

# Python-ohjelmointi

# Harjoitus 15

## TAVOITTEET

- Kerrataan *random*-kirjaston käyttämistä ja silmukkarakenteita.
- Kerrataan pöytätestaus.
- Tutustutaan *datetime*- ja *time*-kirjastoihin.

## Matematiikan sisällöt

Tehtävissä harjoitellaan riippuvuutta sekä funktioon liittyviä käsitteitä.

Ohjelmointiin käytettävä alusta löytyy osoitteesta <http://sade-oppiminen.herokuapp.com/>

## KERTAUSTA HARJOITUKSESTA 14

### TEHTÄVÄ 1

Tavoitteena on palauttaa mieleen ohjelman testaaminen ilman tietokonetta eli pöytätestaus.

Run

Tutki oheisen ohjelman toimintaa vaihe vaiheelta. Mitä arvoja muuttuja saa ohjelman eri vaiheissa ja mitä ohjelma lopulta tulostaa?

```
def f(x):  
    y = 2*x - 4  
    print y  
  
f(3)  
f(-2)  
f(0)  
f(-10)
```

## AIKAISEMPIEN SISÄLTÖJEN KERTAUSTA

Seuraavissa tehtävissä palautetaan mieliin niin listojen käyttämistä, ohjelman suunnitteluvaiheita kuin satunnaiskirjasto *randomin* käyttöä.

### TEHTÄVÄ 2

Tavoitteena on palauttaa mieleen silmukkarakenne ja listojen käyttö tiedon tallennuksessa.

Run

Tee ohjelma, joka tallentaa ja lopuksi tulostaa listan lukujonon kymmenestä ensimmäisestä jäsenestä. Lukujono muodostuu säännöllä ”seuraava jäsen on aina kolme suurempi kuin edellinen”. Lukujonon uusi jäsen lisätään aina listaan. Lukujonon ensimmäinen jäsen on luku 1.

### TEHTÄVÄ 3

Tavoitteena on palauttaa mieleen satunnaiskirjaston käyttö.

Run

Tee ohjelma, joka hyödyntää Yatzy-pelin ideaa. Yatzyssa heitetään viittä noppaa ja pyritään saamaan tiettyjä yhdistelmiä. Noppia heitetään kolmesti ja kierrosten välissä niistä voidaan ottaa osa jo sivuun halutun yhdistelmän varmistamiseksi.

Toisin kun oikeassa Yatzyssä, tässä ohjelmassa yhtä noppaa heitetään viidesti ja viiden heiton muodostama rivi on ns. lopputulos.

Tee tyhjä lista, johon arvotut ”nopan silmäluvut” tallennetaan. Nimeä lista sopivalla nimellä.

”Noppaa” heitetään viisi kertaa ja jokaisen heittokerran tulos tallennetaan listaan. Lopuksi tulostetaan muodostunut lista.

### TEHTÄVÄ 4

Tavoitteena on vahvistaa ohjelman suunnittelun taitoja.

Run

Tee ohjelma, jonka avulla saat laskettua ohjelmassa määritellylle funktiolle funktion arvoja. Ohjelma tulostaa  $x$ :n arvon sekä sitä vastaavan funktion arvon. Tee ensin suunnitelma ohjelman rakenteesta ja kirjoita ohjelmakoodi suunnitelmaa noudattaen. Ohjelmaa voi hyödyntää esimerkiksi piirrettävien kuvaajien  $(x,y)$ -pisteerien laskennassa.

Käytä ohjelman testaamiseen kirjassa esiintyviä funktioita tai keksi omia funktioita.

## PÄIVÄAIKA- SEKÄ AIKA-KIRJASTOT

Päivä- ja aikamodulin avulla voidaan käsitellä päivämääriä ja kellonaikoja.

### ***Datetime*-kirjaston komentoja**

aika nyt eli päivämäärän ja ajan yhdistelmä `datetime.now()`

mikrosekuntien tarkkuudella

päivämäärän syöttäminen

`datetime(vvvv, kk, pp, hh, mm, ss)`

Päivämääriä syötettäessä pitää tiedot antaa yllä ilmoitetussa järjestyksessä. Mikäli päivämäärää annettaessa ei ilmoiteta esimerkiksi syntymäaikaa kellonajan tarkkuudella (tunnit, minuutit, sekunnit), funktio määrittää ne oletusarvoisesti nollassi.

Päivämäärillä voi suorittaa laskutoimituksia kuten luvuilla. Esim. ikää voi selvittää vähennyslaskun avulla. Päivämäärien erotus ilmaisee vastauksen päivien ja millisekuntien tarkkuudella. Tuloste on muotoa *days, hh:mm:ss.microseconds*.

### Esimerkki 1

Tee ohjelma, joka tulostaa

- päivän ja ajan nyt
- päiväyksestä tulostetaan erikseen vuosiluku, kuukausi, päivä, tunnit, minuutit ja sekunnit.

### Ratkaisu

Ohjelman alussa lisätään kirjasto ja määritellään muuttuja, joka saa arvoksi tämän hetken ajankohdan.

```
from datetime import*
# Määritetään muuttuja tätä hetkeä varten.
nyt = datetime.now()

# a)
#Tulostetaan tämän hetkinen ajankohta
print "Nyt on:", nyt
# b)
#Tulostetaan vain vuosiluku
print "Nyt on vuosi", nyt.year
#Tulostetaan vain kuukausi
print "Kuukausi on nyt", nyt.month
#Tulostetaan vain päivä
print u"Päivä", nyt.day
#Tulostetaan tunnit
print"Tunnit:", nyt.hour
#Tulostetaan minuutit
print"Minuutit:", nyt.minute
#Tulostetaan sekunnit
print"Sekunnit:", nyt.second
```

Ohjelman tulostus näyttää seuraavalta.

```
Nyt on: 2019-07-23
09:10:08.242100
Nyt on vuosi 2019
Kuukausi on nyt 7
Päivä 23
Tunnit: 9
Minuutit: 10
Sekunnit: 8
```

## TEHTÄVÄ 5

Tavoitteena on harjoitella *datetime*-kirjaston käyttämistä.

Run

Tee ohjelma, jolla saat tulostettua

- tämän hetken ajankohdasta pelkän päiväyksen muodossa *pp.kk.vvvv*.
- tämän hetken ajankohdasta pelkän ajan muodossa *hh:mm:ss*.
- oman syntymäaikasi.

## TEHTÄVÄ 6

Tavoitteena on syventää *datetime*-kirjaston käyttämistä.

Run

Tee ohjelma, joka laskee, kuinka vanha olet. Ohjelmalle tarvitsee antaa syntymäpäivästä vähintään päivä, kuukausi ja vuosi. Syntymäpäivän ja tämän hetken ajankohdan avulla saadaan laskettua nykyinen ikä.

Tulosta

- ikä tarkasti
- ikä pelkkinä päivinä
- Laske ja tulosta ikä vuosina. (Vuodessa 365 päivää).

## TEHTÄVÄ 7

Tavoitteena on soveltaa *datetime*-kirjastoa ja palauttaa mieliin aliohjelmine käyttö.

Run

Tee ohjelma, jossa laajennat ja muokkaat edellisen kohdan ohjelmaa seuraavasti

- Käyttäjältä kysytään päivämäärä, kuukausi ja vuosiluku.
- Iän laskemisesta tehdään oma aliohjelma.
- Omissa aliohjelmissaan selvitetään
  - o ikä vuosina
  - o kokonaisten vuosien ylittävien kuukausien määrä
  - o kokonaisten kuukausien ylittävien päivien määrä
- Ohjelma tulostaa tiedon: Ikäsi on \_\_\_vuotta, \_\_\_kuukautta ja \_\_\_päivää.

## Time-kirjaston käyttö

*Time*-kirjaston avulla saadaan esimerkiksi käyttöön kello mittaamaan esimerkiksi suoritukseen kuluvaa aikaa. mitattua aikaa. Kello saadaan kutsuttua komennolla *clock()*.

## TEHTÄVÄ 8

Tavoitteena on harjoitella *time*-kirjaston käyttöä.

Run

Tee ohjelma, jossa käyttäjälle esitetään 10 kysymystä. Kysymykset voivat olla funktion arvon laskemista tai kertotaulutesti. Vastaamiseen käytettyä aikaa mitataan. Lopuksi käyttäjälle kerrotaan, kuinka monta hän sai oikein ja kuinka kauan aikaa kului vastaamiseen.

- Lisää ohjelman alussa *time*-kirjasto.
- Tallenna muuttujaan *aika1* aloitusaika eli muuttuja *aika1* saa arvokseen ***time.clock()***-funktion arvon.
- Esitetään kysymykset ja kootaan oikeiden vastausten määrä.
- Tallennetaan lopussa muuttujaan *aika2* lopetusaika.
- Laske vastaamiseen käytetty aika.
- Tulostetaan *Sait oikein \_\_\_\_ kysymystä ja Vastaamiseen kului \_\_\_\_ sekuntia.*