Ehtolauseet

Ohjelmoinnin perusteet

4

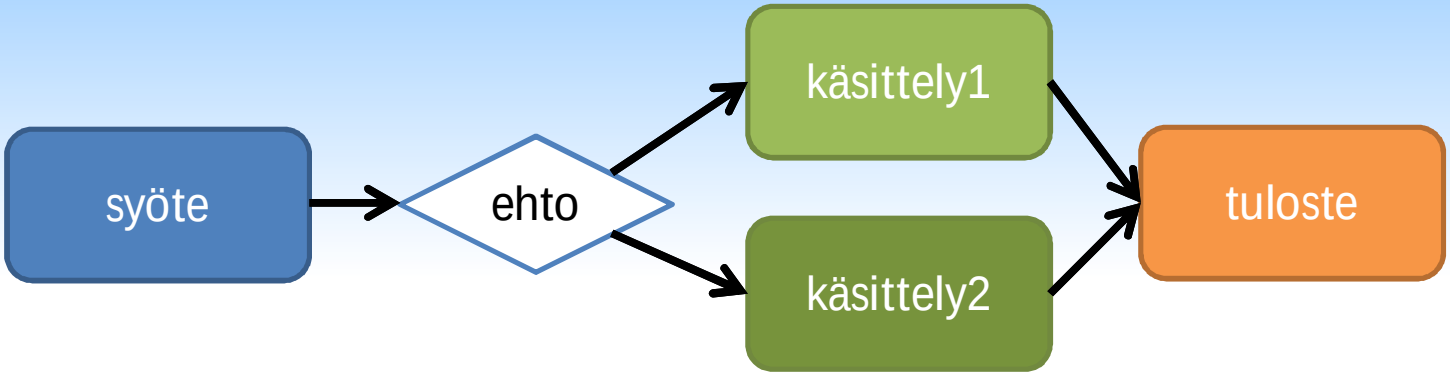
Turun yliopisto
University of Turku

Ohjelman suoritus

- Tähän mennessä kaikki kirjoitetut ohjelmat ovat noudattaneet samaa kaavaa:

Ehdollinen suoritus

- Ohjelmoitaessa on kuitenkin tyypillistä, että suoritus haarautuu esimerkiksi käyttäjän syötteeseen perustuen:



Ehdollinen suoritus

- Näin ollen ohjelma voi esimerkiksi laskea tuloksen negatiivisen syötteen pohjalta eri tavalla kuin positiivisen, tai jättää käsittelemättä sellaiset syötteet joiden avulla tulosta ei voi laskea.

Esimerkki

- Ohjelma voi pyytää käyttäjää syöttämään syntymävuoden, ja laskea iän tämän perusteella

- Jos annetaan virheellinen vuosi (esimerkiksi negatiivinen luku tai suurempi luku kun nykyinen vuosi), ohjelma voi tulostaa iän sijasta virheilmoituksen.

Ehdollinen suoritus Pythonissa

- Pythonissa suoritusta voidaan ohjata **ehtolauseella**.
- Ehtolausetta seuraava lohko suoritetaan **ainoastaan silloin, kun ehto on tosi.**
- Jos ehto on **epätosi**, lohkoa **ei suoriteta**.

Ehtolause: if

- Pythonin `if`-ehtolauseen syntaksi:

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

Ehtolauseke

- **Ehtolauseke** Pythonissa on mikä tahansa lauseke, joka evaluoituu arvoksi `True` tai `False`.

- `True` ja `False` (huomaa **iso alkukirjain)** ovat **totuusarvoja** (eivät siis esimerkiksi merkkijonoja), ja niitä voi käyttää sellaisenaan muuttujien arvoina.

Esimerkki ehtolauseesta

- Totuusarvoa voi suoraan käyttää ehtolauseessa:

```
totuus = True
if totuus:
    print "Totta on" # Tämä suoritetaan

ei_tosi = False
if ei_tosi:
    print "Tämä ei ole" # Tätä ei suoriteta
```

Lohkoista

- Ehtolausetta seuraava lohko voi Pythonissa sisältää yhden tai useamman lauseen.

- Lohko tunnistetaan **sisennyksestä**: kaikki rivit, joilla on sama sisennys (välilyönneillä tai sarkaimella) kuuluvat samaan lohkoon.

Esimerkki

```
totuus = True
if totuus:
    print "Totta on" # Tämä lohko suoritetaan
    print "Tai minut vesihiisi syököön"
    # Lohko päättyy tähän
print "Tämä tulostetaan aina ehdosta riippumatta!"
```

Ehtolauseke (2)

- Yleensä ehtolauseessa vertaillaan kahta **operandia** toisiinsa **vertailuoperaattoreita** käyttäen.

- Lauseke, jossa on vertailuoperaattori **evaluoituu** totuusarvoksi `True` tai `False`.

Pythonin vertailuoperaattoreita

- Eräitä Pythonin tukemia vertailuoperaattoreita:

Operaattori	Merkitys
==	Yhtäsuuri kuin
>	Suurempi kuin
<	Pienempi kuin
!=	Erisuuri kuin
>=	Suurempi tai yhtäsuuri kuin
<=	Pienempi tai yhtäsuuri kuin

Yhtäsuuri kuin

- Yhtäsuuruutta vertaillaan operaattorilla ==
- Huomaa ero asetusoperaattoriin =

Esimerkkejä:

```
a = 3 # Asetetaan a:n arvoksi 3
print a == 3 # Tulostaa True, a on yhtäsuuri kuin 3

if a + 2 == 5:
   print "Summa oli 5!" # Tulostetaan, oli tosi.
if a == 4: # Tämä on False
   print "Oli 4!" # Ei siis suoriteta
```

Erisuuri kuin

- Erisuuruusoperaattori != palauttaa totuusarvon `True`, jos vertailtavat operandit **eivät ole yhtä suuria**.

- **Esimerkiksi:**

```
a = input ("Anna luku:")
if a != 0: # Suoritetaan lohko jos ei ollut nolla
   a = a * 2
   print a
```

Suurempi / pienempi kuin

- Suurempi- tai pienempi kuin -operaattoreilla > ja < voidaan vertailla operandien suuruusjärjestystä.

```
a = 3 # Asetetaan a:n arvoksi 3
print a > 3 # Tulostaa False, a on yhtäsuuri kuin 3

if a + 2 <= 5:
    print "Oli pienempi!" # Tulostetaan, oli tosi.
if a > 2: # Tämä on False
    print "Oli suurempi!" # Tätä ei siis tulosteta
```

Merkkijonojen vertailu

- Myös merkkijonojen vertailu onnistuu samojen operaattoreiden avulla.

- Merkkijonoja vertaillaan merkkien arvon mukaan. Pienet ja isot kirjaimet ovat eriarvoisia (eli `"a" != "A"`).

Esimerkkejä

```
nimi = "Pekka"
print a == "Pek" + "ka" # Tulostaa True

if nimi < "Sami":
    print "On." # Tämä tulostetaan

if "aaa" != "AAA":
    print "Erisuuret" # Tämä tulostetaan myös

print "a" > "A" # Mitä tämä tulostaa?
```

Ehtolause: else

- `if`-lohkon jälkeen voidaan määrittää vaihtoehtoinen lohko, joka suoritetaan silloin kun **ehto on epätosi** (eli evaluoituu arvoksi False).

- Tällaisessa tapauksessa suoritetaan aina jompikumpi lohko.

Syntaksi

- Vaihtoehtoisesti suoritettava lohko merkitään avainsanalla `else`. Syntaksi:

```
if <ehtolauseke>:
    <suoritetaan, jos ehto tosi>
else:
    <suoritetaan, jos ehto epätosi>
```

Esimerkki

```
luku = input("Anna luku:")
if luku > 0:
   print "Luku oli suurempi kuin nolla."
else:
   print "Luku oli pienempi tai yhtäsuuri kuin 0"
```

Miksi else?

- Miksi tarvitaan erillinen else-lause sen sijaan, että käytettäisiin kahta if-lohkoa peräkkäin? Eli miksi tämä ei riitä:

```
if luku > 0:
    # Tee jotain tässä..
if luku <= 0:
    # Tee jotain muuta..
```

Vastaus

- Kaksi peräkkäistä if-lohkoa saattaa joissain tapauksissa johtaa siihen, että molemmat lohkot suoritetaan; else-haaraa käyttäessä näin ei tapahdu koskaan.

```
luku = input("Anna luku:")
if luku > 0:
    luku = luku - 3
    print luku
if luku <= 0:
    # Jos alkuperäinen luku <= 3, tämä suoritetaan myös
```

Ehtolause: elif

- On myös mahdollista liittää if-lohkon jälkeen lisää ehtolauseita, joiden totuus testataan vain silloin, kun alkuperäinen ehto on epätosi.

- Pythonissa tämä toteutetaan `elif`-lauseella. Näitä lohkoja voi olla `if`-lauseen jälkeen niin monta kuin tarvitaan. Lopuksi voidaan kirjoittaa `else`-lohko, mutta pakollista tämä ei ole.

Syntaksi

```
if <ehtolauseke 1>:
   <suoritetaan, jos ehto 1 tosi>
elif <ehtolauseke 2>:
   <suoritetaan, jos ehto 2 tosi>
else:
   <suoritetaan, jos kaikki ehdot epätosia>
```

Esimerkki

- Ohjelma kysyy käyttäjältä luvun, ja tulostaa tiedon siitä onko luku pienempi, suurempi vai yhtä suuri kuin nolla:

```
luku = input("Anna luku:")
if luku < 0:
   print "Negatiivinen luku"
elif luku > 0:
   print "Positiivinen luku"
else:
   # Tämä suoritetaan, jos molemmat aikaisemmat ehdot
   # olivat epätosia
   print "Luku oli nolla."
```

Esimerkki 2

```
kirjain = raw_input("Anna kirjain")
if len(kirjain) > 1:
   print "Annoit useamman kirjaimen!"
elif kirjain >= "a" and kirjain <= "ö":
   print "Pieni kirjain!"
elif kirjain >= "A" and kirjain <= "Ö":
   print "Iso kirjain!"
elif kirjain >= "0" and kirjain <= "9":
   print "Numero!"
else:
   print "Joku muu merkki."
```

Loogiset operaattorit

- Edellisessä esimerkissä oli yhdistetty useampi ehtoja yhteen ehtolausekkeeseen `and`-operaattorilla.

- Python tukee lisäksi muita loogisia operaattoreita, esimerkiksi `or` ja `not`.

Pythonin loogiset operaattorit

Operaattori	Esimerkki	Merkitys
`and`	**`a and b`**	Tosi, jos a tosi ja b tosi
`or`	**`a or b`**	Tosi, jos a tosi tai b tosi tai molemmat tosia
`not`	**`not a`**	a:n negaatio

and

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

or

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

not

a	not a
True	False
False	True

Esimerkki

```
luku = input("Anna luku välillä 1-10")
if luku < 1 or luku > 10:
   print "Luku ei ollut pyydetyltä väliltä!"

nimi = raw_input("Etu- ja sukunimesi (max. 30 mrk.):")
if nimi.find(" ") > -1 and len(nimi) <= 30:
   print "Nimi kelpaa..."
else:
   print "Ei kelpaa."
```