

# Python ohjelmoinnin perusteet (5 op)

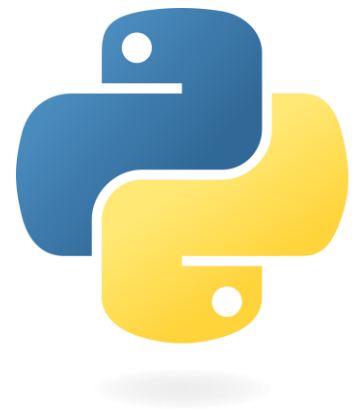
---

Heidi Lehikoinen  
Lehtori  
ICT ja tuotantotalous

# Moduuli 03: Valintarakenne (if)

---

- ❑ Termejä
- ❑ Sisennyksestä
- ❑ Valintarakenne
  - if-lause
  - Vertailuoperaattorit
  - Loogiset operaattorit
- ❑ *else*-haara
- ❑ *elif*-haara



# Termejä

---

- ❑ **Lauseke (statement)** = koodin osa joka suorittaa jotain, yksittäinen komento

```
print("Hello")
```

- ❑ **Lohko (block)** = joukko perättäisiä lausekkeita, jotka ovat samalla ohjelmarakenteen tasolla

```
if age > 18:  
    print("You are an adult!")  
    print("You can drive a car!")  
print("This here belongs to different block")
```

- ❑ **Lauseke (expression)** = pala koodia, joka tuottaa tietyn tietotyypin. Lausekkeella on aina tyyppi, joten sen tulos voidaan tallentaa muuttujaan.

Expression	Value	Type	Python data type
<code>2 + 4 + 3</code>	<code>9</code>	integer	<code>int</code>
<code>"abc" + "de"</code>	<code>"abcde"</code>	string	<code>str</code>
<code>11 / 2</code>	<code>5.5</code>	floating point number	<code>float</code>
<code>2 * 5 &gt; 9</code>	<code>True</code>	Boolean value	<code>bool</code>

# Sisentäminen (indentation)

---

Pythonin syntaksi vaatii ehdollisen lohkon sisällä olevien ilmaisuja sisentämistä yhdellä askeleella:

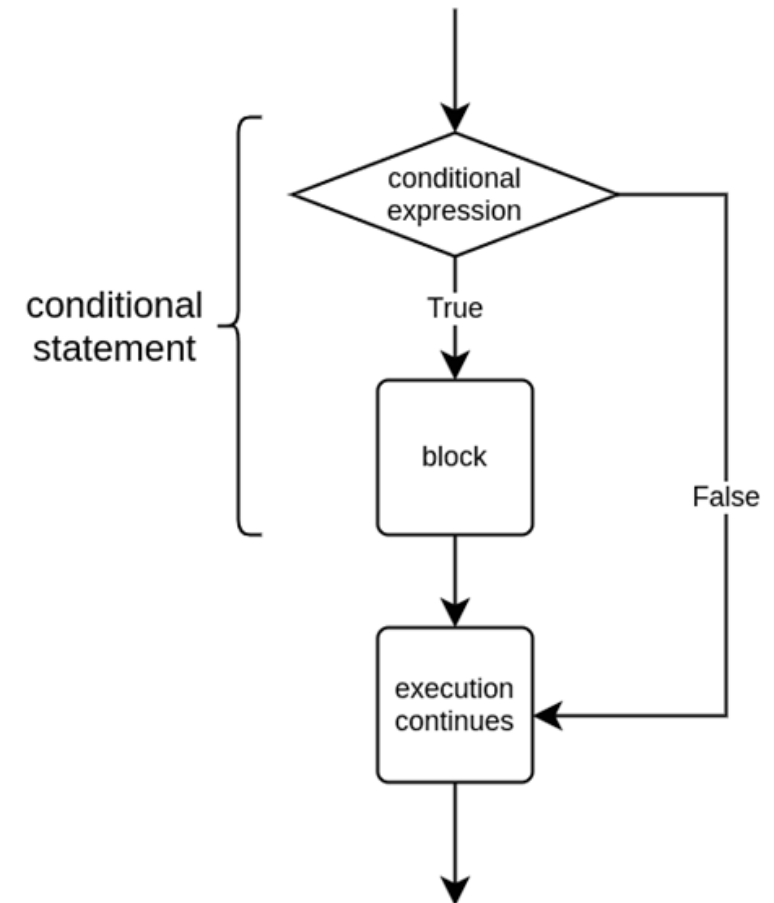
- Enter kaksoispisteen jälkeen
- Tab-näppäin
- Neljä välilyöntiä

```
example.py > ...
1 password = input("Please type in a password: ")
2
3 if password == "kittycat":
4     print("You knew the password!")
5     print("You must be either the intended user...")
6     print("...or quite an accomplished hacker.")
7
8 print("The program has finished its execution. Thanks and bye!")
9
```

# Ehdolliset lausekkeet

---

- ❑ Suoritusjärjestys: haarat
- ❑ Tarkistetaan onko ehto TRUE vai FALSE
  1. Jos ehto on TRUE, ehdollinen lohko koodia suoritetaan.
  2. Jos ehto on FALSE, ohjelma hyppää ehdollisen lohkon ohi ja jatkaa ohjelman suorittamista ehdollisen rakenteen jälkeen.



# if-lauseke

if (ehto):  
    ehdollisesti suoritettava lohko

- ❑ Mikäli ehto täyttyy, koodi koodi lausekkeen sisällä ajetaan.
- ❑ Jos ehto ei täyty, ohjelma jatkuu ehtolausekkeen jälkeen.
- ❑ Huomioi syntaksissa: kaksoispiste ehtolausekkeen jälkeen!

```
example.py X
example.py > ...
1  money = float(input("Enter amount of money you have: "))
2
3  if money >= 5:
4      print("You can buy a coffee.")
5

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
PS C:\Users\heidi\Desktop\Python\Learning_Python> & C:/Users/heidi/AppData/L...
on.exe c:/Users/heidi/Desktop/Python/Learning_Python/example.py
Enter amount of money you have: 6.2
You can buy a coffee.
PS C:\Users\heidi\Desktop\Python\Learning_Python> 
```

```
example.py X
example.py > ...
1  money = float(input("Enter amount of money you have: "))
2
3  if money >= 5:
4      print("You can buy a coffee.")
5

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
PS C:\Users\heidi\Desktop\Python\Learning_Python> & C:/Users/heidi/AppData/L...
on.exe c:/Users/heidi/Desktop/Python/Learning_Python/example.py
Enter amount of money you have: 4.5
PS C:\Users\heidi\Desktop\Python\Learning_Python> 
```

# Vertailuoperaattorit

---

- ❑ Voidaan ketjuttaa yhteen
- ❑ Huomioi **kaksi** 'yhtä kuin' -merkkiä

```
example.py > ...  
1 height = int(input("How tall are you? "))  
2  
3 if 170 <= height < 180:  
4     print("You are a tall!")  
5
```

Operator	Purpose	Example
<code>==</code>	Equal to	<code>a == b</code>
<code>!=</code>	Not equal to	<code>a != b</code>
<code>&gt;</code>	Greater than	<code>a &gt; b</code>
<code>&gt;=</code>	Greater than or equal to	<code>a &gt;= b</code>
<code>&lt;</code>	Less than	<code>a &lt; b</code>
<code>&lt;=</code>	Less than or equal to	<code>a &lt;= b</code>

# Loogiset operaattorit

❑ Monimutkaisemmille ehdoille, joissa voi olla monta yksittäistä ehtoa

❑ Tarkistusjärjestys:

- Not
- And
- Or
- Järjestystä voidaan muuttaa suluilla

Notation	Logical operator
and	"both"
or	"either" or "both"
not	negation, "no"

```
example.py X
example.py > ...
1  a = True
2  b = False
3
4  if a & b:
5      print("Both are true")
6
7  if a | b:
8      print("Either of them, or both, is true")
9
10 if not a & b:
11     print("Neither are true")
12
```



# Kaksi toisensa poissulkevaa vaihtoehtoa: *else* -haara

*if* (ehto):

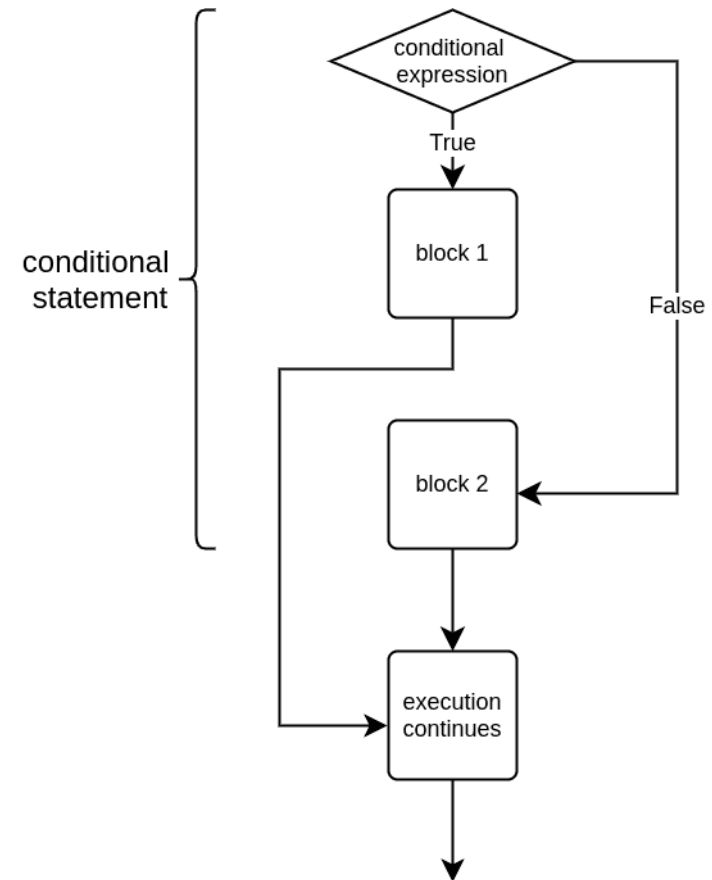
koodilohko, joka ajetaan jos ehto täyttyy

*else*:

koodilohko, joka ajetaan jos ehto ei täyty

- ❑ *else*-haaraa ei voi kirjoittaa ilman *if*-lauseketta
- ❑ *if-else* rakenteessa vain ja ainoastaan **yksi** haara ajetaan.

```
example.py
example.py > ...
1  number = int(input("Please, type in a number: "))
2
3  if number < 0:
4      print("The number is negative.")
5  else:
6      print("The number is positive.")
7
```

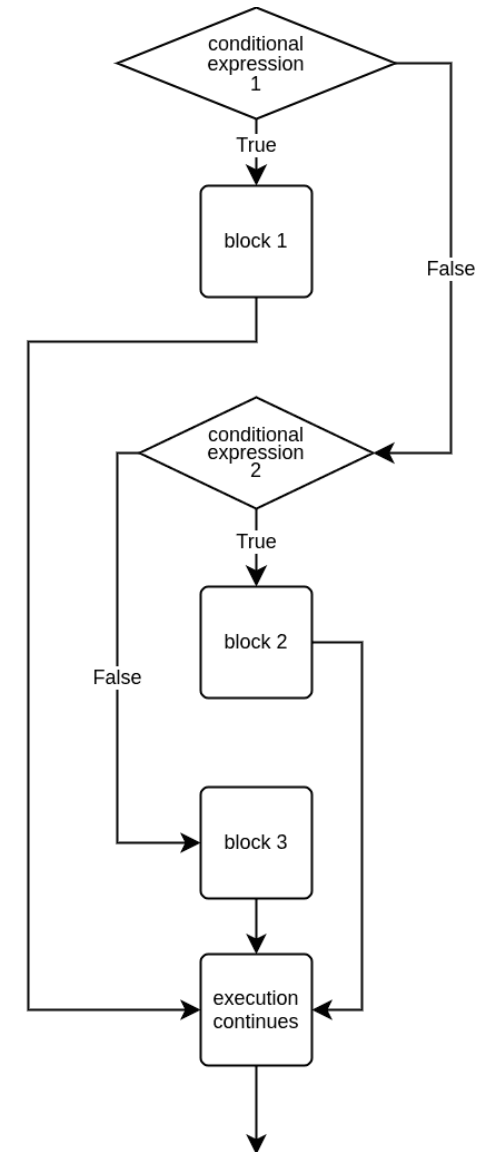


# Monta vaihtoehtoa: *elif* -haara

- ❑ Monta haaraa, joilla jokaisella oma ehto
- ❑ Ohjelma jatkaa haarojen tarkistamista, kunnes se löytää lausekkeen joka on tosi (True) tai se päättyy *else*-haaraan.
- ❑ *elif* -haaroja voi olla rajaton määrä
- ❑ Vain **yksi** haara ajetaan
- ❑ *else* -haara ei ole pakollinen

```
example.py •
example.py > ...
1 age = int(input("Enter your age: "))
2
3 if age >= 65:
4     print("You are retired.")
5 elif age >= 18:
6     print("You are working age.")
7 elif age >= 7:
8     print("You are in school.")
9 else:
10    print("You are a small child.")
11
```

conditional  
statement



# Sisäkkäiset ehdot

---

- ❑ Ehdollinen lauseke ehdollisen lausekkeen sisällä
- ❑ Sisennys erityisen tärkeää!
- ❑ Voi selkeyttää koodia, mutta voidaan usein myös korvata ketjuttamalla loogisia operaattoreita

```
example.py X
example.py > ...
1  number = int(input("Please type in a number: "))
2
3  if number > 0:
4      if number % 2 == 0:
5          print("The number is even")
6      else:
7          print("The number is odd")
8  else:
9      print("The number is negative or zero")
10
11
```

```
example.py X
example.py > ...
1  number = int(input("Please type in a number: "))
2
3  if number > 0 and number % 2 == 0:
4      print("The number is even")
5  elif number > 0 and number % 2 != 0:
6      print("The number is odd")
7  else:
8      print("The number is negative or zero")
```