

# Ohjelmoinnin perusteet Y Python

T-106.1208

3.2.2010

## Esimerkki: asunnon välityspalkkio

- ▶ Kirjoitetaan ohjelma, joka laskee kiinteistönvälittäjän asunnon myynnistä perimän palkkion.
- ▶ Välityspalkkion on 4.5 % asunnon myyntihinnasta, kuitenkin vähintään 2500 euroa.
- ▶ On siis laskettava, kumpi mainituista summista on suurempi.
- ▶ Tähän käytetään funktiota `maximi`.

# Asunnon välityspalkkio, koodi

```
def maksimi(luku1, luku2):  
    if luku1 >= luku2:  
        return luku1  
    else:  
        return luku2
```

## Asunnon välityspalkkio, koodi jatkuu

```
def main():  
    PALKKIOPROSENTTI = 4.5  
    MINIMIPALKKIO = 2500  
    syote = raw_input("Anna asunnon myyntihinta.\n")  
    hinta = float(syote)  
    palkkio = maksimi(MINIMIPALKKIO, \  
                      PALKKIOPROSENTTI / 100.0 * hinta)  
    print "Valityspalkkio on %.2f euroa." % (palkkio)
```

```
main()
```

# Totuusarvon palauttavat funktiot

- ▶ Funktio voi myös palauttaa totuusarvon True tai False
- ▶ Seuraava esimerkkiohjelma laskee käyttäjän iän tämän syntymävuoden perusteella.
- ▶ Ohjelmaan on lisätty funktio `onko_kelvollinen`, joka tutkii, onko sille parametrina annettu syntymävuosi hyväksytyllä välillä 1890–2010.
- ▶ Funktion palauttamaa arvoa voitaisiin käyttää esimerkiksi if-käskyn ehdossa seuraavasti:

```
if onko_kelvollinen(syntymavuosi) == True:
```

- ▶ Vertailu arvoon True on kuitenkin tarpeeton, koska funktio itsessään palauttaa arvon True tai False. Sen vuoksi ehto voidaan kirjoittaa lyhyemmin:

```
if onko_kelvollinen(syntymavuosi):
```

# län laskeminen, koodi

```
NYKYINEN_VUOSI = 2010
```

```
def onko_kelvollinen(vuosi):  
    ALARAJA = 1890  
    if vuosi < ALARAJA or vuosi > NYKYINEN_VUOSI:  
        return False  
    else:  
        return True
```

## län laskeminen, koodi jatkuu

```
def main():
    print "Ohjelma laskee ikasi."
    rivi = raw_input("Anna syntymavuotesi.\n")
    syntymavuosi = int(rivi)
    if onko_kelvollinen(syntymavuosi):
        ika = NYKYINEN_VUOSI - syntymavuosi
        print "Ikasi on", ika, "vuotta."
    else:
        print "Virhe syntymavuodessa."

main()
```

# Moduuli `math`

- ▶ Pythonissa on valmiina suuri joukko funktioita erilaisten toimintojen tekemiseen.
- ▶ Suurin osa näistä funktioista on jaettu moduuleihin. Yksi moduuli sisältää tyypillisesti samaan asiaan liittyviä funktioita ja mahdollisesti myös vakioita.
- ▶ Yksi tärkeä moduuli on `math`, joka sisältää joukon matemaattisia funktioita sekä vakiot `math.pi` ja `math.e`.
- ▶ Jotta moduulin vakioita tai funktioita voisi käyttää, on ohjelmatiedoston alkuun kirjoitettava  

```
import math
```

# Tärkeitä math-moduulin funktioita

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| <code>ceil(x)</code>  | pienin kokonaisluku, joka on suurempi kuin $x$ . |
| <code>floor(x)</code> | suurin kokonaisluku, joka on pienempi kuin $x$ . |
| <code>sqrt(x)</code>  | neliöjuuri                                       |
| <code>exp(x)</code>   | $e$ potenssiin $x$ .                             |
| <code>log(x)</code>   | luonnollinen logaritmi.                          |
| <code>log10(x)</code> | 10-kantainen logaritmi.                          |
| <code>cos(x)</code>   | kosini.                                          |
| <code>sin(x)</code>   | sini.                                            |
| <code>tan(x)</code>   | tangentti.                                       |

- Huomaa, että funktoiden `ceil` ja `floor` paluuarvon tyyppi on `float`.

## Esimerkki: hypotenuusan pituus

```
import math

def laske_hypotenuusa(a, b):
    c = math.sqrt(a * a + b * b)
    return c

def main():
    rivi = raw_input("Anna 1. sivun pituus.\n")
    sivu1 = float(rivi)
    rivi = raw_input("Anna 2. sivun pituus.\n")
    sivu2 = float(rivi)
    sivu3 = laske_hypotenuusa(sivu1, sivu2)
    print "Hypotenuusan pituus on %.2f" % (sivu3)

main()
```

## Esimerkki: toisen asteen yhtälön ratkaisu

- ▶ Kirjoitetaan ohjelma, joka ratkaisee toisen asteen yhtälön  $ax^2 + bx + c = 0$
- ▶ Käytetään ratkaisukaavaa

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- ▶ Ohjelma tulostaa vain reaali juuret.
- ▶ Ratkaisujen lukumäärää tutkitaan neliöjuuren alla olevan diskriminantin arvon perusteella.

## Toisen asteen yhtälö, koodi

```
import math

def ratkaise_yhtalo(a, b, c):
    diskrim = b * b - 4 * a * c
    if diskrim < 0:
        print "Ei reaalijuuria."
    elif diskrim == 0:
        x = -1.0 * b / (2 * a)
        print "Yhtalon ratkaisu on %f." % (x)
    else:
        x1 = (-1.0 * b + math.sqrt(diskrim)) / (2 * a)
        x2 = (-1.0 * b - math.sqrt(diskrim)) / (2 * a)
        print "Ratkaisut ovat %f ja %f." % (x1, x2)
```

## Toisen asteen yhtälö, koodi jatkuu

```
def main():
    print "Anna 2. asteen yhtalon kertoimet a, b ja c."
    rivi = raw_input()
    eka_kerroin = int(rivi)
    rivi = raw_input()
    toka_kerroin = int(rivi)
    rivi = raw_input()
    vakio = int(rivi)
    if eka_kerroin == 0:
        print "Yhtalo ei ole toista astetta."
    else:
        ratkaise_yhtalo(eka_kerroin, toka_kerroin, vakio)

main()
```

# Useampi paluuarvo

- ▶ Funktio voi palauttaa samanaikaisesti myös useamman kuin yhden arvon.
- ▶ Palautettavat arvot erotetaan toisistaan return-käskyssä pilkulla:  
`return arvo1, arvo2, arvo3`
- ▶ Arvot voidaan ottaa *samanaikaisella sijoituksella* talteen siellä, missä funktiota kutsutaan:  
`muuttuja1, muuttuja2, muuttuja3 = funktio(parametrit)`

## Esimerkki: nopeuden yksikön muutos

- ▶ Monet englanninkieliset urheilulehdet ja Internet-sivut antavat harjoitusohjelmissa ohjeellisen nopeuden mailia kohti.
- ▶ Kirjoitetaan ohjelma, joka muuntaa annetun mailinopeuden nopeudeksi kilometriä kohti.
- ▶ Muunnoksen tekevä funktio palauttaa erikseen kilometrinopeuden minuutit ja sekunnit, jotta ne voitaisiin tulostaa siististi.

# Nopeusyksikkömuunnos, koodi

```
def muuta_km_vauhdiksi(minuutit, sekunnit):  
    MAILIKERROIN = 1.609344  
    vauhti = int ((minuutit * 60 + sekunnit) / MAILIKERROIN)  
    km_minuutit = vauhti / 60  
    km_sekunnit = vauhti % 60  
    return km_minuutit, km_sekunnit  
  
def main():  
    rivi = raw_input("Anna mailinopeuden minuutit: ")  
    maili_min = int(rivi)  
    rivi = raw_input("Anna mailinopeuden sekunnit: ")  
    maili_sek = int(rivi)  
    km_min, km_s = muuta_km_vauhdiksi(maili_min, maili_sek)  
    print "Nopeus on %d min %d s/km." % (km_min, km_s)  
  
main()
```

# Valikkopohjaisen ohjelman kirjoittaminen

- ▶ Hyvin yleinen tehtävä on kirjoittaa ohjelma, jossa käyttäjä voi valita valikosta haluamansa toiminnon useasta vaihtoehdosta.
- ▶ Käyttäjä voi valita aina uuden toiminnon niin kauan, kunnes hän haluaa lopettaa.
- ▶ Seuraava esimerkki näyttää, miten tällainen ohjelma kannattaa kirjoittaa.
- ▶ Valikon tulostamista ja käyttäjän valinnan pyytämistä varten kirjoitetaan oma funktio.
- ▶ Myös jokaista toimintoa varten kirjoitetaan oma funktio (nämä funktiot voivat mahdollisesti vielä kutsua muita funktioita).
- ▶ Pääohjelmaan kirjoitetaan toistokäske, joka kutsuu käyttäjän valintaa pyytävää funktiota ja valinnan mukaan aina jonkun toiminnon toteuttavaa funktiota.
- ▶ Toistokäske loppuu, kun käyttäjä valitsee ohjelman lopettamisen.
- ▶ Seuraavassa esimerkissä käyttäjä voi laskea erilaisten kuvioiden pinta-aloja.

# Valikko-ohjelma, koodi

```
import math

def valitse_toiminto():
    print
    print "Valitse toiminto:"
    print "1. laske ympyran pinta-ala"
    print "2. laske suorakulmion pinta-ala"
    print "3. laske kolmion pinta-ala"
    print "4. lopeta"
    vastaus = raw_input()
    return int(vastaus)
```

# Valikko-ohjelma, koodi

```
def ympyran_ala():  
    rivi = raw_input("Anna ympyran sade.\n")  
    sade = float(rivi)  
    ala = math.pi * sade * sade  
    print "Pinta-ala on %.4f." % (ala)  
  
def suorakulmion_ala():  
    rivi = raw_input("Anna suorakulmion leveys.\n")  
    leveys = float(rivi)  
    rivi = raw_input("Anna suorakulmion korkeus.\n")  
    korkeus = float(rivi)  
    ala = leveys * korkeus  
    print "Pinta-ala on %.4f." % (ala)
```

# Valikko-ohjelma, koodi

```
def kolmion_ala():  
    rivi = raw_input("Anna kolmion alisivun leveys.\n")  
    leveys = float(rivi)  
    rivi = raw_input("Anna kolmion korkeus.\n")  
    korkeus = float(rivi)  
    ala = 0.5 * leveys * korkeus  
    print "Pinta-ala on %.4f." % (ala)
```

# Valikko-ohjelma, koodi

```
def main():  
    valinta = valitse_toiminto()  
    while valinta != 4:  
        if valinta == 1:  
            ympyran_ala()  
        elif valinta == 2:  
            suorakulmion_ala()  
        elif valinta == 3:  
            kolmion_ala()  
        valinta = valitse_toiminto()  
    print "Ohjelman suoritus paattyy"
```

```
main()
```