

Koodauksen alkeet, Python (Opettajan opas)

Harjoitellaan ohjelmointia Python ohjelmointikielellä

Kuvaus:

Tutustutaan ja harjoitellaan ohjelmointia Python kielellä. Ohjelmoimiseen tutustutaan verkkoympäristössä, jossa oppilaat pääsevät kirjoittamaan koodia tekstipohjaisesti. Koodin lopputuotos nähdään nappulaa painamalla. Tehtävät on suunniteltu yksinkertaisiksi, jotta ohjelmointia voidaan harjoitella matematiikan tunnilla koko ryhmän kanssa.

Alta löytyy opettajan materiaali, johon on koostettu kaikki oleellisin, jotta selaimessa toimiva kokonaisuus olisi opettajan helppo ottaa haltuun. Lisäksi tehtäviin löytyy materiaalista oikeat/eräät ratkaisut. Halutessaan opettaja voi näyttää niillä ratkaisuehdotuksia ja oppia itsekin yksinkertaisen ohjelmoinnin toimintalogiikkaa.

Ohjelmointia kannattaa harjoitella oppilaiden kanssa kaksi oppituntia kerrallaan, jotta edelliset harjoitellut taidot ovat paremmin lähimuistissa.

Oppiaine: Matematiikka, Tietotekniikka **Vuosiluokka** 7 – 9 lk.

Kesto: 4 – 8h

Välineet: tietokone

Tunnisteet: Ohjelmointi, Python, Ohjelmoinnin alkeet

Oppilas

- Osaa tulkita tekstipohjaista ohjelmointikieltä ja tunnistaa erilaisia rakenteita sillä tehdystä ohjelmakoodista.
- Tietää, miten erilaiset ohjelmointirakenteet, kuten peräkkäiset, toistuvat ja ehdolliset toiminnot sekä muuttujat, toimivat. Oppilas osaa suunnitella ohjelman niitä käyttäen.
- Tuntee ohjelmoituja elementtejä ympäröivässä yhteiskunnassa, kuten algoritmi, automaatio, robotiikka ja tekoäly. Oppilas ymmärtää niiden toimintalogiikkaa ja sovelluksia elämän eri osa-alueilla.
- Tuntee eri oppiaineisiin liittyviä teknologisia sovelluksia ja osaa selittää niiden toimintaperiaatteita. Oppilas osaa hyödyntää ohjelmointiin liittyviä työskentelytapoja kaikissa oppiaineissa.
- Hyödyntää algoritmista ajattelua eri oppiaineisiin ja projekteihin liittyvässä ongelmanratkaisussa.
- Auttaa toisia ongelmatilanteissa.
- Osaa ohjelmoida ohjelmia eri ympäristöissä sekä tuntee perusasiat yhdestä tekstipohjaisesta ohjelmointikielestä.
- Osaa käsitellä erilaisia informaation muotoja, käyttää sen käsittelyssä erilaisia merkintätapoja ja toteuttaa informaation kanssa loogisia operaatioita.



Tämä materiaali sisältää ohjeet opettajalle Python ohjelmoinnin aloittamiseen.

Koodauksen harjoittelemista kannattaa harjoitella. Usein kysytään, minkälaisia taitoja ohjelmointi kehittää. Miksi ohjelmointia harjoitellaan, vaikkei siitä ammattia opiskelekaan?

Ohjelmointia harjoittelemalla riittävän yksikertaisessa ohjelmointikontekstissa opitaan monenlaisia taitoja. **Ohjelmointi kehittää lukutaitoa ja analyyttisiä päättelytaitoja, täsmällisyyttä ja sen avulla päästään harjoittelemaan ongelmanratkaisua.** Taitavista ohjelmoinnin osaajista on kroonisesti pulaa. Jos kiinnostusta riittäisi, siitä saisi itselleen hyväpalkkaisen työn.

Lähtökohdaksi ohjelmointimateriaalissa on valittu se, että ohjelmointia harjoitellaan suurilla oppilasryhmillä matematiikan oppituntien aikana. Siksi tehtävät on laadittu niin, että

- 1) Ensimmäiset tehtävät ovat hyvin helppoja ja niiden tekeminen vaatii ainoastaan esimerkikoodin käyttämistä tai valmiin koodin muokkaamista.
- 2) Tehtävät sisältävät yksinkertaisia ongelmia, jotka eivät vaadi minkäänlaisia esitietoja ohjelmoinnista. Silti loogisen päättelyn avulla oppilaat parhaimmillaan kykenevät ratkomaan ne ilman ohjausta.
- 3) Viimeiset tehtävät, lisä- ja soveltavat tehtävät vaativat useiden ohjelmointikäsitteiden hallintaa. Oppilaiden osaaminen ohjelmoinnissa on kiitettävää tai erinomaista, mikäli he kykenevät näitä ratkomaan joko yksin tai pienissä ryhmissä.

Ajallisesti materiaalin opiskeluun menee 7 - 9lk oppilailla n. 6 - 8 x 45min. Nämä voi opiskella esimerkiksi 2-3 / lukuvuosi, jolloin 9lk aikana päästään tekemään haastavampia ohjelmointitehtäviä.

Tässä materiaalissa on jokaiseen tehtävään kirjoitettu siihen liittyvät **tavoitteet ja malliratkaisu**. Monissa tehtävissä pyritään herättelemään oppilaan luovuutta, joten oppilaiden ei kannata kopioida malliratkaisuja tarkkaan. Mennään ymmärryksen kautta.

Mikäli opettaja itse haluaa, hän voi pitää kirjaa oppilaiden tehtävien suorituksista. Tosin ohjelmointiin käytettäviä oppitunteja on niin vähän, että suosittelen käyttämään yhteisen ajan tehokkaasti ohjelmoinnin harjoitteluun.

Linkki Python-ohjelmoinnin alkeet sivustolle, josta myös ohjelmointitehtävät löytyvät
<https://sites.google.com/view/ohjelmointikurssi/python-koodauksen-alkeet>, <http://dy.fi/a3c>

Sisällys

Ohje Tehtävänäkymän käyttöön Trinket palvelussa	3
Tehtävät ja niiden ratkaisut	4
1.0 Tulostaminen, Hello world!	4
TEHT 1.1.....	4
TEHT 1.2.....	4
2.0 Muuttuja.....	5
TEHT 2.1, a - c.....	5
TEHT 2.2, a - d	6
TEHT 2.3, a - d, e.....	7
3.0 Laskutoimituksia ja tietotyyppejä.....	8
Teht 3.1.....	8
TEHT 3.2.....	10
4.0 Syöte.....	11
TEHT 4.1.....	11
TEHT 4.2.....	12
5.0 Vertailu ja ehtolauseet.....	13
Teht 5.1.....	13
TEHT 5.2.....	14
Teht 5.3.....	15
Teht 5.6.....	17
6.0 Toistorakenne.....	18
Teht 6.1.....	18
Teht 6.2.....	19
Teht 6.5.....	21
Teht 6.6.....	23
7.0 Lista	24
Teht 7.1.....	24
Teht 7.2.....	25

Ohje Tehtävänäkymän käyttöön Trinket palvelussa

Tehtävät ovat tällä hetkellä Trinket nimisessä palvelussa. Sieltä ne toimivat suoraan. Oppilas avaa tehtävän (esim. 1.1) , josta avautuu seuraavanlainen näkymä.

1) Tehtävän-annot ovat kommentoituna.

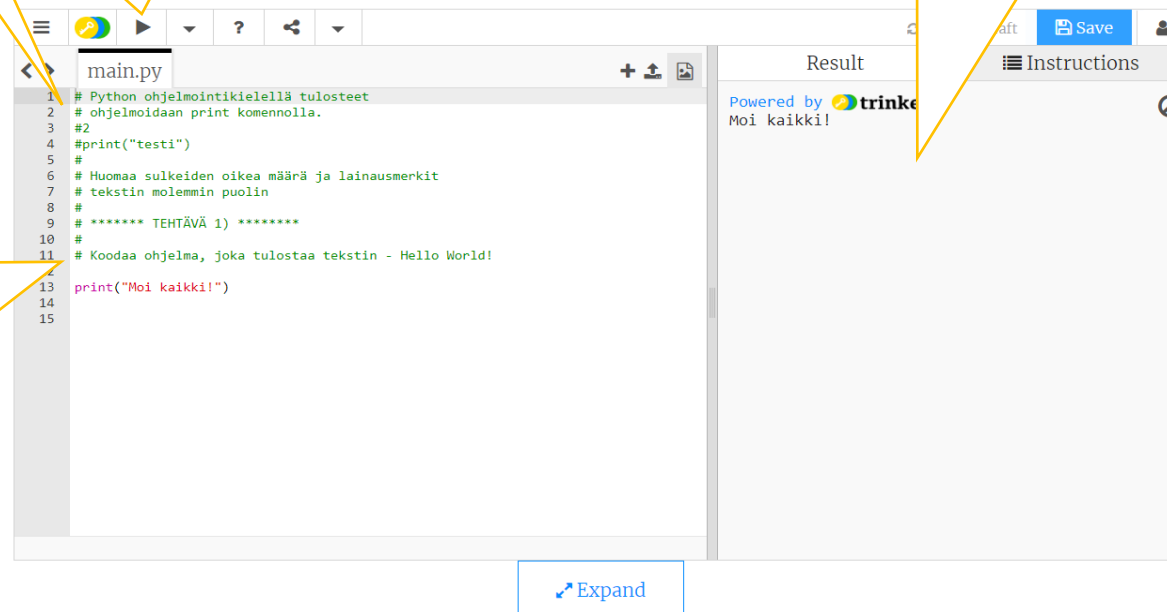
2) Suoritettava koodi kirjoitetaan tekstikenttään tyhjälle riville. Kyseessä on tavallinen tekstitiedosto. Esim. **main.py**

3) Koodin suorittaminen Python kääntäjällä tehdään **play** nappulalla.

Suorita ohjelma uudestaan aina muutoksien jälkeen ohjelma

4) Koodin tuotos tulee näkyville oikealle, kun Python ohjelma suoritetaan, Jos Python kääntäjä ei löydä virheitä koodista.

Koodin tuotoksena voi muodostua esimerkiksi tekstiä, kuva, animaatio tai peli



The screenshot displays the Trinket Python IDE interface. On the left, a code editor window titled 'main.py' contains the following Python code:

```
1 # Python ohjelmointikielellä tulosteet
2 # ohjelmoidaan print komennolla.
3 #2
4 #print("testi")
5 #
6 # Huomaa sulkeiden oikea määrä ja lainausmerkit
7 # tekstin molemmin puolin
8 #
9 # ***** TEHTÄVÄ 1) *****
10 #
11 # Koodaa ohjelma, joka tulostaa tekstin - Hello World!
12
13 print("Moi kaikki!")
14
15
```

On the right, a 'Result' panel shows the output: 'Powered by trinket' and 'Moi kaikki!'. Below the code editor, there is a 'Save' button and an 'Instructions' tab. At the bottom center, there is an 'Expand' button.

Tehtävät ja niiden ratkaisut

1.0 Tulostaminen, Hello world!

Tehtävä ja siihen liittyvät tavoitteet

Tehtävänanto ja esimerkkiratkaisu

Esimerkkikoodin tuotos

TEHT 1.1

- Ensimmäinen tutustuminen koodaukseen tekstiä kirjoittamalla
- Tehtävä on tarkoituksella lyhyt. "Osasin"
- Harjoitellaan merkkijonon tulostamista Pythonilla

```
# ***** TEHTÄVÄ 1) *****  
# Koodaa ohjelma, joka tulostaa  
# tekstin - Hello World!
```

```
print "Hello World!"
```

```
Hello World
```

TEHT 1.2

- Harjoitellaan print -toiminnon käyttöä
- Tuotetaan kuva usealla print-komenolla ja käyttäen mielikuvitusta.
- Tutustutaan tietotyyppiin merkkijono

```
# ***** TEHTÄVÄ 2) *****  
#  
# a) Koodaa ohjelma, joka tulostaa 4 erilaista  
tulostetta.  
# Voit käyttää myös erilaisia merkkejä.
```

```
print "!!!"  
print "***"  
print "---"  
print "#&%"
```

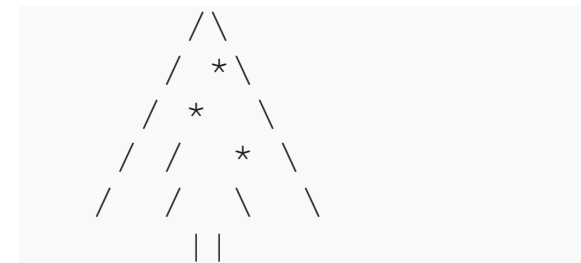
```
!!!  
***  
---  
#&%
```

Merkkijono merkitään lainausmerkein
Esim.

"Merkkijono"

```
# b) Tulosta merkkejä siten, että kokonaisuus näyttää  
puulta.
```

```
print "  
print "  
print "  
print "  
print "  
print "
```



```
# c) Tulosta merkkejä siten, että lopputulos muistuttaa  
# Kalan ruotoa
```

```
print "<(((((-< "
```

```
<(((((-<
```

2.0 Muuttuja

Tehtävä ja siihen liittyvät tavoitteet

Tehtävänanto ja esimerkkiratkaisu

Esimerkkikoodin tuotos

TEHT 2.1, a - c

- Opetellaan, mitä ovat muuttujat ja mihin niitä käytetään.

- Harjoitellaan muuttujien nimeämistä ja niiden sisältöjen tulostamista

```
# TEHTÄVÄ  
#
```

```
# a) Aseta muuttujan nimi arvoksi oma nimesi  
# Tulosta nimi muuttujan sisältö.
```

```
nimi = "Sami"  
print nimi
```

```
Sami
```

```
# b) Luo muuttuja ika ja aseta sinne haluamasi ikä.  
# Tulosta ika muuttujan sisältö
```

```
ika = 35  
print ika
```

```
35
```

```
# c) Luo muuttujat luku1, luku2 ja x.  
# aseta niiden arvoksi haluamasi luvut.  
# Tulosta muuttujien arvot.
```

```
luku1 = 123456  
luku2 = 876543  
x = 1.2
```

```
print luku1  
print luku2  
print x
```

```
123456  
876543  
1.2
```

TEHT 2.2, a - d

- Harjoitellaan muuttujien käyttämistä ja tutustutaan muuttujien nimeämisen hyviin käytänteisiin ja muutamaa rajoitteeseen

```
# TEHTÄVÄ
#
# a) Korjaa koodin muuttujat
#   hyvän käytänteen mukaisiksi
```

```
ika = 35
automerkki = "Volvo"
summa = 0
```

(ei tulostetta)

- Varmistetaan, että oppilas osaa tulostaa tekstin tai merkkijonon konsoliin

```
# b) Luo neljä eri muuttujaa ja talleta niihin
# erilaisia arvoja, tekstiä ja lukuja.
```

```
x1 = 0
y1 = 1
x2 = 2
y2 = 2
```

(ei tulostetta)

- Pohditaan yhdessä, mikä merkitys \n -merkillä on ohjelmakoodissa.

\n --> Kyseessä on rinvaihtomerkki, joita käytetään tekstin muotoilussa esim. Word-dokumenteissa. Rinvaihto tallennetaan oikeasti merkinä!

```
# c) Tulosta luku muuttujan sisältö konsoliin.
```

```
#   Mikä luku muodostuu?
luku = "***** \n* \n *** \n   * \n*****\n"
print luku
```

```
*****
*
```

```
***
```

```
*
```

```
****
```

- Ratkotaan ongelmaa: luvun määrittäminen merkkijonolla rinvaihtomerkkiä apuna käyttäen

```
# d) Määritä muuttuja luku2.
```

```
#   aseta sen sisällöksi merkkejä siten,
#   että se tulostettaessa muodostaa jonkun toisen
luvun
```

```
luku2 = " 888 \n8   8 \n 888 \n8   8\n 888"
print luku2
```

```
888
```

```
8   8
```

```
888
```

```
8   8
```

```
888
```

TEHT 2.3, a - d, e

- Tehtävä kertaa osioissa 1 - 2 opitut taidot. Sen voisi tehdä esim. kun ohjelmoinnin pariin palataan seuraavalla vuosiluokalla.

```
# Tehtävä
#
# a) Aseta etunimi ja sukunimi -muuttujiin omat
tietosi.

etunimi = "Miina"
sukunimi = "Mustinen"

# b) Ohjelmoi etunimi ja sukunimi muuttujien sisältö
tulostumaan
# tähtirivien väliin.
print "*****"
print "Tervetuloa koodaamaan "
print etunimi, sukunimi
print "*****"

# c) Ohjelmoi tietokone tulostamaan pieni laatikko.
# Voit käyttää esim. _ (Shift + viiva) ja |
(AltGr + <) merkkejä

print " _ _ "
print "| | "
print "|_ | "

# d) Ohjelmoi tietokone tulostamaan kuusisakarainen
tähti

print " |\| "
print "< > "
print " |/\| "

# e) Lisätehtävä: Ohjelmoi kone tulostamaan pieni talo
```

(ei tulostetta)

```
*****
Tervetuloa koodaamaan
Miina Mustinen
*****
```

```
 _ _
| |
|_ |
```

```
  |\|
< >
 |/\|
```

```
      - O
     / \ /
    /   \
   |___|
  ___|___|___
```

3.0 Laskutoimituksia ja tietotyyppejä

Tehtävä ja siihen liittyvät avoitteet	Tehtävänanto ja esimerkkiratkaisu	Esimerkkikoodin tuotos
Teht 3.1	<pre>x = 987 y = 123</pre>	
- Harjoitellaan kirjoittamaan laskutoimituksia Python kääntäjällä ja tulostamaan vastaus	<pre># ***** TEHTÄVÄ a) ***** # Koodaa ohjelma tulostamaan edellä määritettyjen muuttujien # summa, erotus, tulo ja osamäärä print(x + y) print(x - y) print(x * y) print(x / y)</pre>	<pre>1110 864 121401 8.0243902439</pre>
- Harjoitellaan hahmottamaan luvut ja merkkijonot erilaisina tietotyyppeinä	<pre># ***** TEHTÄVÄ b) ***** # Luo muuttujat x1, x2, x3, x4, ja x5 ja sijoita niihin lukuarvot. # Koodaa ohjelma tulostamaan niiden tulo. x1 = 1 x2 = 2 x3 = 1839 x4 = 12345 x5 = 1234 print (x1 * x2 * x3 * x4 * x5)</pre>	<pre>56029658940</pre>
- Varmistetaan muuttujakäsitteen osaaminen ja lasketaan niiden avulla.	<pre>x1 = 1 x2 = 2 x3 = 1839 x4 = 12345 x5 = 1234 print (x1 * x2 * x3 * x4 * x5)</pre>	<pre>0.24 0.666666666667</pre>
- Harjoitellaan kasvattamaan ja vähentämään muuttujan alkuperäistä arvoa	<pre># ***** TEHTÄVÄ c) ***** # # Kasvata muuttujan x arvoa yhdellä ja # vähennä muuttujan y arvosta muuttujan x arvo # Tulosta luvut x = x + 1 y = y - x print "luku x", x print "luku y", y</pre>	<pre>1000000000000000000000</pre>
- Pohditaan, mitkä laskutoimituksista on raskaita laskettavaksi ja tuottavat suuria lukuja.	<pre>x = x + 1 y = y - x print "luku x", x print "luku y", y</pre>	<pre>988 -864</pre>
<i>Esim. luku x potenssiin x tuottaa suuren luvun</i>	<pre># ***** TEHTÄVÄ d) ***** # Desimaalilukuihin (Esim 9.0) pitää lisätä piste ja ensimmäinen desimaali # # Luo muuttujat y1, y2. Aseta niihin desimaaliluvut. # Ohjelmoi tulostumaan noiden lukujen sekä tulo että osamäärä. y1 = 0.4</pre>	
<i>sata ensimmäistä</i> <i>for i in range(101):</i> <i>print i ** i</i>		

```
y2 = 0.6
print(y1 * y2)
print(y1 / y2)

# ***** TEHTÄVÄ e) *****
# Koodaa kone laskemaan niin haastavaa laskua,
# että selaimen sivu menee jumiin.

print(100000 ** 100000)
```

TEHT 3.2

```
# Kun tietokoneelle annetaan tekstiä, yhteenlasku
suoritetaan liittämällä
# merkit peräjälkeen
s1 = "11111111"
s2 = "99999999"

- Tutustutaan merkkijonojen
yhdistämiseen

- Harjoitellaan
hahmottamaan luvut ja
merkkijonot erilaisina
tietotyyppeinä

- Harjoitellaan tekstin
muuntamista luvuksi

# *** Tehtävä ***
#
# a) Luo muuttuja nimeltä vastaus
# sijoita siihen yhdistelmä yllänäkyvistä
muuttujista.
# käytä yhdistämiseen + merkkiä. Tulosta vastaus.
vastaus = s1 + s2
print(vastaus)

s3 = "Code "
s4 = "Rulez"
s5 = "!"
# b) Luo muuttuja teksti.
# Sijoita siihen muuttujien s3-s5 sisällöt yhdeksi
tekstiksi.
# Tulosta yhdistetty teksti
teksti = s3 + s4 + s5
print(teksti)

# c) Luo muuttujat x1 ja x2.
# Muunna tekstit luvuiksi komennolla int() ja aseta
ne muuttujiin.
# Esim.
# x1 = int( s1 )
#
# Kokeile uudestaan yhteenlaskua ja tulosta
lopputulos.
x1 = int(s1)
x2 = int(s2)
print(x1 + x2)
```

1111111199999999

Code Rulez!

111111110

4.0 Syöte

Tehtävä ja siihen liittyvät tavoitteet	Tehtävänanto ja esimerkkiratkaisu	Esimerkkikoodin tuotos
TEHT 4.1 - Tutustutaan syöte-toimintoon, jolla saadaan käyttäjän kirjoittama teksti tai luku ohjelman käyttöön - Varmistetaan, että oppilas osaa tallettaa tiedon muuttujaan, jotta tietoa voidaan käyttää vielä uudelleen - Harjoitellaan muuntamaan syötteen teksti luvuksi	<pre># Käyttäjän syöte saadaan poimittua # talteen input toiminnolla. # # Esim. # # print(Mikä on suosikkivuodenaikasi? Kirjoita:.) # vuodenaika = input() # print("syötteesi on: " + vuodenaika) # ***** TEHTÄVÄ 1) ***** # # Koodaa ohjelma, joka pyytää käyttäjää kirjoittamaan # käyttäjätunnuksen. Käyttäjän kirjoittama teksti # tallennetaan # muistiin muuttujaan nimi. Lopuksi ohjelma tulostaa # käyttäjän kirjoittaman nimen print "Kirjoita käyttäjätunnuksesi:" nimi = input() print nimi # ***** TEHTÄVÄ 2) ***** # # Koodaa ohjelma, joka kysyy kaksi lukua. # Tieto talletetaan muuttujiin luku1 ja luku2. # Ohjelma tulostaa lopuksi lukujen tulo. # Muista muuntaa syötteet luvuiksi ennen laskemista. print "kirjoita kaksi lukua, erottele ne painamalla enteriä välissä" luku1 = int(input()) luku2 = int(input()) print "tulo: " print(luku1 * luku2)</pre>	<pre>Kirjoita käyttäjätunnuksesi: Saz Saz kirjoita kaksi lukua, erottele ne painamalla enteriä välissä 12345 765432 tulo: 9449258040</pre>

```
# ***** TEHTÄVÄ *****
#
TEHT 4.2 # Koodaa ohjelma, joka kysyy käyttäjän nimen ja iän
          # (vuoden
- Harjoitellaan soveltamaan syöte toimintoa # lopussa).
          # Ohjelma laskee käyttäjän syntymävuoden
          # ja tulostaa käyttäjälle tuloksen muodossa
- Kerrataan merkkijonon muuntaminen luvuksi # Nimi: Pete, Syntymävuosi: 1984
          # huom: luku muunnetaan tekstiksi str() komennolla
          # tulostusta varten (Python 2.7)
- Harjoitellaan laskulausekkeen tuottamista print "Kirjoita nimesi:"
          nimi = input()

          print "Kirjoita ikäsi"
          ika = int ( input() )

          svuosi = 2023 - ika
          print "Nimi:", nimi, "Syntymävuosi:", svuosi
```

```
Kirjoita nimesi:
Sami
Kirjoita ikäsi
23
Nimi: Sami ,
Syntymävuosi: 2000
```

5.0 Vertailu ja ehtolauseet

Tehtävä ja siihen liittyvät tavoitteet	Tehtävänanto ja esimerkkiratkaisu	Esimerkkikoodin tuotos
Teht 5.1	<pre># *** TEHTÄVÄ 5.1) *** # #</pre>	
- Tutustutaan vertailuarvoihin True ja False	<pre># **** Teht 1) **** # # Muokkaa lukuja tai vertailumerkkejä siten, # että tulosteeksi saadaan joka rivillä # True eli Tosi. # print " --- teht 1 ---" print(10 < 15) print(5 < 10000) print(3 == 3.0) print(3.2 != 3.8) print(1 > 0.9) print(1000 > 1)</pre>	<pre>--- teht 1 --- True True True True True True</pre>
- Muodostetaan vertailuja	<pre># ***** Teht 2) ***** # # Kirjoita vertailuja kuten edellisessä tehtävässä # Käytä vertailuissa ainakin seuraavia # a) erisuuri # b) suurempi tai yhtäsuuri # c) yhtäsuuri print " --- teht 2 ---" print(1 != 0) print(1000 >= 10) print(9.8 == 9.80)</pre>	<pre>--- teht 2 --- True True True</pre>

TEHT 5.2

- Harjoitellaan ehtolauseen kirjoittamista. Tutustutaan satunnaisuuteen valmiin koodin avulla.
- Harjoitellaan vertailumerkkien käyttöä käytännössä eli ehtolausekkeiden kirjoittamista

```
import random
x = random.randint(1, 6) #Arvotaan luku väliltä 1 - 6
                        #Luku sijoitetaan muistiin muuttujaan x

if x == 1:
    print "Kokeile uudestaan!"

# *** tehtävät ***
#
# a) Tulosta muuttujan x arvo
print "x:", x

# b) Koodaa niin että, jos arvotuksi luvuksi tulee 6,
tulostetaan
#   teksti "täysosuma"
if x == 6:
    print "täysosuma"

# c) Koodaa niin että, Jos satunnainen luku on
#   pienempi kuin 4, ohjelma kommentin
#   "Arvottu luku on pienempi kuin 4".
if x < 4:
    print "Arvottu luku on pienempi kuin 4"

# d) Luo muuttuja y. Siihen sijoitetaan myös arvottu luku
#   väliltä 1 - 6. Jos molemmat luvut x ja y saavat arvot 6
#   tulostetaan teksti "Superia!"
y = random.randint(1, 6)
print "y:", y

if x + y == 12:
    print "Superia!"
```

(Suorittamalla ohjelma monesti joskus arvottu luku on myös 6)

6
Arvottu luku on suurempi kuin 4

Täysosuma!

Teht 5.3

- Tehtävä sopii kertaustehtäväksi pidemmän tauon jälkeen. Sisällöltään se on hyvin samanlainen kuin Teht 5.2
- Kerrataan yksinkertaisten ehtolauseiden käyttöä ja syntaksia.
- Viritellään humoristisella otteella nopanheitto-kommentaattori
- Lisätehtävässä opiskelija ohjataan tutustumaan itsenäisesti listojen käyttöön ja mahdollisuuteen ohjelmoida tietokone käyttämään sanapankkia yksittäisen kommentin sijaan.

```
import random
x = random.randint(1, 6)
print "Arvottu luku:", x

# Yläpuolella on koodi, joka arpoo kokonaisluvun
# väliltä 1-6. Luku asetetaan muuttujan x arvoksi.
# Kun ohjelma suoritetaan arvottu luku tulostetaan.
#
#
# *** Teht 1) ***
#
# Ohjelmoi ohjelma, joka heittää kommentteja nopan
silmä-
# luvun perusteella. Jos nopan silmäluvuksi tulee
tietty luku,
# niin ohjelma tulostaa tietyn tekstin.
# Esim.
# 1 -> "Äh"
# .
# .
# 6 -> "Jee!"
if x == 1:
    print("Äh!")

if x == 2:
    print "Aika heikko suoritus"

if x == 3:
    print "Ihan ok, yritä vielä"

if x == 4:
    print "Aika perus."

if x == 5:
    print "Yhtä vaille paras!"

if x == 6:
    print "Jee!"
```

(tuloste voi näyttää
esim. seuraavanlaiselta)

Arvottu luku: 1

Äh!

```
# *** Lisätehtävä - Opiskele: Listat / osio 8 ***
# Muunna ohjelma käyttämään sanapankkia. Mikäli nopasta
saadaan
# luku 6, poimitaan kommentti listasta, josta löytyy
ainakin 4 eri
# vaihtoehtoa

lista6 = ["JEE!", "JIPPII!", "MAHTAVAA!", "OLET PARAS!"]
if x == 6:
    pituus = len(lista6)
    arvottuluku = random.randint(0, pituus-1)
    print lista6[arvottuluku]
```

Arvottu luku: 6

Jee!

OLET PARAS!

Teht 5.6

- Harjoitellaan tietokoneen ja käyttäjän välisen viestinnän ohjelmointia.

```
##### Tietovisa, speksit #####  
#  
# Koodaa tietovisa, jossa on ainakin kaksi kysymystä. #  
# Kysymyksiin esitellään vastausvaihtoehdot. #  
# Kun käyttäjä vastaa oikein, ohjelma tulostaa tekstin "oikein". #  
# Väärästä vastauksesta tulostetaan teksti "väärin" #  
# Ohjelma laskee automaattisesti käyttäjän oikeiden #  
# vastausten määrän ja tulostaa sen ruudulle tietovisan päätyttyä. #  
# #  
#####  
laskuri = 0
```

- Harjoitellaan käyttäjän kirjoittaman syötteen sijoittamista muuttujaan ja harjoitellaan ehtolauseiden käyttöä oikean vastauksen tarkistamisessa.

```
print "Mikä seuraavista on pääkaupunki"  
print "a) Mätä"  
print "b) Lima"  
print "c) Kura"  
  
vastaus = input()  
if vastaus == "b":  
    print "OIKEIN!"  
    laskuri = laskuri + 1  
else:  
    print "Väärin"
```

- Kerrataan muuttujien käyttäminen laskurina. (Osio 3)

```
print "Mikä seuraavista on eläin"  
print "a) Pomppunen"  
print "b) Soijainen"  
print "c) Karhukainen"  
  
vastaus = input()  
if vastaus == "c":  
    print "OIKEIN!"  
    laskuri = laskuri + 1  
else:  
    print "Väärin"  
  
print "Oikeita vastauksia", laskuri, "kpl"
```

```
Mikä seuraavista  
on pääkaupunki  
a) Mätä  
b) Lima  
c) Kura  
b  
OIKEIN!  
Mikä seuraavista  
on eläin  
a) Pomppunen  
b) Soijainen  
c) Karhukainen  
c  
OIKEIN!  
Oikeita vastauksia  
2 kpl
```

6.0 Toistorakenne

Tehtävä ja siihen liittyvät tavoitteet	Tehtävänanto ja esimerkkiratkaisu	Esimerkkikoodin tuotos
Teht 6.1	<pre># Allaoleva malliohjelma tulostaa # luvut 0 - 3 # for i in range(0, 4): print(i)</pre>	0 1 2 3
- Tutustutaan toistorakenteeseen valmiin koodin avulla	<pre>print "-----" # *** tehtävä 1) *** # # Kirjoita ohjelma, # joka tulostaa luvut 2-10. for i in range(2, 11): print(i)</pre>	----- 2 3 4 5 6
- Harjoitellaan toistorakenteen käyttämistä haluttujen lukujen tulostamisessa	<pre>print "-----" # *** tehtävä 2) *** # # Koodaa ohjelma, joka tulostaa parilliset luvut 2-20 for i in range(2, 20, 2): print(i)</pre>	7 8 9 10 ----- 2 4
- Harjoitellaan toistorakenteessa olevan muuttujan arvon käyttämistä osana koodia	<pre>print "-----" # *** tehtävä 3) *** # # Koodaa ohjelma, joka tulostaa 100 ensimmäistä lukua, # luvun yhdeksän kertotaulusta. # # Vinkki: ensimmäinen 1 * 9 # toinen ... 2 * 9 # kolmas ... 3 * 9 # n:s n * 9 for i in range(1, 101): print(i * 9)</pre>	6 8 10 12 14 16 18 ----- 9
	<pre>print "-----" # *** Lisätehtävä *** # # Koodaa ohjelma, joka tulostaa 100 ensimmäistä lukua, # jotka ovat jaollisia luvulla 14921. for i in range(1, 101): print(i * 14921)</pre>	18 27 ... ----- 14921 29842 ...
	<pre>print "-----"</pre>	

Teht 6.2

- Harjoitellaan tunnistamaan toistuva kuvio
- Harjoitellaan määrittämään kuviosta toistuvia osioita
- Tuotetaan toistuvia kuvioita toistorakenteen avulla

```
# *** Teht 1 ****
#
# Ohjelmoi, ohjelma joka tulostaa
# alla näkyvän kuvion
#
# * * * * *
#  * * * *
# * * * *
#  * * * *
# * * * *
#  * * * *
#
# Käytä ratkaisussa for -toistorakennetta.
# Saat käyttää ohjelmassa vain 3 riviä koodia
for i in range(3):
    print "* * * * *"
    print " * * * * "
```

```
* * * * *
 * * * *
* * * *
 * * * *
* * * *
 * * * *
```

```
# *** Teht 2) ***
#
# Ohjelmoi, ohjelma joka tulostaa
# alla näkyvän kuvion
#
# / \ / \ / \
# \ / \ / \ /
# / \ / \ / \
# \ / \ / \ /
# / \ / \ / \
# \ / \ / \ /
#
# Käytä ratkaisussa for -toistorakennetta.
# Saat käyttää ohjelmassa vain 3 riviä koodia
for i in range(3):
    print "/ \ / \ / \ "
    print "\ / \ / \ / "
```

```
/ \ / \ / \
\ / \ / \ /
/ \ / \ / \
\ / \ / \ /
/ \ / \ / \
\ / \ / \ /
```

```

# *** Tehtävä 3) ***
#
# Muodosta seuraava toistuva kuvio
# / / / /
# / / / /
# / / / /
# / / / /
# / / / /
# / / / /
# / / / /

```

```

for i in range(3):
    print " / / / / "
    print " / / / / "
    print " / / / / "

```

```

# *** Lisätehtävä ****
#
# Ohjelmoi, ohjelma joka toistaa
# alla näkyvää kuviota. Voit käyttää apuna
# muuttujia toistorakenteessa.
#
# -----
# 0 x 0 x 0 x 0
# x 0 x 0 x 0 x
# 0 x 0 x 0 x 0
#
# -----
# 0 x 0 x 0 x 0
# x 0 x 0 x 0 x
# 0 x 0 x 0 x 0
#
# -----
t1 = "-----"
t2 = "0 x 0 x 0 x 0"
t3 = "x 0 x 0 x 0 x"

```

```

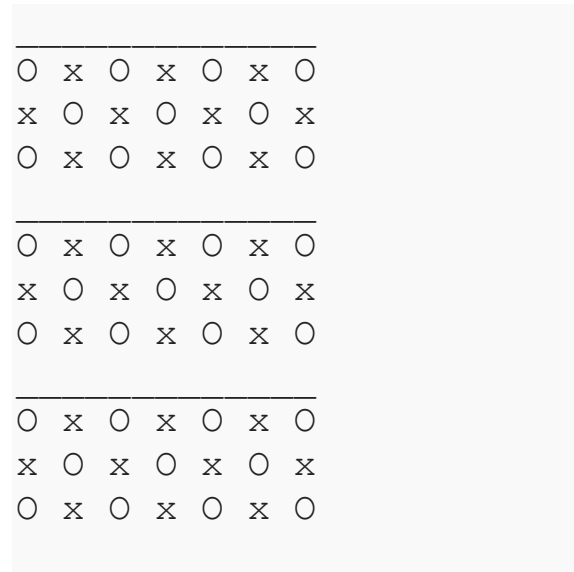
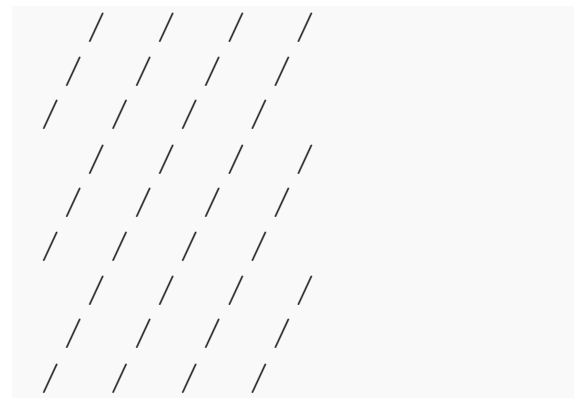
for i in range(3):
    print t1
    print t2
    print t3
    print t2

```

```

print t1

```



Teht 6.5	# while luoppu voidaan ohjelmoida esim. seuraavasti.	1
		2
	i = 1	3
- Tutustutaan while toistorakenteen ja sen käyttöön	while i < 5: print(i) i = i + 1	4
- Harjoitellaan koodin lukemista:	print "-----"	-----
o mitä muuttujia koodissa on?	# *** tehtävä 1) *** #	2
o mistä löytyy luupin päättävä ehto	# Kokeile ensin, mitä nykyinen koodi tekee.	3
o millä rivillä muuttujan i arvoa kasvatetaan yhdellä?	# Kopioi koodi ja muokkaa sitä siten, # että ohjelma tulostaa luvut 2-10. i = 2	4
	while i < 11:	5
	print(i)	6
	i = i + 1	7
- Tuotetaan lukuja käyttämällä while toistorakennetta	print "-----"	8
		9
		10

- Lisätehtävässä tutustutaan ohjelmoinnissa tyypilliseen modulo- eli jakojäännös - operaation	# *** tehtävä 2) *** #	7
	# Kopioi koodi ja muokkaa sitä siten, että se # tulostaa parittomat luvut väliltä 7 - 15	9
	i = 7	11
	while i < 16:	13
	print(i)	15
	i = i + 2	-----
	print "-----"	

```
# *** lisätehtävä ***
#
# Kopioi koodi ja muokkaa sitä siten, että se
# tulostaa parittomat luvut väliltä 1 - 51, mutta
# Jättää tulostamatta luvut, jotka ovat jaollisia
# luvulla 5.
# Jaollisuus kannattaa määrittää jakojäännös
# (modulo) operaation % avulla
# Esim.
# 9 % 3 = 0
# 10 % 3 = 1
# 11 % 3 = 2
# 12 % 3 = 0 jne.

i = 1

while i <= 101:
    if i%5 > 0:
        print(i)

    i = i + 2

print "-----"
```

```
1
3
7
9
11
13
17
19
21
23
27
29
31
33
37 ...
```

```
#Jakoäännös > 0
```

Teht 6.6

- Tulkitaan (yhdessä), mitä tehtävän valmiina oleva koodi tekee:
 - o Toistuu niin kauan kun arvottu luku **ei ole 0**
 - o Jokaisella toistolla arvotaan uusi luku väliltä 0–9
- Kerrataan laskurin ohjelmoiminen: muuttuja ja sen arvon kasvattaminen
- Tutkitaan satunnaisuutta laskurin avulla
- Lasketaan arvottujen lukujen summa

```
import random
# Kertaus: edelliseen lukuun saadaan lisättyä
# luku seuraavasti

laskuri = 0
laskuri = laskuri + 1
print "Esimerkkilaskuri:", laskuri, "\n"

# *** Tehtävä 1) ***
#
# Koodaa tietokone laskemaan,
# kuinka monta toistoa
# while-toistorakenteessa tehdään.
# Tulosta toistojen määrä.
# Täydennä olla olevaa koodia.

laskuri = 0
summa = 0
arpa = -1

while arpa != 0:
    arpa = random.randint(0, 10)
    laskuri = laskuri + 1
    summa = summa + arpa

print "Teht 1) Toistojen lukumäärä:", laskuri
print "Teht 2) Summa:", summa

# *** Tehtävä 2) ***
#
# Ohjelmoi tietokone laskemaan arvottujen lukujen
summa.
# Voit tehdä tarvittavat lisäykset yllä olevaan
valmiiseen koodiin
```

(ratkaisussa lukumäärä ja summa riippuvat satunnaisluvusta)

Esimerkkilaskuri: 1

Teht 1) Toistojen lukumäärä: 19

Teht 2) Summa: 101

7.0 Lista

Tehtävä ja siihen liittyvät tavoitteet

Tehtävänanto ja esimerkkiratkaisu

Esimerkkikoodin tuotos

Teht 7.1

```
# Listan luominen esim.
#
# lista1 = []
# lista2 = ["Kissa", "Karhukainen", "Gorilla"]
# lista2.append("Lumileopardi")

# *** TEHTÄVÄ 1) ***
#
# Luo lista nimeltä lista3.

lista3 = []

# *** TEHTÄVÄ 2) ***
#
# Luo lista nimeltä kauppalista siten,
# että siinä on valmiiksi 2 ostettavaa asiaa listalla.

kauppalista = ["maitoa", "juustoa"]

#TEHTÄVÄ 3) ***
#
# Lisää kauppalistaan vielä 2 ostettavaa asiaa.
# Tulosta lopuksi lista.
kauppalista.append("Jäätelöä")
kauppalista.append("Salsaa")
print kauppalista
```

```
['maitoa', 'juustoa',
'Jäätelöä', 'Salsaa']
```

Teht 7.2

```
import random

lista = [1, 5, 10, 12, 18]

# *** Tehtävä 1 ***
#
# Tulosta yllänäkyvästä listasta listan
# ensimmäinen ja toinen luku
print "1. -> ", lista[0]
print "2. -> ", lista[1]

# *** Tehtävä 2 ***
#
# Luo lista, jonka nimi on luvut. Lisää
# listaan joku valitsemasi 5 lukua.
# Tulosta lista

luvut = [1, 2, 4, 8, 16]
print luvut

# *** Tehtävä 3 ***
#
# Luo lista jokeririvi.
# Lisää jokeririvi -listalle kymmenen
# satunnaista lukua väliltä 0-9.
# Tulosta arvottu jokeririvi.
#
# satunnainen luku saadaan random moduulin
# randint() -toiminnolla esim. seuraavasti
# x = random.randint(0, 3)
jokeririvi = []
for i in range(10):
    x = random.randint(0, 9)
    jokeririvi.append(x)

print jokeririvi
```

```
1. -> 1
2. -> 5
```

```
[1, 2, 4, 8, 16]
```

```
[5, 1, 4, 6, 1, 2, 3, 7,
8, 2]
```
