritetaan, hypätään kaikkien kommenttien yli.

```
#This is a comment

#This is a
#multiline comment

"""
This is also

a multiline comment
"""
```

Kommentointi koodirivillä

```
name = "Onni"   # Set the value "Onni" to variable name
```

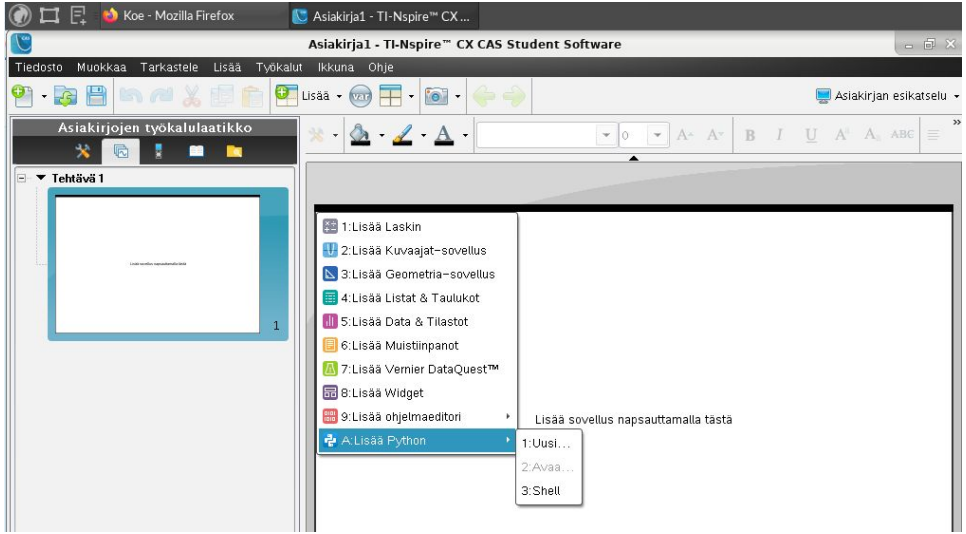
Python 3 -ohjelmakoodi:

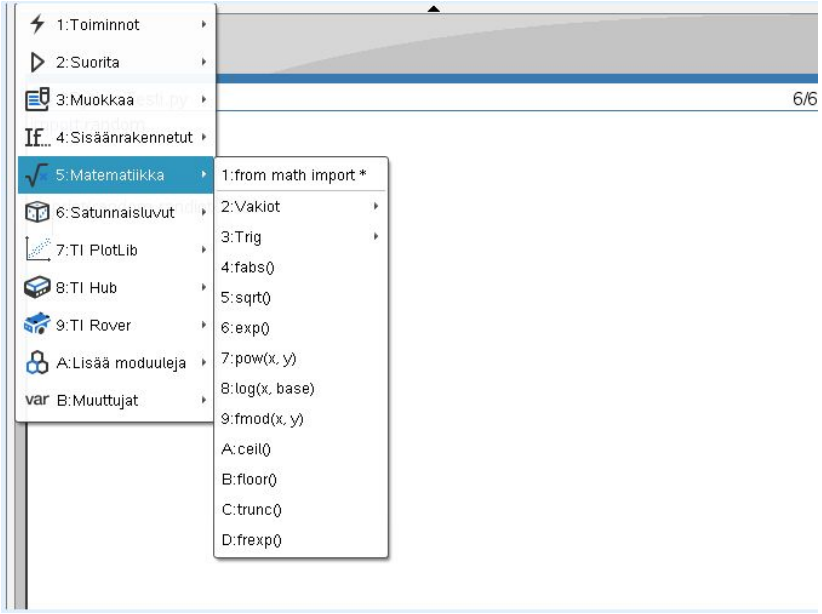
```
1
```

Suorita ohjelma | Kopioi leikepöydälle

Tulosteet / virheet:

TI-Nspire CAS CX CAS





Tiedosto Muokkaa Tarkastele Lisää Työkalut Ikkuna Ohje

Sisältö | Asiakirjat

Lisää

Asiakirjan esikatselu

Asiakirjojen työkalulaatikko

Matematiikkamallit

Symbolit

Lisää kohde kaksoisnapsauttamalla kuvaketta

Ohjatut toiminnot päällä

Kreikkalainen iso delta

Katalogi

Matemaattiset operaattorit

Yksikkömuunnokset

Testikoodi.py 10/10

```
#Tähän kirjotetaan koodia

x = 55

a = 84

print(a*x)

print(a+x)
```

1:Äskeinen
2:Muuttujat: Testikoodi.py
3:Leikkaa
4:Kopioi
5:Liitä
6:Suorita
7:Tarkista syntaksi ja tallenna (Ctrl+B)
8:Siirry Shelliin

Python Shell 5/5

```
>>>#Running Testikoodi.py
>>>from Testikoodi import *
4620
139
>>>
```

Mitä (ehkä) osaan jo nyt?

- Osaan käyttää Pythonia laskuihin
- Osaan tulostaa tervehdyksiä
- Osaan pyytää syötettä käyttäjältä
- Osaan kirjoittaa ehtolauseen
- Osaan toistaa asioita

- Jos aloitat nollasta → https://tie.koodariksi.fi/maa11 Perusteet

Ensimmäinen ohjelma

```
print("Hei maailma!")

print("Tervetuloa mukaan!")

# Tämä on kommentti

print("lasketaan laskuja")

print(2+2)

print("2+2")

print(5*(3+4))
```

Komennot suoritetaan järjestyksessä rivi kerrallaan ylhäältä alas

Jos rivin alussa on #-merkki, ei rivi vaikuta ohjelman toimintaan

Lainausmerkeillä merkitään *merkkijono*, joka tulostuu ruudulle sellaisenaan

Muuttujat

- Muuttujiin voi tallentaa tietoa, jota tarvitaan ohjelmassa myöhemmin
- Asetetaan muuttujaan nimeltä *`nimi`* arvo *`Topias`*
- Muuttujaan **asetetaan arvo** yhtäsuuruusmerkillä (=)

Muuttujan nimi → `nimi = "Topias"` ← Muuttujan arvo

```
a = 3

summa = 5

tulos = 5 * summa
```

Muuttujan tyyppi

- **Tyyppi** (engl. type) tarkoittaa millainen jokin arvo on

```
nimi = "Tuomas"
```
str, merkkijono

```
tulos = 500
```
int, kokonaisluku

```
keskiarvo = 7.78
```
float, desimaaliluku

- muuttuja *nimi* on tyyppiä *merkkijono,* ***str***
- *tulos* on tyyppiä *kokonaisluku,* ***int***
- *keskiarvo on tyyppiä liukuluku (desimaaliluku),* ***float***
- muita tietotyyppejä: ***boolean, list*** jne.

funktio type() kertoo tyypin

print(type(nimi))
<class ´str´>

Syötteen lukeminen käyttäjältä, *input-komento*

```
nimi = input("Anna nimesi:")
```

input lukee käyttäjältä *merkkijonon* ja asettaa sen **muuttujan** nimi arvoksi

```
print("Terve:", nimi)

print("Terve:" + nimi)

print(f"Terve vaan: {nimi}")
```

Erilaisia tapoja tulostaa tekstiä ja muuttujan nimi arvo

Syötteen lukeminen käyttäjältä, *input-komento*

```
nimi = input("Anna nimesi:")
```

input lukee käyttäjältä *merkkijonon* ja asettaa sen muuttujan nimi arvoksi

```
syote = input("Minä vuonna olet syntynyt? ")

vuosi = int(syote)
```

merkkijono on muutettava kokonaisluvuksi, jolloin laskeminen onnistuu

```
print("Ikäsi vuoden lopussa on:", (2021 - vuosi))

print(f"Ikäsi vuoden lopussa on: {2021-vuosi}")
```

math-moduuli

- *math*-moduuli sisältää hyödyllisiä matemaattisia funktioita ja vakioita
- Ensin math-moduuli täytyy tuoda mukaan **import**-käskyllä

```
import math
```

Tuodaan koko math-moduuli käyttöön

```
print(math.exp(4))      # exp(x) -> Eksponenttifunktio e^x
print(math.sqrt(9))     # sqrt(x) -> Luvun x neliöjuuri (kuten x**(1/2))
print(math.pi)          # pi -> pii (ei ole funktio vaan vakio)
print(math.log(10,2))     #  log_2(10)

round(a,n)             # pyöristää luvun a n:n desimaalin tarkkuuteeen
```

Valmiit kirjastomoduulit -kertaus

- Sisältävät rutiineja ja toimintoja erilaisiin tehtäviin (math, random, turtle)
- Usein testattuja ja dokumentoituja
- “Olisiko joku keksinyt jonkin tavan tehdä tämän helpommin?”
- Koko moduuli otetaan käyttöön `import` -lauseella
- Moduulin osia voi valita käyttöön `from` -komennolla

`from math import sqrt, log` tai `from math import *` (ottaa kaiken käyttöön math -moduulista)

```
print(sqrt(25))

print(log(5,2))
```

random

- random -moduulilla voit esim. generoida satunnaislukuja
- dokumentaatio: https://docs.python.org/3/library/random.html

```
import random

luku = random.randint(a, b)
```

tai

```
from random import *

toka = randint(1,11)
```

Asettaa muuttujan luku arvoksi satunnaisluvun a <= N <=b

Esimerkki 1.

Kirjoitetaan lyhyt tietokilpailu

- Ohjelma kysyy käyttäjän nimeä ja syntymävuotta
- Tervehtii käyttäjää ja kertoo käyttäjän iän
- Ohjelma kysyy kaksi kysymystä ja antaa pisteitä
- Ohjelma laskee pisteiden neliöjuuren

Ehtorakenne

- **Ehtorakenteiden** avulla ohjelman toiminta voi vaihdella sen mukaan, mitä tietoja sille kulloinkin annetaan.
- **Lohko** (*engl. block*) on joukko peräkkäisiä lauseita, jotka ovat samalla tasolla rakenteessa

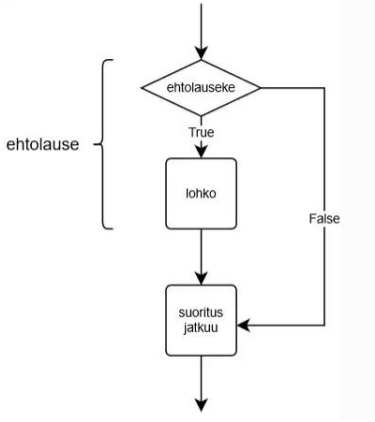
```
if ika >= 18:
    # lohko alkaa
    print("Voit luovuttaa verta!")
    print("Voit äänestää!")
    # lohko loppuu
print("Tämä on eri lohkossa")
```

Ehtorakenne, joko tosi (True) tai epätosi (False)

sisennys (tab)

Lohko

"päälohkon" on oltava sisennetty tiedoston vasempaan reunaan



Lisää ehtolauseita *else-if*, eli *elif*

```
luku = int(input("Anna kokonaisluku: "))
if luku < 0:
    print("Luku on negatiivinen.")
elif luku == 0:
    print("Luku on nolla.")
elif luku > 0:
    print("Luku on positiivinen.")
else:
    print("Liekö tätä olemassakaan?")
```

jokaisella if-elif -rivillä totuustestaus

elif -lohkoista toteutetaan vain se, jossa ehto on tosi

Toistolause eli silmukka, *while*

- Silmukalla voidaan toistaa samaa koodia useamman kerran
- **while** -silmukka (“*luuppi*”)

```
while True:
    luku = int(input("Anna luku, 0 lopettaa "))
    if luku == 0:
        break
    print(luku ** 2)

print("Ohjelma loppui")
```

ehto toistolle (kun TOSI)

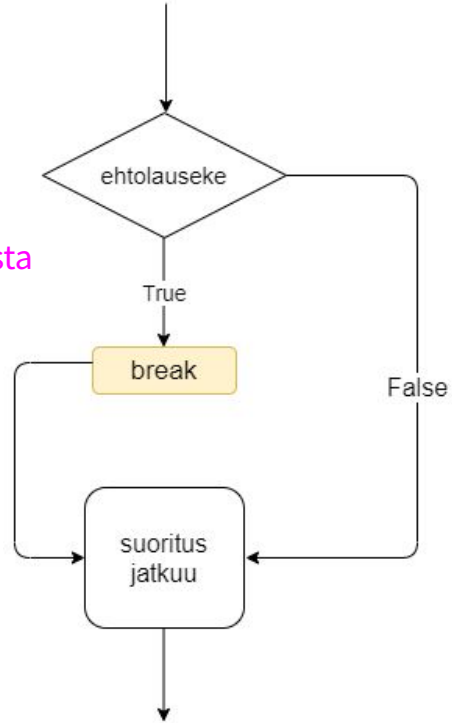
sisennys (tab)

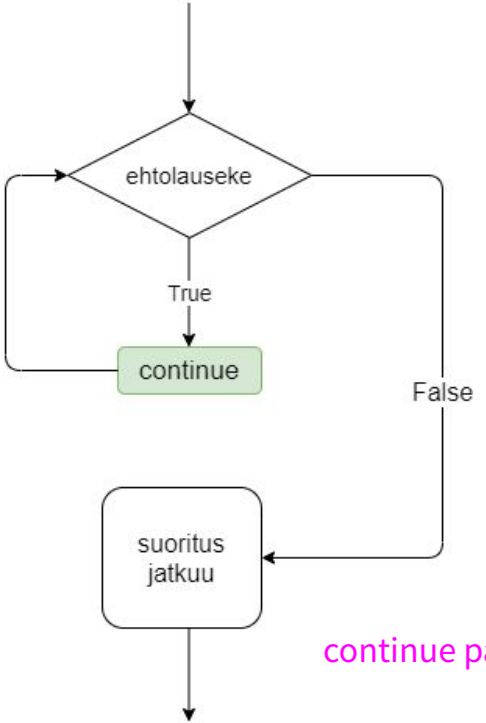
kun käyttäjä syöttää luvun 0, suoritetaan ehtolohkossa break-komento, ja ohjelma poistuu while-lohkosta

ohjelman suoritus jatkuu pääohjelman tasolta

break ja continue

break poistuu toistolauseesta





continue palaa toistolauseen alkuun

while ja ehto silmukassa

```
laskuri = 0

luku = int(input("Anna luku, 0 lopettaa "))

while laskuri < 10:

    print(luku ** 2)

    laskuri = laskuri + 1


print("Ohjelma loppui")
```

Toisto: for-silmukka

- Jos haluamme toistaa jonkin asian tietyn määrän kertoja tämä onnistuu **for**-silmukalla ja funktion **range** avulla.

laskurimuuttuja

laskurimuuttuja
saa arvot 1, 2, 3, 4

```
for i in range(1,5):

    print("Rakastan ohjelmointia!")
```

- **range(a,b)** aloittaa luvusta a ja *lopettaa juuri ennen lukua b*
- **range(n)** käy läpi luvut 0, 1, 2, . . . , n-1
- **range(a,b,c)**, nyt luku kasvaa c:llä joka askeleen jälkeen

Esimerkki 2.

Kirjoitetaan arvauspeli

- Ohjelma arpoo jonkin kokonaisluvun väliltä 1-100
- Käyttäjällä on viisi yritystä arvata luku

  Versio 1. for-toisto

  Versio 2. while-ehto

while vs. for

- Tulostetaan luvut 1-10
- Kumpi tapa parempi?

Yleensä:

- **for** -> valmis data (lista, sanakirja, sana jne.) jota haluamme käsitellä
- **while** -> ei valmista dataa, “suorita toiminto kunnes…”

for-silmukka ja Python visualizer

https://pythontutor.com/visualize.html#mode=display

Harjoitellaan!

Tehtävämoniste Osa 1. -Kertaavia tehtäviä

bit.ly/ohjelmointimaol

https://www.pythoncheatsheet.org/