

Ohjelmointi lukion matematiikassa, mitä opettaa opintojaksossa MAA11?

Perusteet
osa 1.



Hello world!

Teppo Harju, fysiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio

Ville Tilvis, matematiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio

Mika Setälä, matematiikan opettaja, Lempäälän lukio

Antti Laaksonen, yliopistonlehtori, Helsingin yliopisto, tietojenkäsittelytieteen osasto



Hei maailma!

- Kerro lyhyesti: kuka olet ja mistä tulet?
- Kerro myös, mitä ajatuksia ja tunteita ohjelmoinnin opetus herättää.



Koulutuksen tarkoitus ja päivän tavoite

- Johdatus lukiomatematiikan ohjelmoinnin opetukseen Python-kielellä
- Ideoita moduulin maa11 opettamiseen

Päivän päätteeksi:

- Osaan valita itselleni sopivan alustan ohjelmoinnin opetukseen
- Osaan Pythonin perusteet
- Osaan ratkaista yo-tehtävän ohjelmoimalla
- Tiedän mistä saan lisätietoa ja mitä kokeilen seuraavaksi



Päivän aikataulu

9.00 - Hei kaikki, esittelyt, tämä dia

9.15 - Opetussetti 1.

- LOPS 2021 ja MAA11
- Erilaiset ympäristöt
- Pythonin perusteet
- Harjoituksia pienryhmissä

11.00 - Lounas

12.00 - Opetussetti 2.

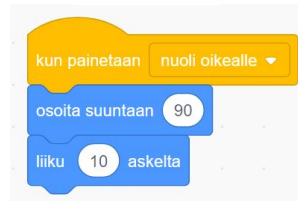
- Lisää harjoituksia
- Esimerkki Maa11 -kurssin rakenteesta
- Yo-tehtävä

14.20 - Palautekeskustelu ja eväitä jatsoon

15.00 - Lopetus



	Laaja-alainen osaaminen: Tieto- ja viestintä teknologinen osaaminen (L5)	Matematiikan opetuksen tavoitteet ja keskeiset sisältöalueet	Matematiikan arviointikriteerit: Arvosana 8
1-2	Oppilaat saavat ja jakavat keskenään kokemuksia digitaalisen median parissa työskentelystä sekä ikäkaudelle sopivasta ohjelmoinnista	Tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla vaiheittaisia toimintaohjeita, joita myös testataan.	
3-6	Ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista.	T14 innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä. Suunnitellaan ja toteutetaan ohjelmia graafisessa ohjelmointiympäristössä.	T14 Oppilas osaa ohjelmoida toimivan ohjelman graafisessa ohjelmointiympäristössä.
7-9	Ohjelmointia harjoitellaan osana eri oppiaineiden opintoja	T20 ohjata oppilasta kehittämään algoritmista ajatteluaan sekä taitojaan soveltaa matematiikkaa ja ohjelmointia ongelmien ratkaisemiseen. Syvennetään algoritmista ajattelua. Ohjelmoidaan ja samalla harjoitellaan hyviä ohjelmointikäytäntöjä.	T20 Oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja osaa ohjelmoida yksinkertaisia ohjelmia.



```
a = 3
b = 5
c = 2*a+b
print(c)
```

Toteutus yläkoulussa vaihtelee paljon

Kuutio (Sanoma Pro)

- noin viiden tunnin mittainen ohjelmointiosio jokaisella vuosiluokalla
- ohjelmointikielenä Python

Pii (Otava)

- ohjelmoinnillista ajattelua erilaisin tehtävin
- ei tiettyä ohjelmointikieltä

Säde (Edita)

- ohjelmallisen ajattelun ja ohjelmoinnin opiskeluun 12–16 tuntia lukuvuoden aikana
- ohjelmointiharjoitukset on liitetty opiskeltaviin matematiikan sisältöihin ja vahvistavat ja syventävät myös matematiikan taitoja
- harjoitellaan ensin ohjelmointikielen kirjoittamista ja ohjelmallisia rakenteita ja taitojen kasvaessa siirrytään suunnittelemaan ja toteuttamaan pieniä ohjelmia ja soveltamaan opittuja taitoja itsenäisissä projekteissa
- ohjelmointikielenä Python (myös Scratch)

https://www.editapublishing.fi/ContentFiles/materiaalit/oppimateriaalit/Opodemot/Python_harjoitus_1_opettajan_ohjeet.pdf

Esimerkkejä peruskoulun tehtävistä

- Kirjoita funktio, joka laskee kahden annetun luvun keskiarvon.
- Kirjoita funktio, joka tulostaa a) kahdesta b) kolmesta c) n:stä annetusta luvusta suurimman.
- Kirjoita funktio, joka tulostaa annetun luvun kertotaulun.
- Kirjoita funktio, joka laskee annetun luvun kertoman.
- Kirjoita funktio, joka tarkistaa annetun luvun jaollisuuden luvulla 7.
- Kirjoita funktio, joka jakaa kokonaisluvun tekijöihin.
- Kirjoita funktio, joka tulostaa annetun luvun neliöjuuren likiarvon haarukoimalla.

LOPS2019 ja ohjelmointi

MAA11 Algoritmit ja lukuteoria (2 op)

Tavoitteet

Moduulin tavoitteena on, että opiskelija

- tietää, mikä on algoritmi, sekä oppii tutkimaan, kuinka algoritmit toimivat
- oppii toteuttamaan yksinkertaisia algoritmeja ohjelmoidulla
- perehtyy logiikan käsitteisiin
- hallitsee lukuteorian peruskäsitteet ja perehtyy alkulukujen ominaisuuksiin
- osaa tutkia kokonaislukujen jaollisuutta.

Keskeiset sisällöt

- algoritmisen ajattelun peruskäsitteet: peräkkäisyys, valinta ja toisto
- vuokaavio
- yksinkertaisten algoritmien, lajittelualgoritmien tai yhtälön numeeriseen ratkaisuun liittyvän algoritmin ohjelmointi
- konnektiivit ja totuusarvot
- kokonaislukujen jaollisuus, jakoyhtälö ja kongruenssi
- Eukleideen algoritmi
- aritmetiikan peruslause

Opetushallituksen ja Matematiikan opettajien liiton tukimateriaali ehdottaa seuraavaa

Ohjelmistotaidot

Moduulissa käytetään ohjelmointia apuna, kun tutkitaan lukujen ominaisuuksia ja erilaisia algoritmeja. Keskeisenä päämääränä on kokonaisuuksien ymmärtäminen ja mielenkiinnon herättäminen.

Esimerkiksi: Laadi ohjelma, joka

- tulostaa lukujen a ja b jakoyhtälön
- tulostaa jäännösluokka $a \bmod n$ sata pienintä positiivista jäsentä
- tutkii, onko n alkuluku peräkkäisillä jakolaskuilla, joissa jakajina $2, 3, 4, \dots, \sqrt{n}$.

Ohjelmoimalla voidaan ratkaista lisäksi esimerkiksi seuraavia ongelmia:

- neliöjuuren likiarvon laskeminen
- funktion nollakohdan etsiminen puolitusmenetelmällä
- yhtälön ratkaiseminen Newtonin menetelmällä
- alkulukututkimus Monte-Carlo-menetelmällä.

Ohjelmointi. Ohjelmointi toteutetaan jollakin ohjelmointikielellä, esimerkiksi Pythonilla. Opetussuunnitelman tavoitteita ei voi saavuttaa pelkällä taulukkolaskennalla.

https://maol.fi/app/uploads/2020/01/LOPS2019_MAA_MAOL.pdf

<https://maol.fi/materiaalit/>

Kevät 2021, pitkä matematiikka

8. Pinta-alan arviointi simuloinnilla 12 p.

Tasojoukon A pisteet (x, y) määräytyvät epäyhtälöistä $0 \leq x \leq 2$, $0 \leq y \leq 4$ ja $y \geq x^2$.

Tässä tehtävässä on tarkoitus arvioida joukon A pinta-alaa simulaation avulla käyttämällä sitä tietoa, että todennäköisyys on suoraan verrannollinen pinta-alaan. Arvotaan pisteitä (x, y) suorakulmiosta B , jonka määräävät epäyhtälöt $0 \leq x \leq 2$ ja $0 \leq y \leq 4$.

1. Tee sopivalla ohjelmistolla koodi, joka arpoo 1 000 pistettä suorakulmiosta B ja tulostaa vastauksena niiden pisteiden lukumäärän, jotka kuuluvat joukkoon A . Kerro sanallisesti ja sopivien kuvakaappausten avulla, miten toteutit koodisi. (Vihje: Voit käyttää esimerkiksi taulukkolaskennan satunnaislukugeneraattoria.) (6 p.)
2. Hille ajoi kohdassa 1 tekemänsä koodin 10 kertaa ja sai alla olevat luvut. Laske tulosten keskiarvo ja arvioi tämän perusteella joukon A pinta-alaa. (6 p.)

Hillen koodin tulosteet: 673, 664, 672, 679, 667, 650, 640, 678, 660, 667

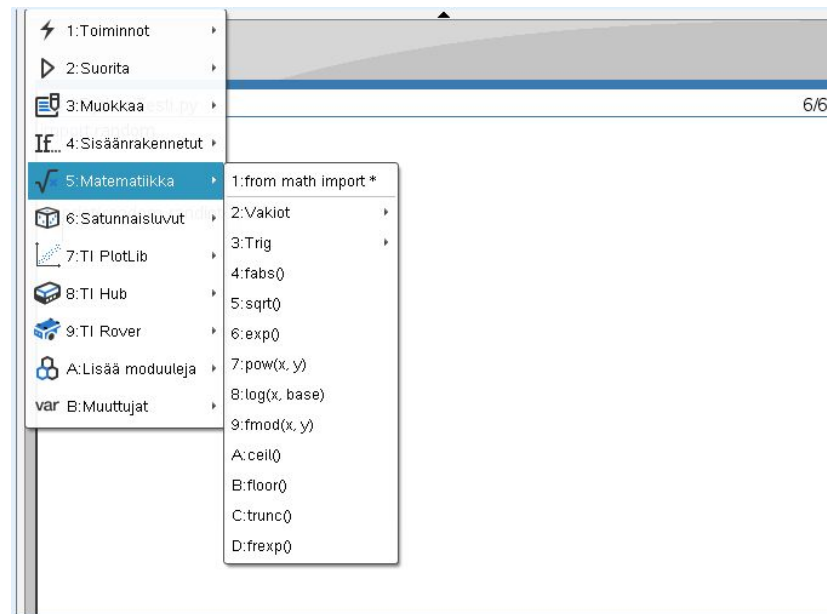
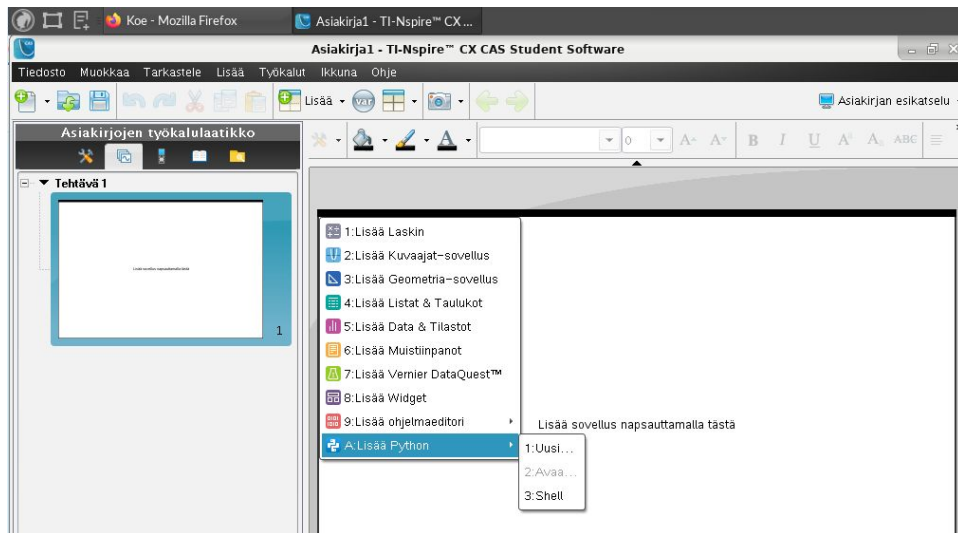
Miten abitissa voi ohjelmoida tällä hetkellä?

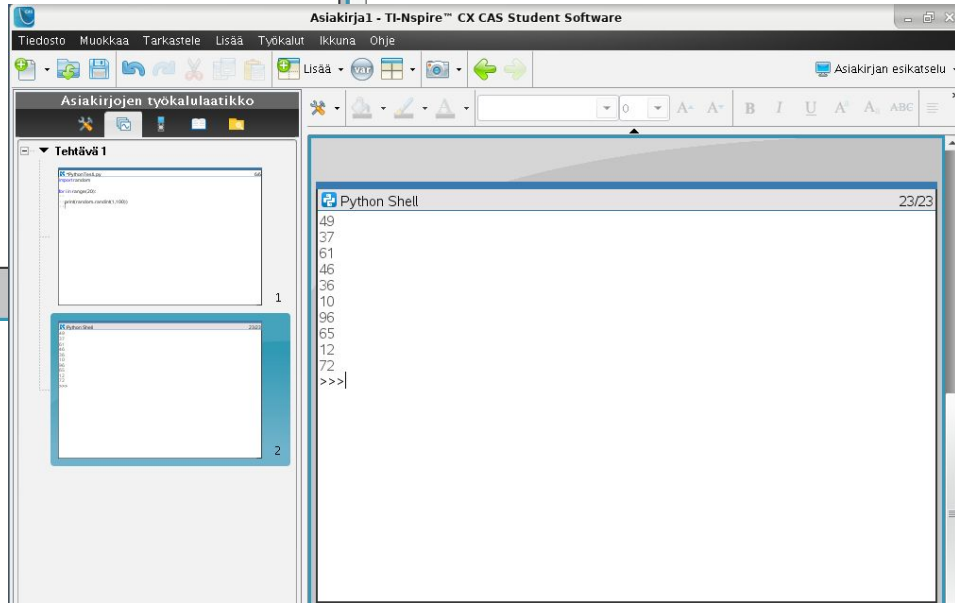
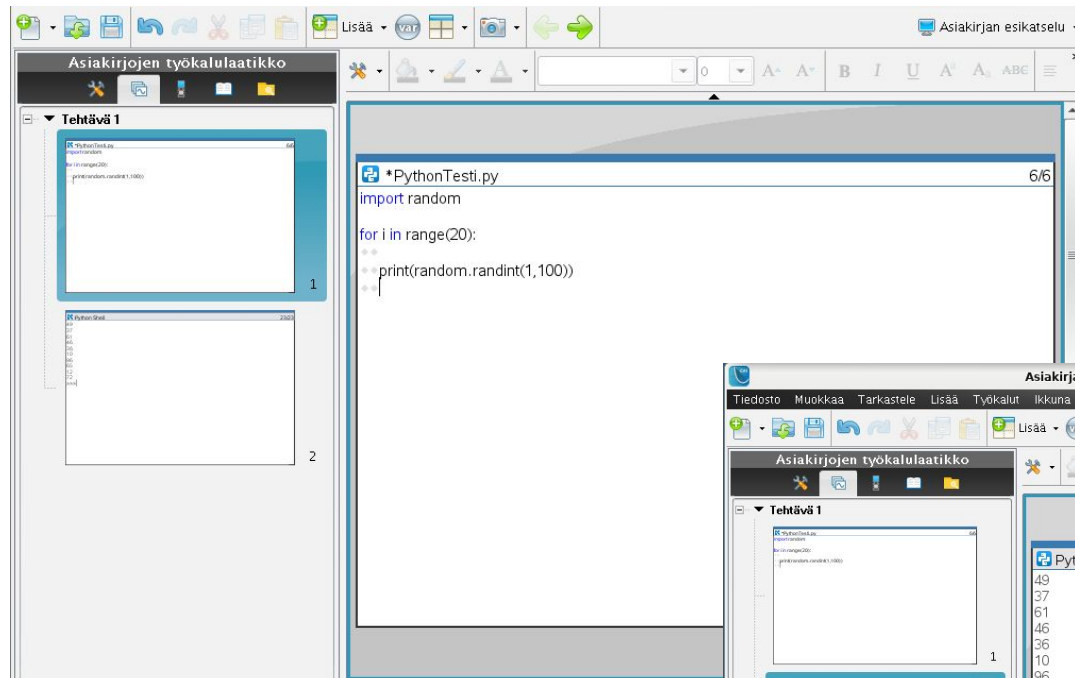
Alkuvuodesta 2021 Abitista löytyvät ainakin nämä mahdollisuudet:

- Javascript (HTML-dokumentti + selain)
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Python (Ti-nSpiren kautta)
- TI-Basic
- Lisää varmaankin tulossa uuden LOPS:n myötä.



Python ja abitti





tiohjelma

1/

Define **tiohjelma()**=

Prgm

☐

EndPrgm



Ohjelmointi Abitissa: Ti-nSpire ja Python

<https://education.ti.com/en/activities/ti-codes/python/nspire>

Aloituis:

<https://screencast-o-matic.com/watch/cYXnYxsLfB>

Silmukka:

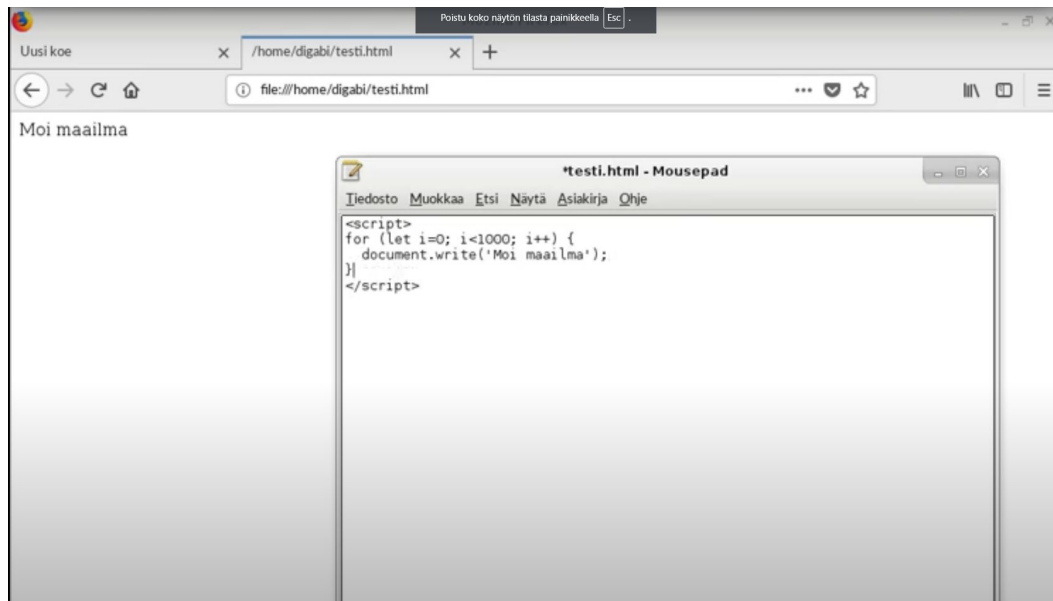
<https://screencast-o-matic.com/watch/cYXnYLsL2B>

Ehtolause:

<https://screencast-o-matic.com/watch/cYXnrAsLyM>



Ohjelmointi abitissa Javascriptiä käyttäen



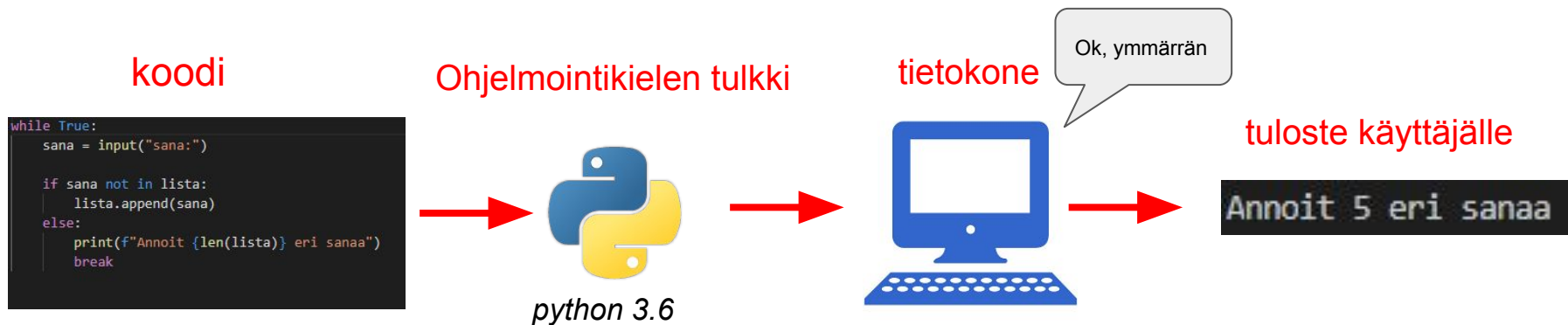
<https://www.youtube.com/watch?v=CsbJl-O8Qx4>



Miksi Python?



- Aloittaminen helppoa (selkeä, yksinkertainen syntaksi, runsaasti oppaita)
- Taipuu moneen (harjoittelu, tieteellinen laskenta, matematiikka)
- Laajalti käytetty



Mitä ohjelmointi on ja miksi se on tärkeää?

- Ohjelmointi on ohjeiden eli käskyjen antamista tietokoneelle.
- Koodaus on ohjeiden kirjoittamista
- Koodi on kieli, jolla maailma pyörii
- Siirry passiivisesta palveluiden käyttäjästä aktiiviseksi sisällön tuottajaksi
- Harjoittaa ongelmanratkaisutaitoja ja loogista ajattelua



Haasteita ja mahdollisuuksia

Haasteita

- en osaa...
- kurssin aikataulu
- eritasoiset opiskelijat
- tekniset ongelmat
- miten osaan valita oikean materiaalin?

Mahdollisuuksia

- ei yhtä “oikeaa” tapaa
- omat sovellukset
- oppiaineiden yhdistäminen
- kokeilemalla oppii
- yhteistyö
- erilaisia materiaaleja runsaasti



Erilaiset alustat Python-ohjelmointiin

Toiveita:

Nopea ja helppo käyttää

Ei ohjelmien asentelua

Windows, Mac ja chromebook -yhteensopiva

Koodin saa talteen



<https://tie.koodariksi.fi/>

- Valmis alusta, ei vaadi asennusta
- Helppokäyttöinen
- Opettajanäkymä
- Valmiita tehtäviä
- Sisältää teoriaa ja esimerkkejä
- Ei omia kokeiluja
- Matikkapainotteinen

Luku 2: Ensimmäinen ohjelma

Kieli: suomi ▼

Tietokoneen ohjelmointi tarkoittaa, että kirjoitamme *ohjelman* eli *koodin*, jossa olevat komennot kertovat koneelle, mitä sen tulee tehdä. Kun ohjelma suoritetaan, kone käy läpi koodirivit yksi kerrallaan ylhäältä alaspäin ja toteuttaa niissä olevat komennot.

Aloitamme ohjelmoinnin komennosta `print`, joka *tulostaa* tekstiä eli näyttää tekstiä ruudulla. Tämän komennon avulla tietokone pystyy näyttämään tietoa ohjelman käyttäjälle.

Ensimmäinen ohjelmamme on tässä:

```
# kolme riviä tekstiä
print("a on apina")
print("b on banaani")
print("c on cembalo")

# lasketaan laskuja
print(1+2)
print(5*3+4)
print(5*(3+4))

# 365 päivää, 24 tuntia, 60 minuuttia, 60 sekuntia
print("vuodessa on", 365*24*60*60, "sekuntia")
```

Kun suoritamme ohjelman, tietokone tulostaa seuraavat rivit:

```
a on apina
b on banaani
c on cembalo
3
19
35
vuodessa on 31536000 sekuntia
```

Katsotaan nyt tarkemmin, miten ohjelma on muodostunut. Koodin ensimmäisellä rivillä on *kommentti*, jonka edessä on risuaita `#`. Tämä on ohjelmoijan tekemä huomautus, josta tietokone ei välitä. Kun kone suorittaa koodin, se hyppää yli kaikki kommentit, joiden alussa on risuaita.

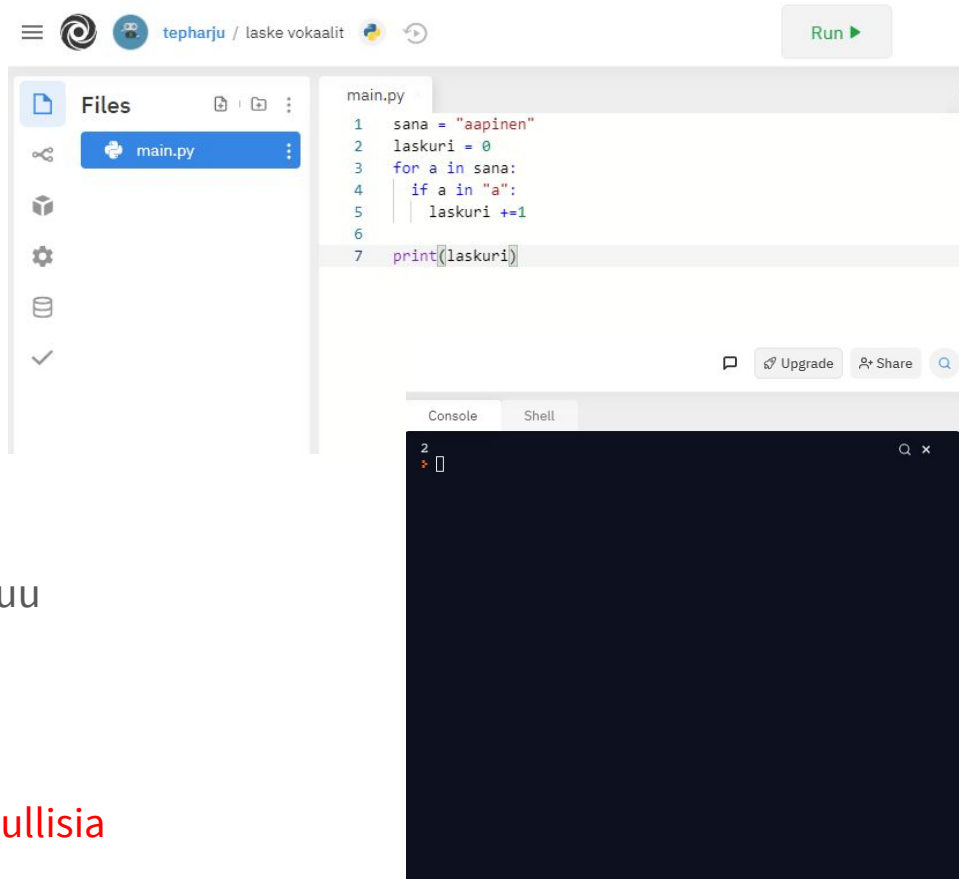
Tämän jälkeen koodissa on kolme `print`-komentoa, joista jokainen tulostaa rivin tekstiä. Teksti on annettu lainausmerkkien sisällä, mikä tarkoittaa, että kone tulostaa tekstin sellaisenaan. Näiden komentojen jälkeen koodissa on tyhjä rivi, jonka kone jättää huomiotta. Voimme laittaa koodiin tyhjiä rivejä aina halutessamme.



replit.com

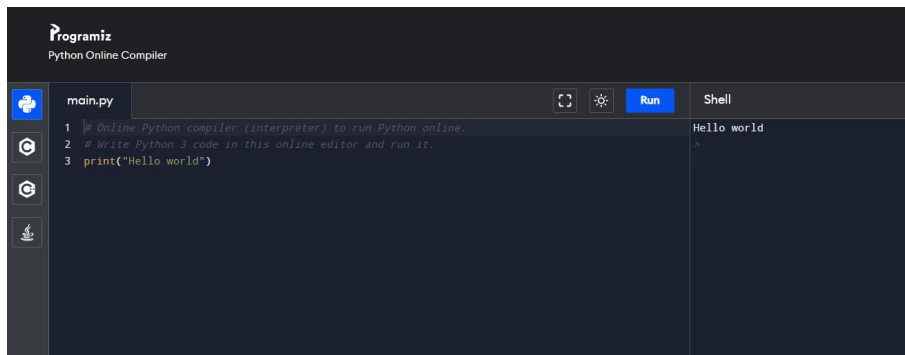
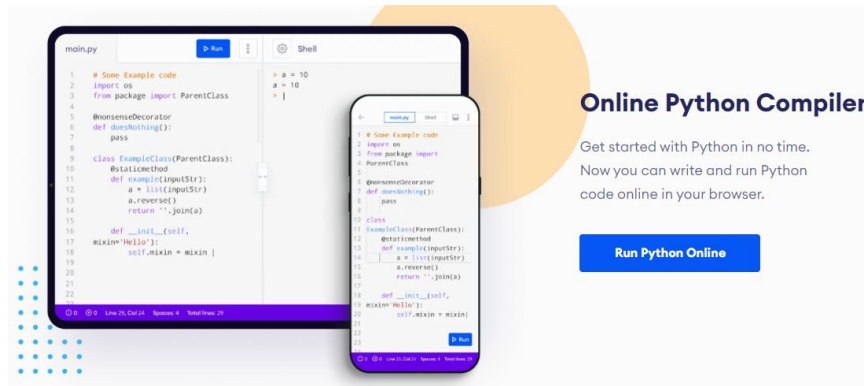
- Ei asennettavia ohjelmia
- Ilmainen (for now...)
- Selkeä ja helppokäyttöinen
- Koodit saa talteen
- Kirjaston yms. lisääminen onnistuu
- Myös mm. C++, Ruby, Javascript, onnistuu
- Vaatii tunnuksen
- Joskus “hidas”
- Lisäominaisuudet (ryhmät, yms.) maksullisia

<https://replit.com/>



Programiz

- Valmis alusta, ei vaadi asennusta
- Sopii nopeaan kokeiluun
- Toimii myös kännykällä



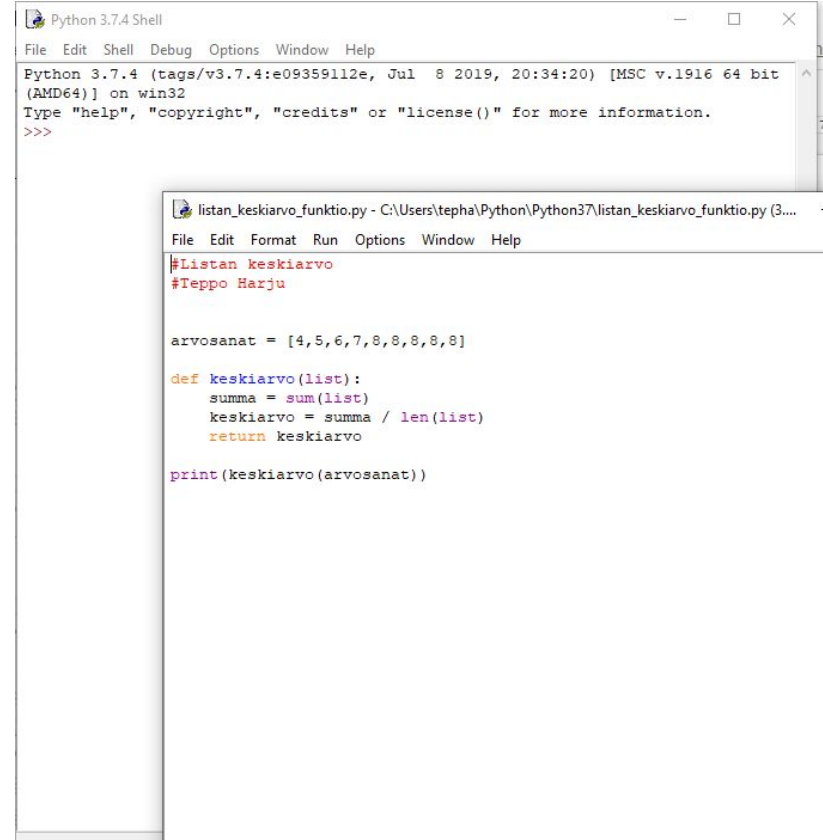
<https://www.programiz.com/python-programming/online-compiler/>



Asennus omalle koneelle: Python 3.x ja IDLE

- Voi kirjoittaa omia ohjelmia
- Koodit säilyvät tallessa
- Kirjastojen yms. lisääminen onnistuu
- Onnistuuko asennus?
- Käytön opettelu vie hieman aikaa ja vaivaa
- Erilaiset koneet - nippu erilaisia ongelmia?
- Onko fokus oikeassa asiassa?

<https://www.python.org/>



The image shows two overlapping windows from a Python 3.7.4 installation. The top window is the 'Python 3.7.4 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a command prompt showing the version and architecture: 'Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32'. The bottom window is a text editor showing a Python script named 'listan_keskiarvo_funktio.py'. The script contains a function 'keskiarvo' that calculates the average of a list and a call to print the result for a specific list of numbers.

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>

listan_keskiarvo_funktio.py - C:\Users\tepha\Python\Python37\listan_keskiarvo_funktio.py (3...
File Edit Format Run Options Window Help
#Listan keskiarvo
#Teppo Harju

arvosanat = [4,5,6,7,8,8,8,8]

def keskiarvo(list):
    summa = sum(list)
    keskiarvo = summa / len(list)
    return keskiarvo

print(keskiarvo(arvosanat))
```

Asennus koneelle: Python Anaconda



- Yhdessä latauksessa
 - Python-tulkki
 - Spyder-editori
 - Kaikki tärkeät tiedepaketit valmiiksi asennettuna
- Ilmainen tämäkin

A screenshot of the Spyder Python IDE interface. The main editor window shows a Python script named 'collatz.py' with the following code:

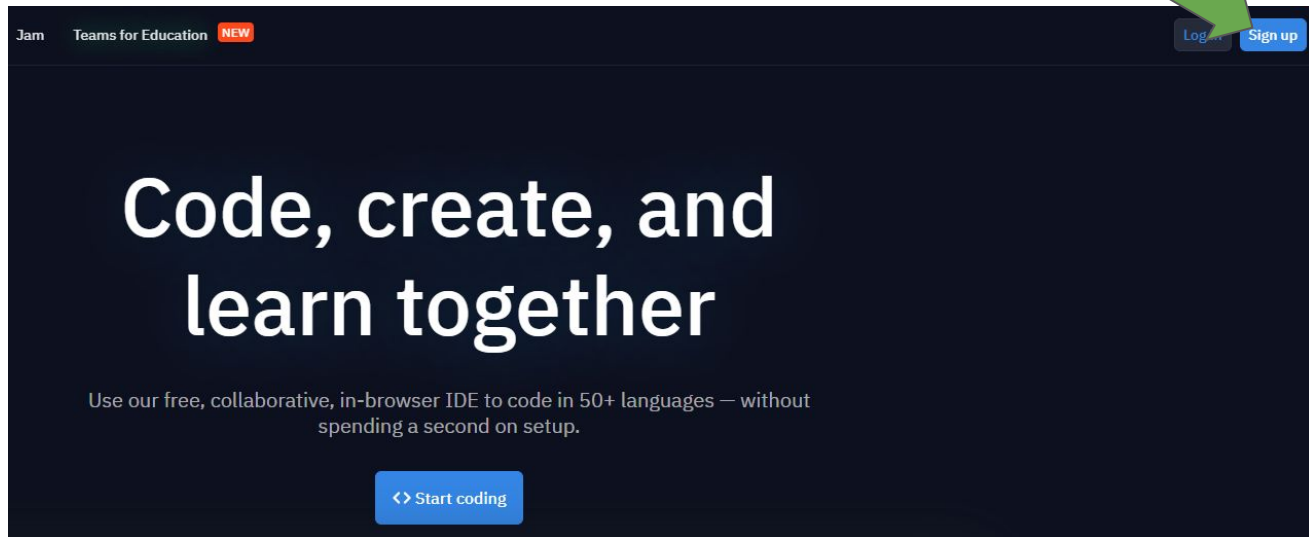
```
1 # Collatzin konjektuuri
2
3 luku = int(input("Anna kokonaisluku: "))
4
5 print(luku)
6
7 while (luku > 1):
8     if luku % 2 == 0:
9         luku = luku//2
10
11     else:
12         luku = 3*luku+1
13
14     print(luku)
15
16
17
18
19
```

The console window on the right shows the output of the script, which is a sequence of numbers: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. The console also shows the Python version (3.8.5) and the IPython version (7.19.0).

Tänään käytettävät ympäristöt

- <https://tie.koodariksi.fi/maa11>
- <https://replit.com>

Tee nyt tili replit.com -sivulle.



Ensimmäinen ohjelma

```
print("Hei maailma!")
```

```
print("Tervetuloa mukaan!")
```

```
# Tämä on kommentti
```

```
print("lasketaan laskuja")
```

```
print(2+2)
```

```
print("2+2")
```

```
print(5*(3+4))
```

Komennot suoritetaan järjestyksessä ylhäältä alas

Jos rivin alussa on #-merkki, ei rivi vaikuta ohjelman toimintaan

Lainausmerkeillä merkitään *merkkijono*, joka tulostuu ruudulle sellaisenaan



Esimerkkiohjelma 1.

- Kirjoitetaan ohjelma, joka tervehtii käyttäjää ja tulostaa laskuja

```
Terve käyttäjä!  
Minä lasken sinulle muutaman laskun!  
10  
10  
25  
5+5  
5*2  
❏
```



Syntaksi (engl. *syntax*), eli lauseoppi

- Syntaksi määrittää miten koodi tulee kirjoittaa
- Jokaisella ohjelmointikielellä on omanlainen syntaksinsa

```
nimi = input("Kerro nimesi:")
```

```
ika = int(input("Kerro ikäsi:"))
```

Python-kielessä

- Isot ja pienet kirjaimet merkitseviä
- Varatut sanat (*False, if, return, def, while*)
- Yksirivinen kommentointi alkaa #-merkillä
- Koodilohkoja ei määritellä aaltosulkeilla vaan *sisennyksin*
- Lohkon alkaessa kaksoispiste (*def funktio():* tai *if muuttuja < 10:*)
- Puolipisteitä ; ei käytetä rivin lopussa



Muuttujat

- Muuttujiin voi tallentaa tietoa, jota tarvitaan ohjelmassa myöhemmin
- Asetetaan muuttujaan nimeltä *nimi* arvo *Topias*
- Muuttujaan **asetetaan arvo** yhtäsuuruusmerkillä (=)

Muuttujan nimi \longrightarrow `nimi = "Topias"` \longleftarrow Muuttujan arvo

`a = 3`

`summa = 5`

`tulos = 5 * summa`



Muuttujat

- Muuttuja on kuin lokero, johon voi tallentaa tietyn *tyypistä* tietoa
- Tiedolla on aina tyyppi. (*teksti, kokonaisluku jne.*)
- Asetettaessa arvo, kyse ei ole yhtäsuuruudesta vaan arvon asettamisesta

`a = 3`

`a = a + 1`

`5 = b`, *ei toimi...*



Muuttujien nimeäminen

- Muuttujan nimi ei saa sisältää erikoismerkkejä (*!!**super**!!*, *rahat\$\$*)
 - *SyntaxError: invalid character in identifier*
- Muuttujan nimen täytyy alkaa kirjaimella (*5kissaa*, *1ala*)
 - *SyntaxError: invalid syntax*
- Isot kirjaimet (*nimi*, *Nimi* ja *NIMI* ovat eri muuttujia)
- Muuttujan nimi ei saa olla avainsana (*if*, *else*, *False* jne.)
 - *SyntaxError: invalid syntax*
- Käytä kuvaavia nimiä:
(*sana*, *laskuri*, *ala*, *summa*, *kaikki_luvut*, *eka_sana*)



Lisää tulostamisesta

```
nimi = "Matti"
```

```
ika = 78
```

```
print(nimi, "jää tänään eläkkeelle.")
```

```
print(nimi, "täyttää tänään", ika, "vuotta.")
```

pilkku luo välilyönnin

```
print("Nimesi on " + nimi + " ja olet" + str(ika) + "vuotta.")
```

srt(muuttuja)

```
print(f"Onneksi {nimi}, joka on {ika}-vuotias jää nyt eläkkeelle.")
```

python 3.6 ->



laskutoimituksia ja math-kirjasto

Operaattori	Merkitys	Esim.	Tulos
+	yhteenlasku	$2 + 5$	7
-	vähennyslasku	$10 - 2$	8
*	kertolasku	$-2 * 5$	-10
/	jakolasku (desimaali)	$9 / 2$	4.5
//	jakolasku (kokonaisluku)	$9 // 2$	4
%	jakojäännös	$9 \% 2$	1
**	potenssi	$2^{**}3$	8



Syötteen lukeminen käyttäjältä, *input-komento*

```
nimi = input("Anna nimesi:")
```

*input lukee käyttäjältä merkkijonon ja
asettaa sen muuttujan nimi arvoksi*

```
syote = input("Minä vuonna olet syntynyt? ")
```

```
vuosi = int(syote)
```

*merkkijono on muutettava kokonaisluvuksi, jolloin
laskeminen onnistuu*

```
print("Ikäsi vuoden lopussa on:", (2021 - vuosi))
```

```
print(f"Ikäsi vuoden lopussa on: {2021-vuosi}")
```



Esimerkkiohjelma 2.

- Kirjoitetaan ohjelma, joka laskee kolmion pinta-alan annetuilla arvoilla.

```
Kanta (m):5  
Korkeus (m):6  
Kolmion pinta-ala on: 15.0 m^2  
➤
```



Harjoitellaan!

<https://tie.koodariksi.fi/maa11>

- Tehtävät 1. - 6.

Harjoitustehtäviä osa 1.

Materiaalikansio: bit.ly/ohjelmointimaol

<https://www.pythoncheatsheet.org/#Python-Basics>

How to actually learn any new programming concept



Essential

Changing Stuff and
Seeing What Happens



Tehtäviä osa 1 .

1. Tee ohjelma, joka kysyy käyttäjän nimeä ja ikää. Ohjelma tulostaa tervehdyksen ja laskee käyttäjän syntymävuoden ja elinajan sekunneissa. (*Vihje input*)
2. Tee ohjelma, joka kysyy kahta kokonaislukua. Ohjelma tulostaa lukujen summan, tulon, erotuksen ja osamäärän.
3. Ruokakustannukset. Tee ohjelma, joka arvioi viikottaisia ruokakustannuksia. Ohjelma kysyy työpaikkaruokailun hintaa, määrää ja muiden viikottaisten ruokaostosten hintaa. Ohjelma kertoo keskimääräiset ruokamenot päivässä ja viikossa.

