

Ohjelmointi lukion matematiikassa, mitä opettaa opintojaksossa MAA11?

Perusteet
osa 2. Ehtolauseet ja toisto



Yleisiä ongelmia

SyntaxError: invalid syntax

Tarkista oikeinkirjoitus. Lukeeko koodissasi print vai pirrt, vai Print

IndentationError: expected an indented block

Pythonissa esim. Ehtolauseet vaativat sisennyksen (Tab)

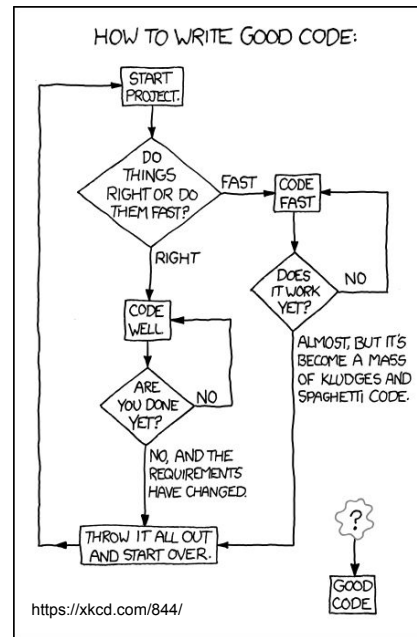
NameError: global name '---' is not defined

Et ole määritellyt muuttujaa tai sen arvoa ennen muuttujan käyttöä

Kirjoitusvirhe?

TypeError: Can't convert 'int' object to str implicitly

Tietotyyppejä ei voi yhdistää. Answer = "Hello" +1



Muuttujan tyyppi

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

```
nimi = "Tuomas"
```

```
tulos = 500
```

```
keskiarvo = 7.78
```

funktio `type()` kertoo tyyppin

```
print(type(nimi))  
<class 'str'>
```

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



Ehtorakenne

- Usein on tarpeen määritellä ohjelmaan osia, jotka suoritetaan vain tietyissä tilanteissa

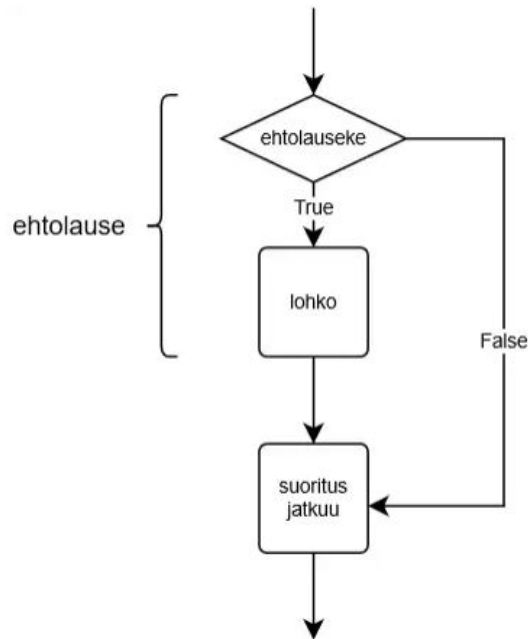
```
ika = int(input("Kuinka vanha olet? "))
```

```
if ika > 17:                      ehdon lopussa  
                                  kaksoispiste
```

```
           print("Samppanjaa, ole hyvä!")
```

sisennys
(tab)

```
print("Ohjelma jatkuu...")
```



Lause ja lohko

- **Lause** (*engl. statement*) on ohjelman osa joka suorittaa toiminnon

```
print("Rakastan ohjelmointia") tai alkuarvo = 0
```

- **Lohko** (*engl. block*) on joukko peräkkäisiä lauseita, jotka ovat samalla tasolla rakenteessa

```
if ika >= 18:
```

```
    # lohko alkaa
```

```
    print("Voit luovuttaa verta!")
```

```
    print("Voit äänestää!")
```

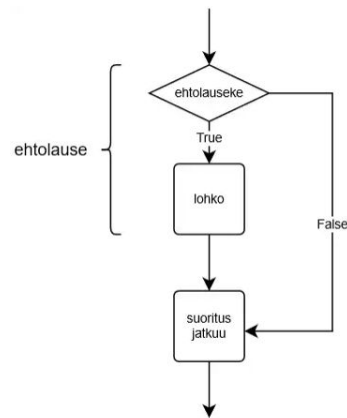
```
    # lohko loppuu
```

```
print("Tämä on eri lohkoissa")
```

sisennys
(tab)

Lohko

"päälohkon" on oltava sisennetty
tiedoston vasempaan reunaan



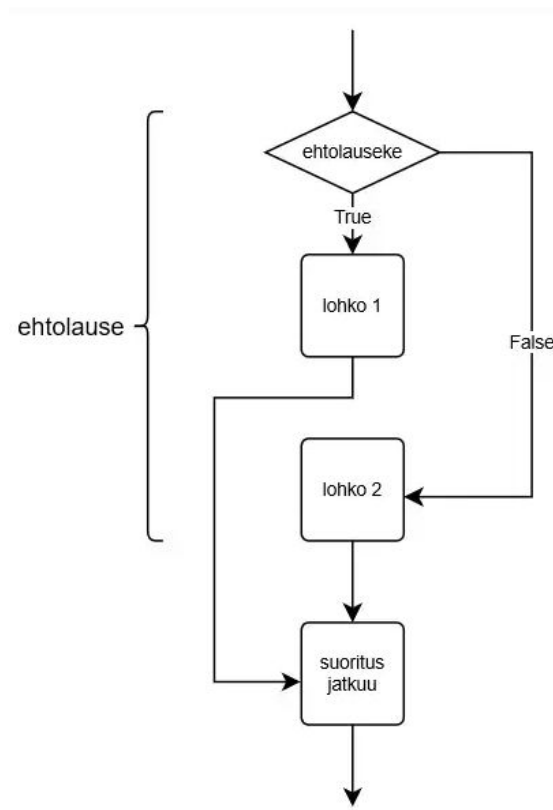
Vertailuoperaattorit

Operaattori	Merkitys	Esimerkki
==	Yhtä suuri	$a == b$
!=	Eri suuri	$a != b$
>	Suurempi	$a > b$
>=	Suurempi tai yhtä suuri	$a >= b$
<	Pienempi	$a < b$
<=	Pienempi tai yhtä suuri	$a <= b$



Lisää ehtolauseita, else

```
luku = int(input("Anna luku: "))  
  
if luku < 0:  
    print("Luku on negatiivinen.")  
  
else:  
    print("Luku on positiivinen tai nolla.")
```



Loogiset operaattorit: *and*, *or* ja *not*

- Operaattori **and** vaatii, että useampi ehto pätee samaan aikaan

```
if luku >= 4 and luku <= 8:  
    print("Luku on välillä [4,8]")
```

- Operaattori **or** vaatii, että yksi tai useampi ehdoista pätee

```
if luku < 5 or luku > 8:  
    print("Luku ei ole välillä [5,8]")
```



Esimerkkiohjelma

```
Anna luku: 6  
Luku on parillinen!  
Anna alaraja: 1  
Anna yläraja: 8  
Luku 6 on välillä 1...8  
➤ 
```



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: for-silmukka



Harjoituksia pienryhmissä <https://tie.koodariksi.fi/maa11>

Tie koodariksi

MAA11 Algoritmit ja lukuteoria



Tehtäväsivut

- [Johdanto Pythoniin](#)
- [Yhtälön ratkaiseminen](#)
- [Monte Carlo -simulaatioita](#)
- [Liukuluvuista](#)
- [Järjestämisalgoritmit](#)
- [RSA-salaus](#)

Tee tämä
valmiiksi

Jatka näihin,
jos ehdit

Software can be chaotic, but we make it work



Expert

Trying Stuff
Until it Works

○ RLY?

The Practical Developer
@ThePracticalDev

@elbruno



while: toisto niin kauan, kun ehto on voimassa

```
1  # Collatzin konjektuuri
2
3  luku = int(input("Anna kokonaisluku: "))
4
5  print(luku)
6
7  while (luku > 1):
8
9      if luku % 2 == 0:
10
11          luku = luku//2
12
13      else:
14
15          luku = 3*luku+1
16
17      print(luku)
```

```
Anna kokonaisluku: 7
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
```



Liukuluvut (float)

```
print(f"{(1/3):.18f}")
```

```
0.333333333333333315
```

```
1 import math
2 print(math.sin(math.pi))
```

```
1.2246467991473532e-16
```

```
1 print(2**100 + 1 - 2**100)
2
3 print(2.0**100 + 1 - 2.0**100)
```

```
1
0.0
```

$99^{100^{101}} > 101^{100^{99}}$ ▶ false

$99^{100^{101}} < 101^{100^{99}}$ ▶ false

$99^{100^{101}} = 101^{100^{99}}$ ▶ true



Harjoituksia osa 2, perustaso

1. Tie koodariksi -harjoitukset 7 - 18.
2. Tee ohjelma, joka kysyy käyttäjältä lukua, kunnes luku on positiivinen.
3. Tee ohjelma, joka testaa onko käyttäjän syöttämä salasana oikein. Käyttäjällä on vain kolme yritystä, jonka jälkeen ohjelma loppuu.
4. Tee ohjelma joka laskee valitsemaasi laskutoimitusta ikuisesti. Korjaa sitten ohjelmasi toistamaan laskutoimitusta vain haluttu määrä tai kunnes jokin ehto täyttyy.
5. Tee ohjelma, joka kysyy kahden henkilön nimiä ja heidän syntymävuottaan. Ohjelma tulostaa kumpi on vanhempi.
6. Tee ohjelma joka kysyy positiivisen kokonaisluvun N. Tämän jälkeen ohjelma tulostaa luvut väliltä -N...N, nollaa lukuunottamatta!
7. Kansiossa <https://bit.ly/ohjelmointimaol> on tiedosto **yo-tehtavat.pdf** , jossa on ohjelmoimalla ratkaistavissa olevia vanhoja yo-tehtäviä



Mistä löydän apua? Stackoverflow?

- Pythonin dokumentaatio (käyttöohjeet):
<https://docs.python.org/3/tutorial/index.html>
- googlaaminen (“*python logarithm*”)
- Stackoverflow, ohjelmointiin keskittynyt Q&A -sivusto: <https://stackoverflow.com/>



Calculate logarithm in python

If you use `log` without base it uses `e`.

Therefor you have to use:

```
import math
print( math.log(1.5, 10))
```



Mitä seuraavaksi?

Tie koodariksi: <https://tie.koodariksi.fi/alkeet/>

Ohjelmoinnin MOOC 2021: <https://ohjelmointi-21.mooc.fi/>

Ohjelmoinnin opas opettajalle: <https://shop.edita.fi/tuote/python-ohjelmoinnin-opas-opettajalle>

Coursera, Python for everybody: <https://www.coursera.org/specializations/python>

Project Euler: <https://projecteuler.net/>

Lukema-verkosto: <https://www.lukemaverkosto.fi/materiaali/algoritmit-ja-ohjelmointi-lukion-matematiikassa/>

Codecademy: <https://www.codecademy.com/>

<http://pythontutor.com/>

<https://www.pythoncheatsheet.org/>



Kysyttävää?



Ei käytetty->



Toistolause eli silmukka, *while*

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

```
while True:
```

ehto toistolle

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

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

```
        break
```

kun käyttäjä syöttää luvun 0, suoritetaan
break-komenta, ja suoritus hyppää
while-lohkon jälkeiselle riville

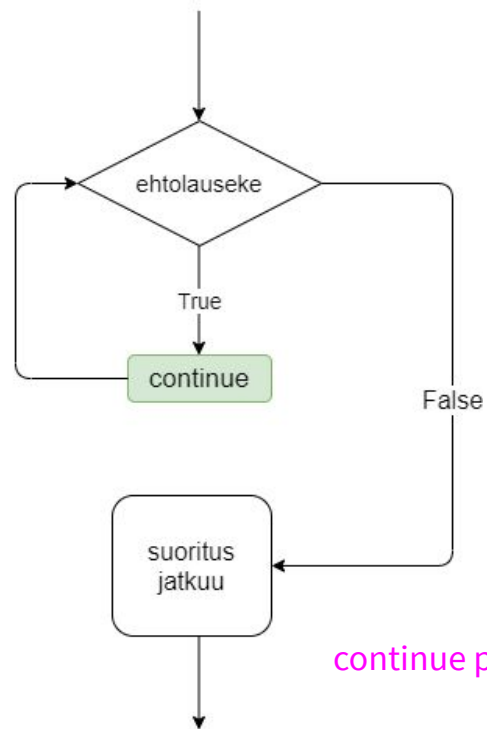
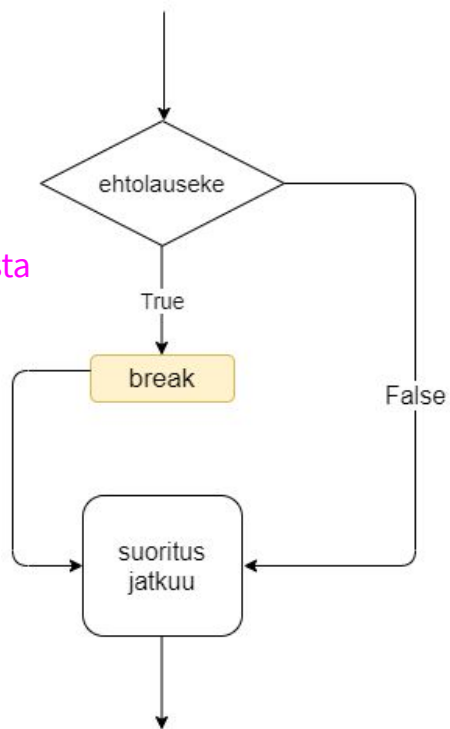
```
    print(luku ** 2)
```

```
print("Ohjelma loppui")
```



break ja continue

break poistuu toistolauseesta



continue palaa toistolauseen alkuun



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 se, jossa ehto on tosi



Loogiset operaattorit: *and*, *or* ja *not*

- **not** muuttaa totuusarvon käänteiseksi

```
if not luku % 2 == 0:  
    print("Luku on pariton!")
```



Lauseke

- Lauseke (*engl. expression*) on koodin osa, jolla on jokin tyyppi

Lauseke	Arvo	Tyyppi	python
$2 + 4 + 6$	12	kokonaisluku	int
"abc" + "xd"	"abcxd"	merkkijono	str
$11 / 2$	5.5	liukuliku(desimaaliluku)	float
$2 * 5 > 9$	True	totuusarvo	bool



Miten käy nyt?

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

```
while True:
```

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

```
        break
```

```
    print(luku ** 2)
```

```
print("Ohjelma loppui")
```



while ja laskuri ehdossa

```
laskuri = 0
```

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

```
while laskuri < 10:
```

```
    print(luku ** 2)
```

```
    laskuri = laskuri + 1
```

```
print("Ohjelma loppui")
```

