

Ohjelmoinnin perusteet Y Python

T-106.1208

25.1.2010

Valintakäsky if

- ▶ Tähänastiset ohjelmat ovat toimineen aina samalla tavalla. Usein ohjelman pitäisi kuitenkin muuttaa toimintaansa käyttäjän syötteen mukaan.
- ▶ Esimerkki: kirjoita ohjelma, joka pyytää käyttäjältä tentin pistemäärän ja kertoo, menikö tentti läpi, kun läpipääsyraja on 50 pistettä.
- ▶ Valinta voidaan tehdä `if`-käskyn avulla. Yleinen muoto:

```
if ehto:
    kasky1
else:
    kasky2
```

Tenttiesimerkki

```
def main():  
    syote = raw_input("Kerro tenttipisteesi.\n")  
    pisteet = int(syote)  
    if pisteet >= 50:  
        print "Tentti meni lapi!"  
    else:  
        print "Reputit!"  
  
main()
```

Toinen esimerkki: luvun itseisarvo

```
def main():  
    print "Ohjelma laskee desimaaliluvun itseisarvon."  
    rivi = raw_input("Anna luku.\n")  
    luku = float(rivi)  
    if luku < 0:  
        itseisarvo = - luku  
    else:  
        itseisarvo = luku  
    print "Sen itseisarvo on", itseisarvo  
  
main()
```

Useampi suoritettava käsky if-käskyssä

- Sisennyksillä osoitetaan, mitkä käskyt kuuluvat suoritettavaan vaihtoehtoon.

```
def main():  
    print "Ohjelma laskee desimaaliluvun itseisarvon."  
    rivi = raw_input("Anna luku.\n")  
    luku = float(rivi)  
    if luku < 0:  
        itseisarvo = - luku  
        print "Sen itseisarvo on", itseisarvo  
    else:  
        print "Sen itseisarvo on", luku  
  
main()
```

If-käsky ilman else-osaa

- ▶ Else-osa voi myös puuttua. Tällöin siirrytään suoraan ohjelmassa eteenpäin (if-käskyä seuraavaan käskyyn), jos ehto on epätosi.

```
def main():  
    print "Ohjelma laskee desimaaliluvun itseisarvon."  
    rivi = raw_input("Anna luku.\n")  
    luku = float(rivi)  
    if luku < 0:  
        luku = - luku  
    print "Sen itseisarvo on", luku  
  
main()
```

Vertailuoperaattoreita

- > suurempi kuin
- < pienempi kuin
- == yhtäsuuri kuin
- != erisuuri kuin
- >= suurempi tai yhtäsuuri kuin
- <= pienempi tai yhtäsuuri kuin

- ▶ Huomaa yhtäsuuruusoperaattorin == ja sijoitusoperaattorin = ero.
- ▶ Desimaalilukujen yhtäsuuruutta ei yleensä kannata tutkia, koska pyöristysvirheet voivat aiheuttaa yllätyksiä.

Loogiset operaattorit

- ▶ Halutaan lisätä tenttiesimerkkiin tarkistus: jos pisteet ovat alle 0 tai yli 100, ilmoitetaan virheellisistä pisteistä.
- ▶ Tarvittava ehto saadaan operaattorin `or` avulla:
`if pisteet < 0 or pisteet > 100:`
- ▶ Tai-operaattori `or` on yksi loogisista operaattoreista.

Loogiset operaattorit, jatkoa

- ▶ Tärkeimmät loogiset operaattorit
 - and ja
 - or tai
 - not ei
- ▶ Operaattorit and ja or ovat ehdollisia: jälkimmäisen operandin totuusarvo tutkitaan vain silloin, jos ensimmäisen operandin arvosta ei voida päätellä koko lauseen totuusarvoa.

Sisäkkäisiä if-käskyjä

```
def main():
    syote = raw_input("Kerro tenttipisteesi.\n")
    pisteet = int(syote)
    if pisteet < 0 or pisteet > 100:
        print "Virheelliset pisteet!"
    else:
        if pisteet >= 50:
            print "Tentti meni lapi!"
        else:
            print "Reputit!"

main()
```

Useita eri ehtoja

```
def main():
    syote = raw_input("Kerro tenttipisteesi.\n")
    pisteet = int(syote)
    if pisteet < 0:
        print "Kelvottomat pisteet!"
    elif pisteet < 50:
        print "Arvosana on 0."
    elif pisteet < 60:
        print "Arvosana on 1."
    elif pisteet < 70:
        print "Arvosana on 2."
    elif pisteet < 80:
        print "Arvosana on 3."
    # Ohjelma jatkuu seuraavalla kalvolla.
```

Useita eri ehtoja (jatkuu)

```
elif pisteet < 90:  
    print "Arvosana on 4."  
elif pisteet <= 100:  
    print "Arvosana on 5."  
else:  
    print "Kelvottomat pisteet!"
```

```
main()
```

Toistokäsky

- ▶ Usein ohjelma joutuu toistamaan samaa käskyä monta kertaa.
- ▶ Esimerkki: käyttäjä syöttää ohjelmalle 30 päivän lämpötilat. Ohjelma laskee niiden keskiarvon
- ▶ Samaa käskyä ei tarvitse kirjoittaa monta kertaa, jos käytetään *toistokäskyä*.
- ▶ Pythonissa on kaksi toistokäskyä, `while` ja `for`.
- ▶ While-käskyn yleinen muoto on

```
while ehto:  
    kasky
```

Kuuden kertotaulu while-käskyn avulla

```
def main():  
    i = 1  
    while i <= 10:  
        tulos = i * 6  
        print i, "* 6 =", tulos  
        i += 1
```

```
main()
```

Esimerkki: lämpötilojen keskiarvo

```
def main():  
    LKM = 30  
    i = 0  
    summa = 0.0  
    while i < LKM:  
        rivi = raw_input("Anna seuraava lampotila.\n")  
        lampotila = float(rivi)  
        summa = summa + lampotila  
        i = i + 1  
    keskiarvo = summa / LKM  
    print "Lampotilojen keskiarvo on", keskiarvo  
  
main()
```

Toinen versio lämpötilalaskurista: lukumäärä käyttäjältä

```
def main():  
    rivi = raw_input("Montako lampotilaa annat?\n")  
    lkm = int(rivi)  
    i = 0  
    summa = 0.0  
    while i < lkm:  
        rivi = raw_input("Anna seuraava lampotila.\n")  
        lampotila = float(rivi)  
        summa = summa + lampotila  
        i = i + 1  
    if lkm > 0:  
        keskiarvo = summa / lkm  
        print "Lampotilojen keskiarvo on", keskiarvo  
    else:  
        print "Et antanut yhtään lampotilaa."
```

Kolmas versio lämpötilalaskurista: lopetus pienellä luvulla

```
def main():
    print "Anna lampotiloja, lopeta -300:lla."
    lkm = 0
    summa = 0.0
    rivi = raw_input("Anna ensimmäinen lampotila.\n")
    lampotila = float(rivi)
    while lampotila > -300.0:
        summa = summa + lampotila
        lkm = lkm + 1
        rivi = raw_input("Anna seuraava lampotila.\n")
        lampotila = float(rivi)
    if lkm > 0:
        keskiarvo = summa / lkm
        print "Lampotilojen keskiarvo on", keskiarvo
    else:
        print "Et antanut yhtään lampotilaa."
```

If-käsky toistokäskyn sisällä

```
def main():
    HELLERAJA = 25.0
    print "Anna lampotiloja, lopeta -300:lla."
    hellepaivien_lkm = 0
    rivi = raw_input("Anna ensimmäinen lampotila.\n")
    lampotila = float(rivi)
    while lampotila > -300.0:
        if lampotila >= HELLERAJA:
            hellepaivien_lkm += 1
            rivi = raw_input("Anna seuraava lampotila.\n")
            lampotila = float(rivi)
    print "Hellepaivia oli", hellepaivien_lkm, "kpl."
```

main()

Toistokäsky for

- ▶ For-käskyn avulla voidaan käydä jonkun rakenteen alkiot järjestyksessä.
- ▶ Funktion `range` avulla voidaan generoida lukujono (oikeasti lista), jonka kaikki alkiot voidaan käydä läpi for-käskyn avulla.
- ▶ Esimerkiksi `range(11)` generoi lukujonon, joka sisältää luvut 0:sta 10:een asti.
- ▶ Voidaan kirjoittaa toistokäsky

```
for i in range(11):  
    tee jotain arvolle i
```

jossa i saa vuorotellen arvot 0–10.

Esimerkki: kertotaulu for-käskyn avulla

```
def main():  
    for i in range(11):  
        tulos = i * 6  
        print i, "* 6 =", tulos  
  
main()
```

Lisää range-funktiosta

- ▶ Lukujonoja generoidessa voi antaa myös jonon ensimmäisen alkion, esimerkiksi `range(1, 11)` generoi lukujonon, jonka ensimmäinen alkio on 1 ja viimeinen 10.
- ▶ On myös mahdollista antaa jonon kahden peräkkäisen alkion väli, esimerkiksi `range(1, 11, 2)` generoi lukujonon, joka sisältää luvut 1, 3, 5, 7 ja 9.
- ▶ Väli voi olla myös negatiivinen, jolloin lukuja generoidaan ylhäältä alaspäin. Esimerkiksi `range(10, 1, -2)` generoi lukujonon, joka sisältää luvut 10, 8, 6, 4 ja 2.

Kertotaulu uudelleen

```
def main():  
    for i in range(1, 11):  
        tulos = i * 6  
        print i, "* 6 =", tulos
```

```
main()
```