

EHTOLAUSEET

Ohjelmoinnin peruskurssi yläkouluun

© 2019 Oppimisanalytiikan keskus / Erkki Kaila, Heidi Kaarto ja Heidi Laine, kuvat Johan Sjövall



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

OHJELMAN SUORITUS

- Tähän mennessä ohjelmien suoritus on ollut suoraviivaista.

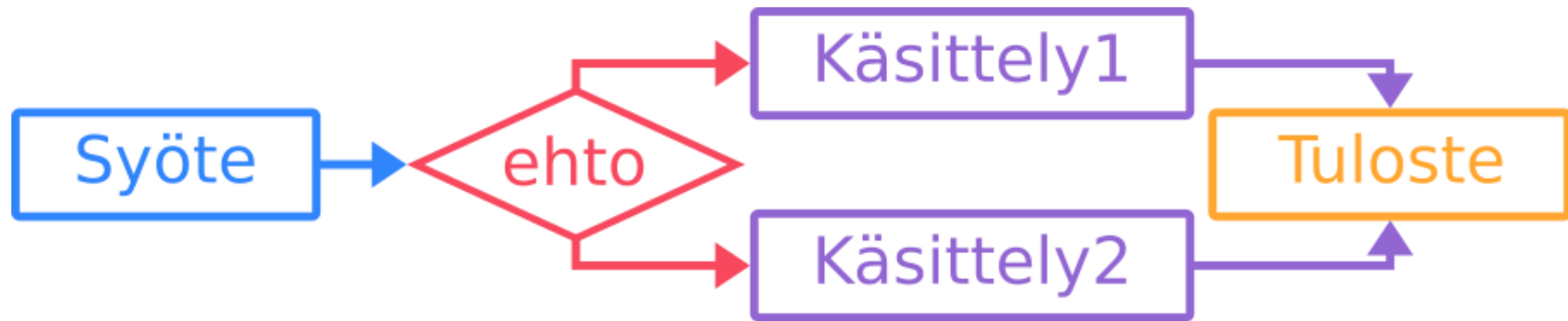


**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

EHDOLLINEN OHJELMAN SUORITUS

- On kuitenkin tyypillistä, että ohjelman suoritus **haarautuu** jonkin ehdon perusteella.



ESIMERKKI

- Ehdollisen suorituksen avulla on mahdollista esimerkiksi:
 - laskea tulos eri tavalla negatiiviselle tai positiiviselle arvolle,
 - tulostaa eri asioita perustuen käyttäjän syötteeseen tai
 - jättää käsittelemättä sellaiset syötteet tai arvot, joilla tulosta ei voida laskea.



ESIMERKKI

- Ohjelma voi pyytää käyttäjää syöttämään syntymävuoden ja laskea sen perusteella iän.
- Jos käyttäjä syöttää virheellisen vuosiluvun (esimerkiksi negatiivisen luvun tai suuremman luvun kuin nykyinen vuosiluku), ohjelma voi tulostaakin iän sijaan virheilmoituksen.



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

PYTHONISSA

- Ehdollinen suoritus on mahdollista Pythonissa **ehtolauseiden** avulla.
- Ehtolauseessa testataan jotakin ehtoa ja sen perusteella **suoritetaan tai ei suoriteta** joitain lauseita.
- Ehtolauseissa **ehto on aina totta tai epätotta**. Tähän käytetään Pythonissa totuusarvotyyppejä.



TOTUUSARVO

- **Totuusarvo** on yksi Pythonin tyypeistä. Niitä on **kaksi**: True (tosi) ja False (epätosi).
- Totuusarvoja voi käyttää lausekkeina ja muuttujien arvoina.
- Totuusarvot **eivät ole merkkijonoja**, eivätkä ne tarvitse lainausmerkkejä. Muista kuitenkin kirjoittaa totuusarvot **isolla alkukirjaimella**.



ESIMERKKI

```
totuus = True
```

```
onkoTotta = False
```

```
print(onkoTotta) # Tulostaa False
```

```
print(True) # Tulostaa True
```

```
eiTotuus = "True" # Tämä on merkkijono!!!
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

VERTAILUOPERAATTORIT

Operaattori	Merkitys	Esimerkki
==	Yhtä suuri kuin	4 == 5 # <i>False</i>
!=	Ei yhtä suuri kuin	4 != 5 # <i>True</i>
<	Pienempi kuin	4 < 5 # <i>True</i>
>	Suurempi kuin	5 > 5 # <i>False</i>
<=	Pienempi tai yhtä suuri kuin	5 <= # <i>True</i>
>=	Suurempi tai yhtä suuri kuin	5 >= 4 # <i>True</i>



ESIMERKKI

```
pienempi = 3 < 6 # True
```

```
suurempi = 3 > 6 # False
```

```
print( (3 * 4) >= (6 + 6) ) # tulostaa True
```

```
luku = 5
```

```
print(luku == 4) # tulostaa False
```

```
print(luku != 4) # tulostaa True
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

MERKKIJONOJEN VERTAILU

- Vertailuoperaattoreilla voi myös **vertailla merkkijonoja** toisiinsa.
- Merkkijonoja vertaillaan **merkkien arvon mukaan**. Merkit on tietokoneen muistissa numeroina, joita käytetään merkkijonojen vertailussa.
- Pienillä ja isoilla kirjaimilla on **eri arvot**, joten "a" \neq "A".



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
onko = ("Python" == "python") # False
```

```
sana = "ohjelmointi"
```

```
samat = ("ohje" + "lmointi" == sana) # True
```



**TURUN
YLIOPISTO**

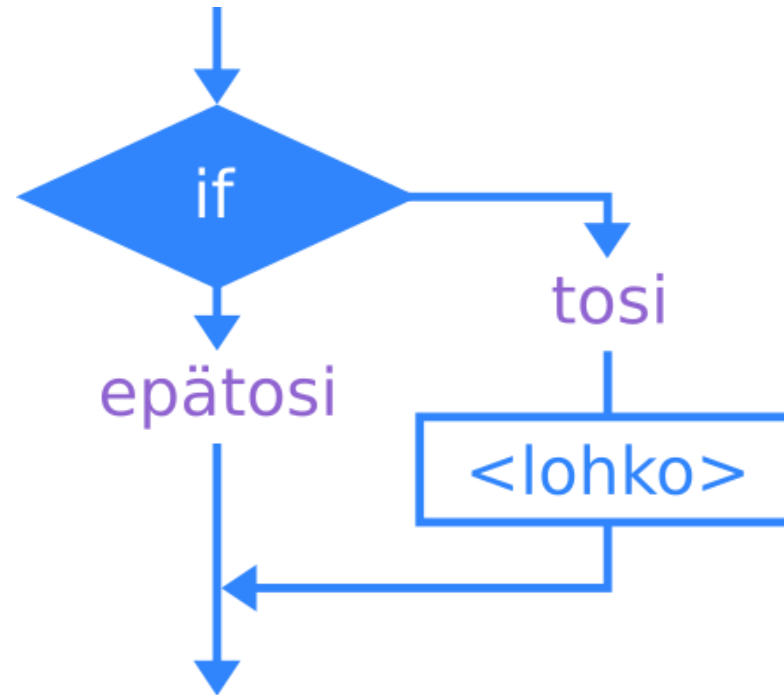
Oppimisanalytiikan keskus

EHTOLAUSE if

- Ehtolauseke ei yksinään riitä haarauttamaan ohjelman suoritusta vaan siihen tarvitaan **ehtolause**.
- Ehtolauseen tunnistaa `if`-sanasta.
- Jos ehtolauseen **ehto on totta, ehtolauseen jälkeinen lohko suoritetaan**. Jos ehto ei ole totta, ehtolauseen jälkeistä lohkoa ei suoriteta.



EHTOLAUSEEN SUORITUS



EHTOLAUSEEN SYNTAKSI

```
if <ehtolauseke>:  
    <lohko>
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
totuus = True
```

```
if totuus:
```

```
    print("Totta on.") # Tämä suoritetaan.
```

```
ei_tosi = False
```

```
if ei_tosi:
```

```
    print("Tämä ei ole.") # Tätä ei suoriteta.
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
if True:
```

```
    print("Lohkossa ollaan.")
```

```
print("Nyt ei enää olla lohkossa.")
```

```
if False:
```

```
    print("Tämä on eri lohko, mutta tätä ei suoriteta.")
```

```
print("Tämä suoritetaan kuitenkin!")
```



**TURUN
YLIOPISTO**

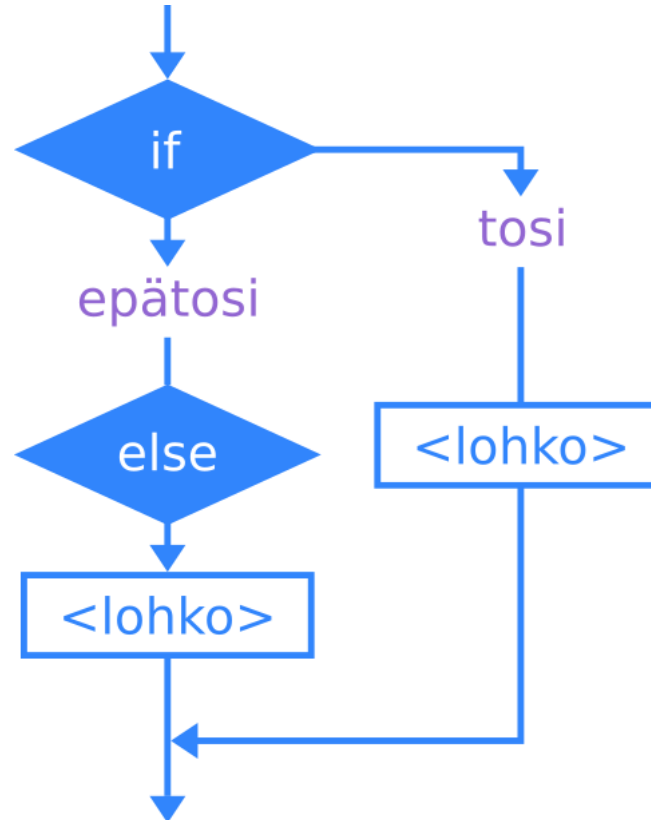
Oppimisanalytiikan keskus

VAIHTOEHTOINEN LAUSE `else`

- Ehtolauseen `if` jälkeen voidaan määritellä **vaihtoehtoinen lohko**, joka suoritetaan silloin, kun ehtolauseen `if` ehto on **epätosi** eli saa arvon `False`.
- Pythonissa tämä toteutetaan `else`-lauseella.
- `if-else`-rakenteessa suoritetaan **aina jompikumpi lohko**.



VAIHTOEHTOISEN LAUSEEN SUORITUS



if-else-RAKENTEEN SYNTAKSI

if <ehtolauseke>:

 <lohko>

else:

 <vaihtoehtoinen lohko>



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
if True:
```

```
    print("Moi.") # Tämä suoritetaan.
```

```
else:
```

```
    print("Hei.") # Tätä ei suoriteta.
```

```
if False:
```

```
    print("Moikka.") # Tätä ei suoriteta.
```

```
else:
```

```
    print("Heippa.") # Tämä suoritetaan.
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

MIKSI `if-else` EIKÄ `if-if`?

- Kaksi peräkkäistä `if`-lausetta voi johtaa siihen, että molempien lohkot suoritetaan. `if-else`-rakenteessa näin ei käy koskaan.

```
x = 7
```

```
if x < 10: # totta
```

```
    x = x + 5 # x = 12
```

```
if x > 10: # nyt tämä on totta myös
```

```
    x = x - 5
```



**TURUN
YLIOPISTO**

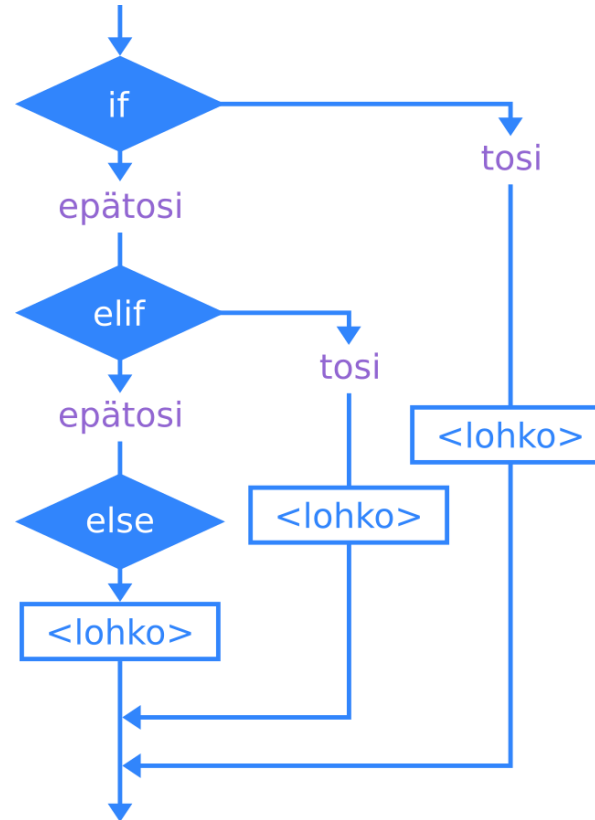
Oppimisanalytiikan keskus

EHTOLAUSE `elif`

- Useita ehtolauseita on mahdollista **ketjuttaa**. Pythonissa tähän käytetään `elif`-lausetta.
- `elif`-lauseiden ehdot tarkastetaan vain, jos alkuperäisen `if`-lauseen ehto on epätosi.
- `elif`-lauseita voi olla `if`-lauseen jälkeen **niin monta kuin niitä tarvitaan**. Lopuksi voidaan kirjoittaa `else`-lause, mutta se ei ole pakollista.



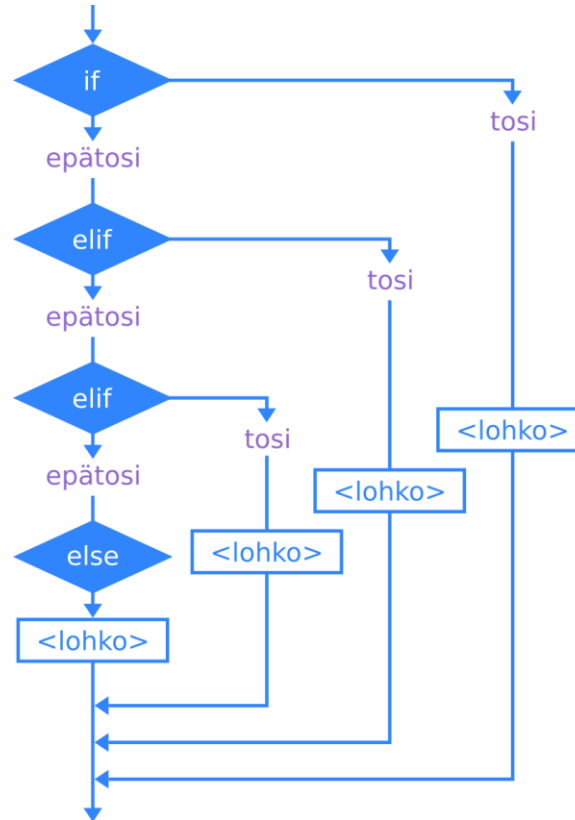
KETJUTETTUIJEN EHTOLAUSEIDEN SUORITUS



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

KETJUTETTUIJEN EHTOLAUSEIDEN SUORITUS



if-elif-else-RAKENTEEN SYNTAKSI

if <ehtolauseke>:

<lohko>

elif <ehtolauseke>:

<lohko>

else:

<lohko>

if <ehtolauseke>:

<lohko>

elif <ehtolauseke>:

<lohko>

elif <ehtolauseke>:

<lohko>

else:

<lohko>



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

LOGISET OPERAATTORIT

Operaattori	Merkitys	Esimerkki
and	Tosi, jos molemmat tosia	a = True b = False x = a and b # False
or	Tosi, jos molemmat tai jompikumpi tosia	a = True b = False x = a or b # True
not	Tosi, jos epätosi eli kääntää totuusarvon päinvastaiseksi	a = True b = not a # False x = not b # True



LOGISTEN OPERAATTOREIDEN TOTUUSTAULUT

and

a	b	a and b
True	True	True
True	False	False
False	True	False
False	False	False

or

a	b	a or b
True	True	True
True	False	True
False	True	True
False	False	False

not

a	not a
True	False
False	True



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus

ESIMERKKI

```
luku = input("Anna luku:")  
luku = int(luku)  
if (luku > 0) and (luku < 10):  
    print("Luku on välillä 1-9")  
elif (luku <= 0) or (luku >= 10):  
    print("Luku ei ole välillä 1-9")
```



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus



**TURUN
YLIOPISTO**

Oppimisanalytiikan keskus