

## Python-ohjelmointi

## Harjoitus 8

## TAVOITTEET

- Kerrataan harjoituksen 7 keskeisiä sisältöjä.
- Opitaan rakenne *lista*.
- Opitaan hyödyntämään silmukan loppuarvoa ja jatkamaan ohjelmaa silmukan ulkopuolella.
- Opitaan käyttämään satunnaisuuden mahdollistavaa random-kirjastoa.

## Matematiikan sisällöt

Tehtävissä harjoitellaan ja syvennetään todennäköisyyslaskennan sekä tilastomatematiikan osaamista ja vahvistetaan prosenttilaskennan taitoja.

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

## KERTAUSTA HARJOITUKSESTA 7

## TEHTÄVÄ 1

Tavoitteena on tehdä ohjelma, jossa kerrataan round-komennon käyttämistä.

Run

Tee ohjelma, joka pyöristää annetut luvut pyydettyyn tarkkuuteen ja tulostaa sekä lähtöarvon että pyöristetyn vastauksen.

Alkuperäinen luku	Vastaustarkkuus
0,5678904	3 desimaalia
13,3492	1 desimaali
467,28	kymmenien tarkkuus
23 840 256	miljoonien tarkkuus

## TEHTÄVÄ 2

Tavoitteena on tehdä ohjelma, jossa kerrataan while-toistorakennetta.

Run

Tee ohjelma, joka hyödyntää *while*-silmukkarakennetta ja tulostaa käyttäjän antamien tietojen perusteella havainnollistavan kuvion pehmolelujen, autojen ja pelien lukumääristä.

Pehmolelut ### ## IIII  
Autot ### ## ##  
Pelit ### II

- Määritä *muuttuja* ja sille alkuarvoksi nolla.
- Kirjoita *while*-silmukkaan ehto, että alkuarvon ollessa suurempi kuin  $-0,1$ , silmukan sisältö toteutetaan.
- Silmukan sisällä määritetään, että muuttuja saa arvon käyttäjän vastauksesta kysymykseen ”Anna lelujen lukumäärä”. Kysymykseen voi lisätä ohjeistuksen, että negatiivinen luku lopettaa ohjelman toiminnan ja tulostaa kuvion.
- Määriä havainnollistetaan X-merkillä. Tulostuskomennolla määritetään, että X-merkkiä piirretään muuttujan määrittämä määrä. Jos muuttujan nimi on esim. *a*, merkki X piirretään *a* kertaa komennolla `print a * 'X'`.

## LISTAT

Muuttujia käytetään yksittäisen tiedon tallentamiseen. Kun tallennettavana on useita yksittäisiä tietoja ja ne halutaan tallentaa samaan paikkaan, tiedot tallennetaan listana. Lista voidaan tallentaa lukuja, merkkijonoja, listoja tai näiden yhdistelmiä.

Listan määrittäminen aloitetaan antamalla sille nimi. Nimeämiseen soveltuvat samat periaatteet kuin muuttujien nimeämiseen.

Lista kirjoitetaan hakasulkeiden [ ] sisään. Listan yksittäistä tietoa kutsutaan alkioiksi. Alkiot erotetaan toisistaan pilkulla. Tekstimuotoiset tiedot kirjoitetaan puolilainausmerkkien ' ' sisään. Jokainen alkio saa järjestysnumeron eli indeksin, jolla alkioon voidaan myöhemmin viitata. Python aloittaa laskemisen luvusta nolla eli esimerkiksi ensimmäinen alkio saa indeksin 0, toinen indeksin 1 ja kolmas indeksin 2.

### Listat

```
lista = ['alkio1', 'alkio2', 'alkio3']
indeksi 0      1      2
```

### Listalle tehtävät perusoperaatiot

Listojen yhdistäminen	$lista1 + lista2$
Listan toistaminen neljä kertaa	$lista1 * 4$
Listan pituus	$len(lista1)$
Alkion lisääminen listaan	$lista1.append(lisättävä\_alkio)$
Alkion tietojen muuttaminen	$lista1[muutettavan\_alkion\_indeksi] = muutos$

### Esimerkki 1

Tee ohjelma, jolla voit muodostaa ja tulostaa listan tilastoihin liittyvistä nimityksistä.

- Tee lista, joka sisältää nimitykset *frekvenssi*, *keskiarvo*, *moodi* ja *mediaani*. Tulosta koko lista ja listan pituus.
- Tulosta listan ensimmäinen ja viimeinen alkio.
- Lisää listaan nimitys *vaihteluväli* ja tulosta lista.
- Muuta moodin tilalle *tyyppiarvo* ja tulosta lista.
- Tulosta listan alkiot omille riveilleen *for*-silmukan avulla.

**Ratkaisu**

a)

```
# Annetaan listalle nimeksi nimitykset.
# Listan alkioita ovat frekvenssi, keskiarvo, mediaani ja moodi.
nimitykset = ['frekvenssi', 'keskiarvo', 'mediaani', 'moodi']
# Tulostetaan koko lista.
print nimitykset
a = len(nimitykset)
print a
```

b) Listan alkion saa tulostettua hyödyntämällä alkion indeksii. Jos lista on lyhyt, viimeisen alkion indeksin voi selvittää laskemalla. Listassa on neljä alkioita. Koska ensimmäisen alkion indeksi on aina 0, viimeisen alkion indeksi on  $4 - 1 = 3$ .

```
# Tulostetaan listan ensimmäinen ja viimeinen alkio.
print nimitykset[0]
print nimitykset[3]
```

Jos lista on pitkä, listan pituus kannattaa selvittää *len*-komennolla. Viimeisen alkion indeksi on listan pituus - 1.

```
# Selvitetään listan pituus len-komennolla ja tulostetaan listan
viimeinen alkio. Listan viimeisen alkion indeksi on listan pituus
vähennettynä yhdellä.
print nimitykset[len(nimitykset) - 1]
```

c) Lista voidaan lisätä uusia alkioita *append*-komennolla. Tällöin uusi alkio lisätään olemassa olevan listan loppuun. Lista tulostetaan lisäämisen jälkeen.

```
# Lisätään listan loppuun alkio vaihteluväli. Muista lisätä u ennen
puolilainausmerkkiä, jotta ohjelma hyväksyy ä-kirjaimen.
nimitykset.append(u'vaihteluväli')
# Tulostetaan uusi lista.
print nimitykset
```

d) Muutetaan nimitys *moodi* muotoon *tyyppiarvo*. Tähän tarvitaan sen alkion indeksi, joka halutaan muuttaa ja tieto, mitä alkion tilalle laitetaan. Lista tulostetaan muutoksen jälkeen.

```
# Moodin indeksi on 3. Muutetaan moodin tilalle tyyppiarvo.
nimitykset[3] = 'tyyppiarvo'
# Tulostetaan uusi lista.
print nimitykset
```

e) Määritetään apumuuttuja  $n$ , jonka arvo on listan pituus. For-rakenne tarvitsee toistokertojen laskemiseen range-toimintoa. Range-komennolla määritetään, mitkä kaikki muuttujan arvoista otetaan mukaan. Koska kaikki listan alkiot halutaan tulostaa, komento on range(n).

```
n = len(nimitykset)
for i in range(n):
    print nimitykset[i]
```

Merkinnällä `nimitykset[i]` ohjelma hakee silmukkalaskurin määräämän alkion nimitykset-listasta.

#### Lisää listalle tehtäviä toimintoja

Alkion poistaminen listalta	<code>del lista1[poistettavan_alkion_indeksi]</code>
Listan suurimman alkion arvo	<code>max(lista1)</code>
Listan pienimmän alkion arvo	<code>min(lista1)</code>
Tietyn alkion esiintymiskerrat	<code>lista1.count(alkion arvo)</code>
Tietyn alkion ensimmäisen esiintymiskerran indeksi	<code>lista1.index(alkion arvo)</code>
Uuden alkion lisääminen tiettyyn kohtaan listaa	<code>lista1.insert(lisäyskohdan_indeksi, 'lisättävä tieto')</code>

### TEHTÄVÄ 3

Tavoitteena on harjoitella listan tekemistä ja listan alkioiden tulostamista.

Run

Tee ohjelma, jossa luodaan lista kuukausien nimistä ja harjoitellaan tulostamaan listan alkioita.

- Tee lista kuukausien nimistä. Anna listalle sisältöä kuvaava nimi. Tulosta koko lista.
- Tulosta listan ensimmäinen alkio.
- Tulosta listan viimeinen alkio.
- Tulosta listan kuukaudet kesäkuusta elokuuhun.
  - Rajaa tulostusalue indeksien avulla. Tulostusalue alkaa siitä indeksistä, joka viittaa ensimmäiseen tulostettavaan alkioon. Määritä alkamispisteeksi muuttuja  $a$ , jonka arvo on ensimmäisen tulostettavan kuukauden indeksi. Kesäkuun indeksi arvo on `kuukaudet.index(u'kesäkuu')`.
  - Tulostusalueen lopetuskohtaksi pitää määrittää yksi indeksi pidemmälle kuin mitä halutaan tulostaa. Käytetään apumuuttujaa  $b$  määrittämään tulostusalueen lopetuskohta. Kun halutaan tulostaa elokuuhun asti, täytyy tulostusalueen pituus olla yksi indeksi pidemmälle kuin on elokuun indeksi. Muuttujan  $b$  arvoksi tulee siten elokuun indeksi +1.
  - Tulosta alue komennolla `print lista[alkamispiste:lopetuspiste]`.

## TEHTÄVÄ 4

Tavoitteena on tehdä ohjelma, joka yhdistää kahden listan tiedot ja piirtää tiedoista havainnollistavan kuvion.

Run

Tee ohjelma, joka yhdistää alla olevan taulukon tiedot luokan oppilaiden arvosanoista ja niiden lukumääristä ja tulostaa tiedoista havainnollistavan kuvion.

<b>arvosana</b>	4	5	6	7	8	9	10
<b>lukumäärä</b>	0	2	3	4	5	3	2

- Määritä ensimmäinen lista, joka kuvaa arvosanoja. Listan alkiot ovat arvosanat 4–10.
- Määritä toinen lista, joka kuvaa saatujen arvosanojen lukumääriä.
- Määritä muuttuja  $a$  ja sille arvoksi jommankumman listan pituus *len*-komennolla.
- Määritä *for*-silmukka ja sille laskurimuuttujaksi  $i$ . *Range*-komennon avulla määritetään, että silmukkaa toistetaan muuttujan  $a$  verran.
- Määritä *for*-silmukan sisään apumuuttuja  $b$  ja tulostustoiminto seuraavasti:
  - Määritä apumuuttuja  $b$ , jonka avulla tulostetaan havainnollistava kuvio. Jotta lukumäärä-listan alkioita voidaan käyttää kertoimena palkin merkeille, pitää alkio muuttua kokonaisluvuksi komennolla `lkm[i]`. (Voit kerrata kuvion tulostamisen tehtävästä 2.)
  - Määritä tulostusrivi, jossa yhdistetään arvosanalistan alkio, välilyönti ja apumuuttujan  $b$  arvo.

## Esimerkki 2

Tee ohjelma, jolla voit muodostaa listan, lisätä listaan uusia alkioita ja lopuksi laskea ja tulostaa listan alkioiden summan.

- Tee lista numeroista 1, 5, 7 ja 9.
- Lisää listan loppuun numero 13.
- Lisää listan toiseksi alkioiksi numero 3.
- Laske ja tulosta listan alkioiden summa.

## Ratkaisu

Ohjelman toiminta on selitetty ohjelmakoodin kommentteissa.

```
1 #a) Tehdään lista nimeltä luvut.
2 luvut = [1, 5, 7, 9]
3
4 #b) Lisätään listan loppuun luku append-komennolla.
5 luvut.append(13)
6
7 #c) Jotta listan toiseksi jäseneksi saadaan lisättyä uusi alkio, täytyy tietää
   paikan indeksi. Listan toisen jäsenen indeksi on 1. Lisäys tehdään insert-
   komennolla.
8 luvut.insert(1, 3)
9
10 #d) Lasketaan listan lukujen summa for-silmukalla.
11 # Määritetään muuttuja summa ja sille alkuarvoksi nolla.
12 # Silmukkaa toistetaan listan pituuden verran. Listan pituus saadaan len-
   komennolla.
13 # Jokaisella kierroksella summaan lisätään se listan alkio, joka vastaa
   laskurimuuttujan arvoa.
14 # Silmukan loputtua tulostetaan summa.
15 summa = 0
16 for i in range(len(luvut)):
17     summa = summa + luvut[i]
18 print summa
```

## TEHTÄVÄ 5

Tavoitteena on tehdä ohjelma, jossa lasketaan listan alkioiden keskiarvo.

Run

Tee ohjelma, jolla lasket kuvan arvosanojen keskiarvon.

- Määritä lista kuvan todistusarvosanoista.
- Määritä muuttuja *summa* ja sen arvoksi nolla.
- Määritä muuttuja, jonka arvo on listan pituus.
- Tee *for*-silmukka, joka laskee listan alkioiden summan. Silmukkaa toistetaan listan pituuden verran.
- Laske todistusarvosanojen keskiarvo silmukan ulkopuolella.
- Tulosta keskiarvo.

Äidinkieli	8
Englanti	9
Ruotsi	8
Matematiikka	8
Fysiikka	7
Kemia	9
Biologia	9
Maantieto	6
Historia	8
Terveystieto	9
Liikunta	8
Kuvaamataito	8
Kotitalous	8
Käsityöt	8
Musiikki	7

## TEHTÄVÄ 6

Tavoitteena on tehdä ohjelma, jossa harjoitellaan listalle soveltuvia komentoja.

Run

Määritä lista kuvan todistusarvosanoista. Käytä kohdissa a-e listalle soveltuvia komentoja.

- Muuta matematiikan arvosanaksi 10. Käytä komentoa, jolla voit muuttaa alkion tietoja.
- Määritä listan suurin arvo ja tulosta se.
- Tulosta listan pienin arvo.
- Tulosta arvosanojen 8 lukumäärä.
- Tulosta arvosanan 7 ensimmäisen esiintymiskerran indeksi.

Äidinkieli	8
Englanti	9
Ruotsi	8
Matematiikka	8
Fysiikka	7
Kemia	9
Biologia	9
Maantieto	6
Historia	8
Terveystieto	9
Liikunta	8
Kuvaamataito	8
Kotitalous	8
Käsityöt	8
Musiikki	7

## VALMIIN KIRJASTON LISÄÄMINEN OHJELMAAN

Pythoniin voidaan tuoda valmiita kirjastoja. Esimerkiksi piirtämisessä voidaan käyttää *turtle*-kirjastoa.

Kirjasto voidaan lisätä ohjelmaan kahdella eri tavalla.

### Tapa 1

Kirjasto voidaan lisätä ohjelman alkuun seuraavalla komennolla:

```
from kirjaston nimi import*
```

Yllä oleva komento kirjoitetaan kerran ohjelman alkuun. Komentoa kannattaa käyttää silloin, kun ohjelmaan tuodaan vain yksi kirjasto.

### Tapa 2

Jos ohjelmassa tarvitaan useita kirjastoja, kirjastot kannattaa lisätä seuraavalla komennolla:

```
import kirjaston nimi
```

Tällöin jokaisen komennon alussa täytyy kertoa, mistä kirjastosta komento on peräisin.

## KIRJASTO RANDOM

Esimerkiksi lukujen arpomiseen tai listan satunnaisen alkion valitsemiseen tarvitaan *random*-kirjastoa. Kirjasto voidaan tuoda ohjelmaan joko komennolla *from random import\** tai *import random*.

### Random-kirjaston komentoja

Satunnaisen kokonaisluvun arpominen annetulta lukualueelta	<code>randint(alku, loppu)</code>
Satunnaisen kohteen arpominen listalta	<code>choice(listan nimi tai lista)</code>
Listan alkioden sekoittaminen satunnaisesti	<code>suffle(listan nimi)</code>
Satunnaisen alkion valitseminen määritellystä joukosta.	<code>randrange(alku, loppu, askel)</code>



### Esimerkki 3

Tee ohjelma, jolla arvotaan viisi lukua luvuista 1–10 ja arvotut luvut tulostetaan listana.

### Ratkaisu

Jotta ohjelmassa voidaan käyttää satunnaislukujen arpomista, täytyy ohjelmaan tuoda kirjasto *random*. Ohjelman alussa täytyy määritellä tyhjä lista, johon lisätään kierroksella arvottu luku.

Ohjelman toiminta on selitetty ohjelmakoodin kommentteissa.

```
# Lisätään kirjasto random.
from random import*

# Määritetään tyhjä lista.
arvotutLuvut = []

# For-silmukkaa toistetaan viisi kertaa.
for i in range(5):

# Muuttuja x saa arvoksi satunnaisesti valitun luvun väliltä 1-10.
    x = randint(1,10)

# Lisätään arvottu luku listan loppuun.
    arvotutLuvut.append(x)

# Tulostetaan lista.
print arvotutLuvut
```

Ohjelma tulostaa jokaisella suorituskerralla listan satunnaisesti valituista luvuista. Ohjelman tuloste voi näyttää esimerkiksi seuraavanlaiselta:

```
[7, 1, 3, 10, 6]
```

Listalle voi satunnaisesti valikoitua sama luku useammin kuin kerran, koska ohjelmassa ei ole ehtoa, joka estäisi saman numeron käyttämisen uudestaan.

### TEHTÄVÄ 7

Tavoitteena on harjoitella satunnaisluvun arpomista.

Run

Tee ohjelma, joka toimii noppana.

- Tuo ohjelmaan kirjasto *random*.
- Arvo yksi luvuista 1–6 ja tulosta se.

## TEHTÄVÄ 8

Tavoitteena on tehdä ohjelma mallintamaan kolikonheittoa.

Run

Tee ohjelma, jossa luodaan lista kruunasta ja klaavasta. Kolikkoa heitetään 10 kertaa ja ohjelma arpoo listalta heittotuloksen. Heittotulos tallennetaan toiseen listaan. Kun heitot on suoritettu, ohjelma tulostaa heittojen tulokset ja lasketaan eri vaihtoehtojen prosenttiosuudet.

- Tuo ohjelmaan kirjasto *random*.
- Määritä lista *kolikot*, jonka alkioita ovat kruuna ja klaava.
- Määritä tyhjä lista *heitot*, johon kootaan arvotut heittotulokset.
- Tee *for*-silmukka, jota toistetaan 10 kertaa.
- Määritä muuttuja *a*, joka saa arvokseen arvonnän tuloksen. Arvonta suoritetaan *choice*-komennolla, sillä kyseessä on merkkijonon muodostama lista.
- Lisää arvonnän tulos *a* listaan *heitot*.
- Tulosta *heitot*-lista.
- Silmukan loputtua *count*-komennolla lasketaan, kuinka monta kruunaa listalla on ja kuinka monta klaavaa listalla on.
- Laske kummankin vaihtoehdon prosenttiosuudet.
- Tulosta omat vastausrivit kruunalle ja klaavalle. Vastausriviin tulee sisällyttää vaihtoehdon nimi, vaihtoehdon lukumäärä ja prosenttiosuus.

## TEHTÄVÄ 9

Tavoitteena on suunnitella ja tehdä ohjelma, jossa heitetään kahta noppaa ja lasketaan heittojen silmälukujen summa. Heitetyt silmäluvut ja niiden summa tulostetaan.

Run

Laadi suunnitelma pyydetyistä ohjelmasta, ja kirjoita tarvittava ohjelmakoodi.

## TEHTÄVÄ 10

Tavoitteena on tehdä ohjelma, jolla pystyy arpomaan lottonumerot.

Run

Lottonumerot arvotaan 40 luvun joukosta. Ohjelma kannattaa kuitenkin ensin suunnitella ja testata pienemmällä joukolla, esim. luvuilla 1–10.

Luku saa esiintyä arvottujen lukujen rivissä vain kerran, joten ohjelma tulee rakentaa siten, että saman luvun esiintyminen useammin kuin kerran ei ole mahdollista.

Joukko, josta luvut arvotaan, kannattaa tehdä listaksi.

Kun listalta on arvottu jokin luku, ohjelma hakee kyseisen luvun indeksin listalta ja poistaa sen. Arvotut luvut kerätään listaan, ja lopuksi lista tulostetaan.

### Näin luot pitkän listan luvuista

For-silmukan avulla saadaan luotua pitkiä listoja.

```

numerot = []
for i in range(1, 11):
    numerot.append(i)

```

Yllä oleva koodi luo listan

```
numerot = [1, 2, 3, 4, 5, 6, 7, 8, 9,
```