

# Ohjelmointi lukion matematiikassa, mitä opettaa opintojaksossa MAA11?

Edistynyt ryhmä,  
osa 1.



**Materiaalikansio:**

**[bit.ly/ohjelmointimaol](https://bit.ly/ohjelmointimaol)**

# Hello world!

**Teppo Harju**, fysiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio,  
Harppi-hanke

**Ville Tilvis**, matematiikan opettaja, Maunulan yhteiskoulu ja Helsingin matematiikkalukio,  
Harppi-hanke

**Mika Setälä**, matematiikan opettaja, Lempäälän lukio

**Antti Laaksonen**, yliopistonlehtori, Helsingin yliopisto, tietojenkäsittelytieteen osasto



# Hei maailma!

Kerro lyhyesti:

Kuka olet, mistä tulet ja mikä fiilis?



# Koulutuksen tarkoitus ja päivän tavoite

- Ohjelmointi tulee lukioon: mitä meiltä odotetaan?
- Ympäristöt ja alustat
- Pythonin pikakertaus
- Matematiikkaa Pythonilla
- Ideoita moduulin maa11 opettamiseen

## Päivän päätteeksi:

- Osaan valita itselleni sopivan alustan ohjelmoinnin opetukseen
- Olen oppinut matematiikkaa / ohjelmointia
- Minulla on ajatus opintojakson Maa11 toteuttamisesta
- Tiedän mistä saan lisätietoa ja mitä kokeilen seuraavaksi



# Päivän aikataulu to 3

9.00 - Hei kaikki, esittelyt, tämä dia

9.15- Opetussetti 1.

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

11.30 - 12.30 Lounas

12.30 Opetussetti 2

- Yo-tehtäviä ohjelmoimalla
- Hieman järjestelyalgoritmeista

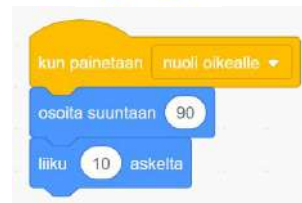
13.45 Maa11-opintojakson suunnittelu

15.00 - Lopetus



# POPS 2014

	Laaja-alainen osaaminen: Tieto- ja viestintäteknologinen 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. <b>Ohjelmoidaan ja samalla harjoitellaan hyviä ohjelmointikäytäntöjä.</b>	T20 Oppilas osaa soveltaa algoritmisen ajattelun periaatteita ja <b>osaa ohjelmoida yksinkertaisia ohjelmia.</b>



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

## Arvioinnin kohde

Algoritminen ajattelu ja ohjelmointitaidot

### Matematiikan päättöarvioinnin kriteerit oppimäärän päättyessä

Osaamisen kuvaus	Arvion kuvaus
Osaamisen kuvaus arvosanalle 5	Oppilas tunnistaa yksinkertaisen algoritmin askeleet ja testaa ohjattuna valmiita ohjelmia.
Osaamisen kuvaus arvosanalle 7	Oppilas käyttää ehto- ja toistorakennetta ohjelmoinnissa sekä testaa ja tulkitsee ohjelmia.
Osaamisen kuvaus arvosanalle 8	Oppilas soveltaa algoritmisen ajattelun periaatteita ja ohjelmoi pieniä ohjelmia.
Osaamisen kuvaus arvosanalle 9	Oppilas hyödyntää ohjelmointia ongelmien ratkaisussa. Oppilas muokkaa ja kehittää ohjelmaa.



# 7-9 lk ohjelmointitaitojen kuvaukset, suositus

<https://uudetlukutaidot.fi/> (OPH)

Looginen ajattelu ja tiedon käsittely	Ongelmien ratkaiseminen ja mallintaminen	Ohjelmoinnin käsitteet ja perusrakenteet	Käytännön taidot
Oppilas käsittelee erilaisiin yleistyksiin sisältyviä tietoja, käyttää erilaisia merkintätapoja ja toteuttaa erityyppisen tiedon kanssa loogisia operaatioita.	Oppilas analysoi ongelmia ja arvioi niiden mahdollisia ratkaisuja erilaisten kriteerien perusteella sekä visualisoi ongelmia ja ratkaisuja yleistysten ja kaavioiden avulla.	Oppilas ymmärtää algoritmin merkityksen ja osaa suunnitella ohjelman, jossa hyödynnetään tarkoituksenmukaisesti ohjelmoinnin perusrakenteita, kuten peräkkäisiä, toistuvia ja ehdollisia toimintoja.	Oppilas ohjelmoi ohjelmia eri ympäristöissä sekä tuntee perusasiat yhdestä tekstipohjaisesta ohjelmointikielestä ja osaa tulkita sillä tehtyä ohjelmakoodia.

# LOPS 2021-> 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 ohjelmoimalla
- 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/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

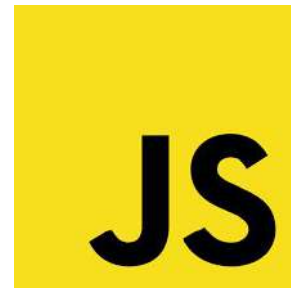
**Abitti tällä hetkellä**



# Miten abitissa voi ohjelmoida tällä hetkellä?

Alkuvuodesta 2021 Abitista löytyvät ainakin nämä mahdollisuudet:

- Javascript (HTML-dokumentti + selain)
- Python (Ti-nSpiren kautta)
- TI-Basic
- Lisää varmaankin tulossa...



# Mitä sanoo YTL? (epävirallisesti)

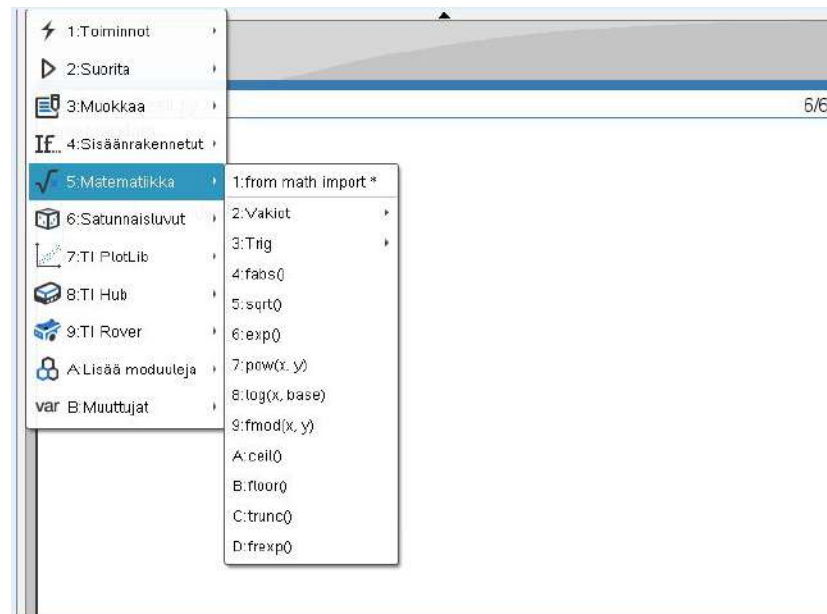
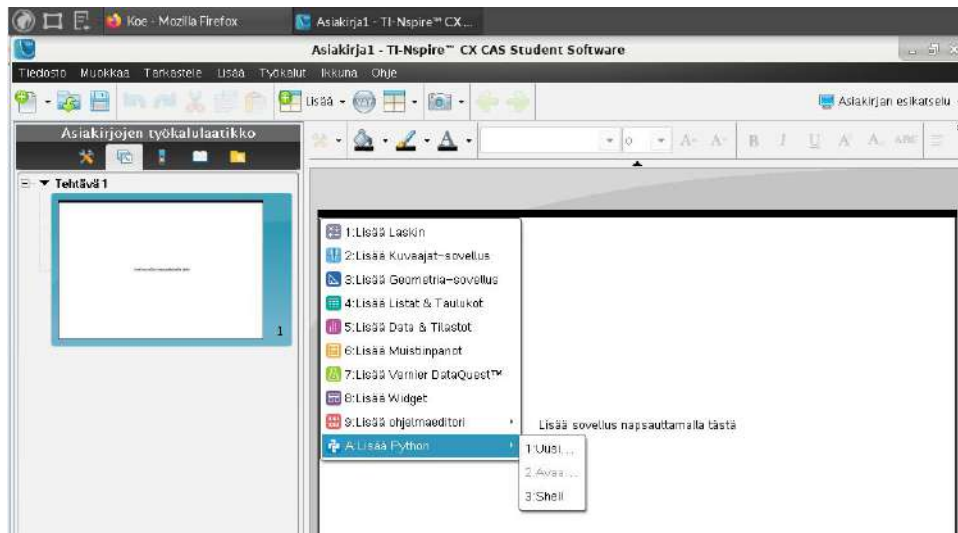
*Lisää ohjelmointiympäristöjä on todennäköisesti tulossa vuonna 2022.*

*Tällä hetkellä Abitista löytyy Pythonin lisäksi Perl-tulkki sekä tietysti sh ja bash.*

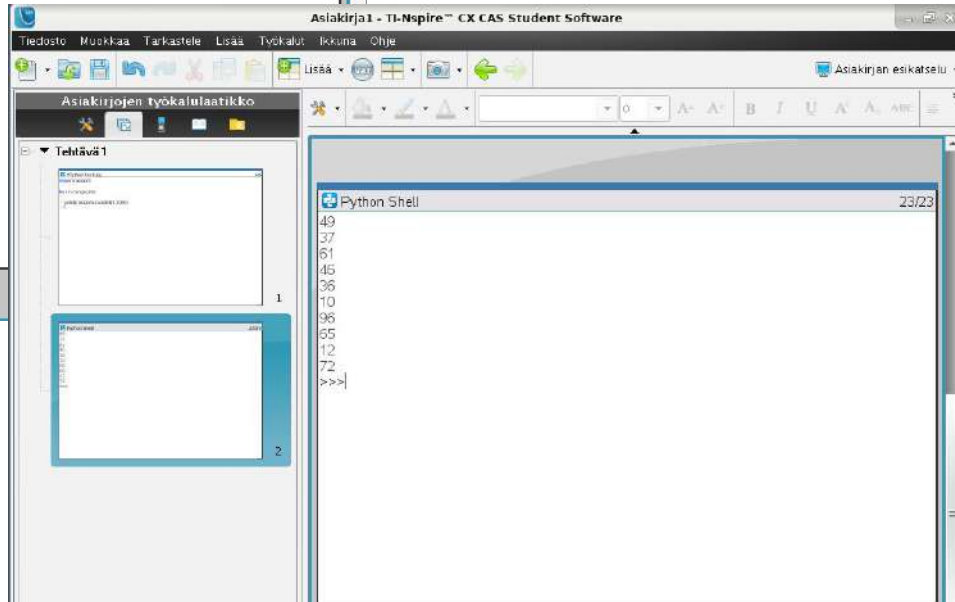
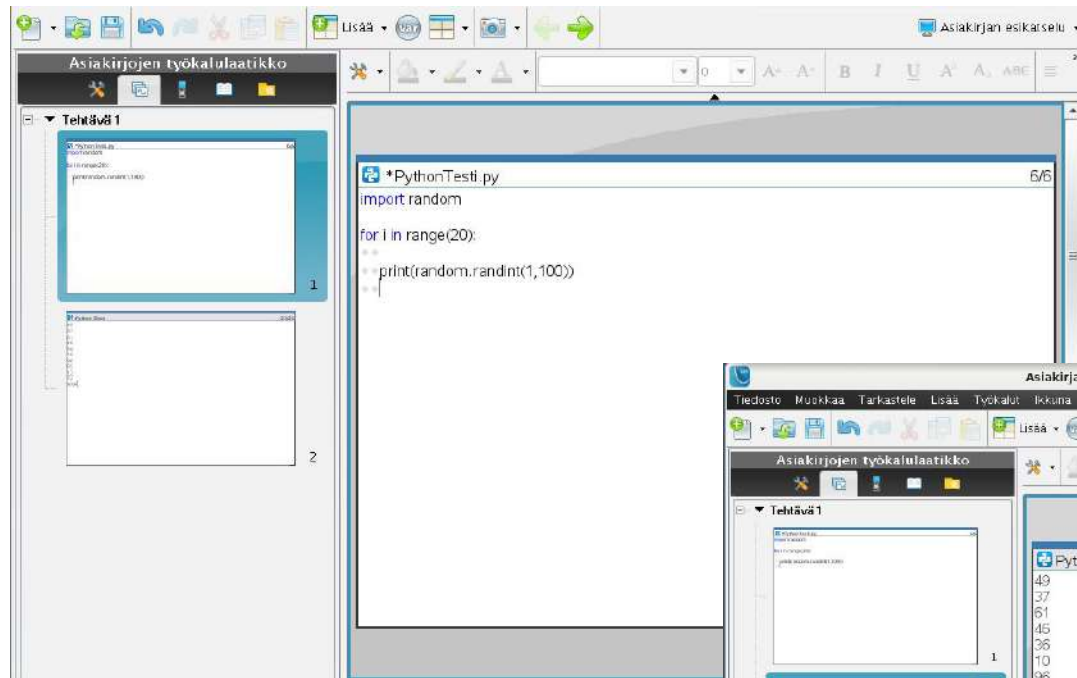
*Ohjelmia voi kirjoittaa kotihakemistoon ja ajaa ne lisäämällä skriptille suoritusbitin. Käytännössä lähtöarvot on annettava muuttujissa ja tulosteet on kirjoitettava tiedostoon.*



# Python ja abitti







## Asiakirjojen työkalulaatikko

Python-editor

- 1 Toiminnot  
 2 Suorita  
 3 Muokkaa  
 4 Sisäänrakennetut  
 5 **Matematiikka**  
 6 Satunnaisluvut  
 7 Ti PlotLib  
 8 Ti Hub  
 9 Ti Rover  
 A Lisää moduuleja  
 vär B Muutustat

# TI-Basic

```
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:

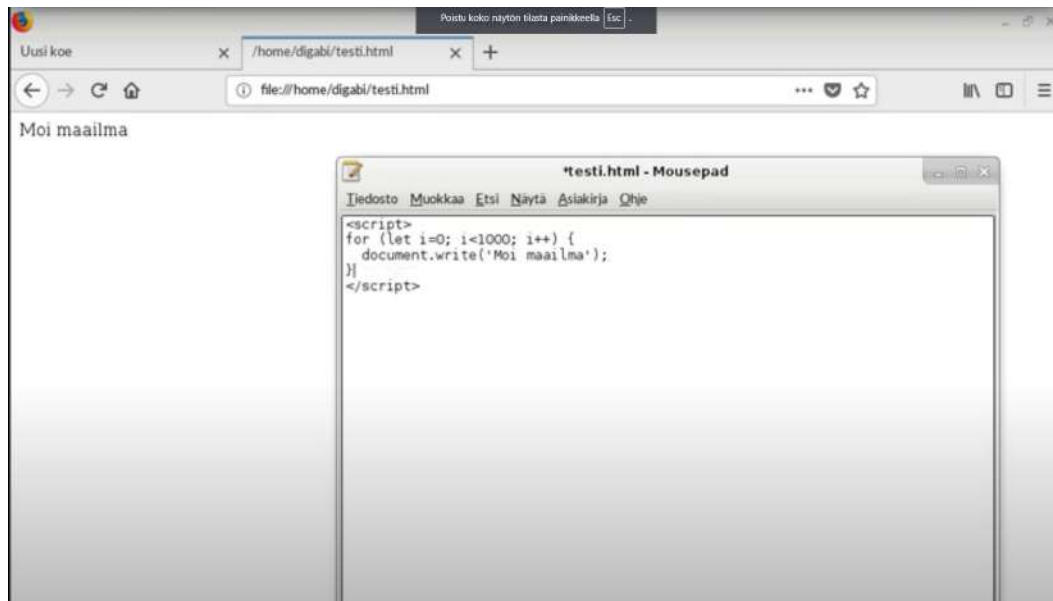
<https://screencast-o-matic.com/watch/cYXnYLsL2B>

Ehtolause:

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



# Ohjelmointi abitissa Javascriptiä käyttäen



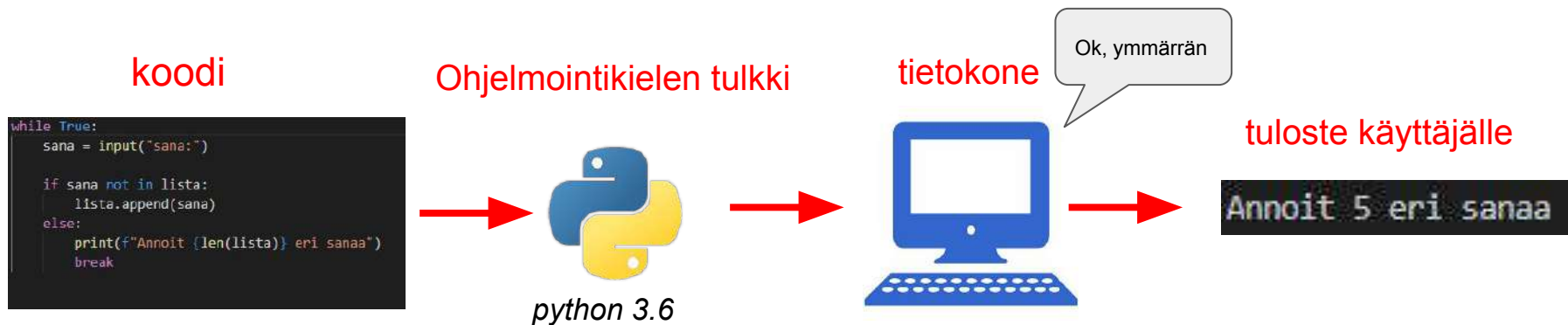
<https://www.youtube.com/watch?v=CsbJl-O8Qx4>



# Miksi Python?



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



# Erilaiset alustat Python-ohjelmointiin

## Opettajien toiveita:

Nopea ja helppo käyttää

Ei ohjelmien asentelua

Windows, Mac ja chromebook -yhteensopiva

Koodin saa talteen



**Alustat: missä koodia kirjoitetaan?**





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

- 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 koodinrivit 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.

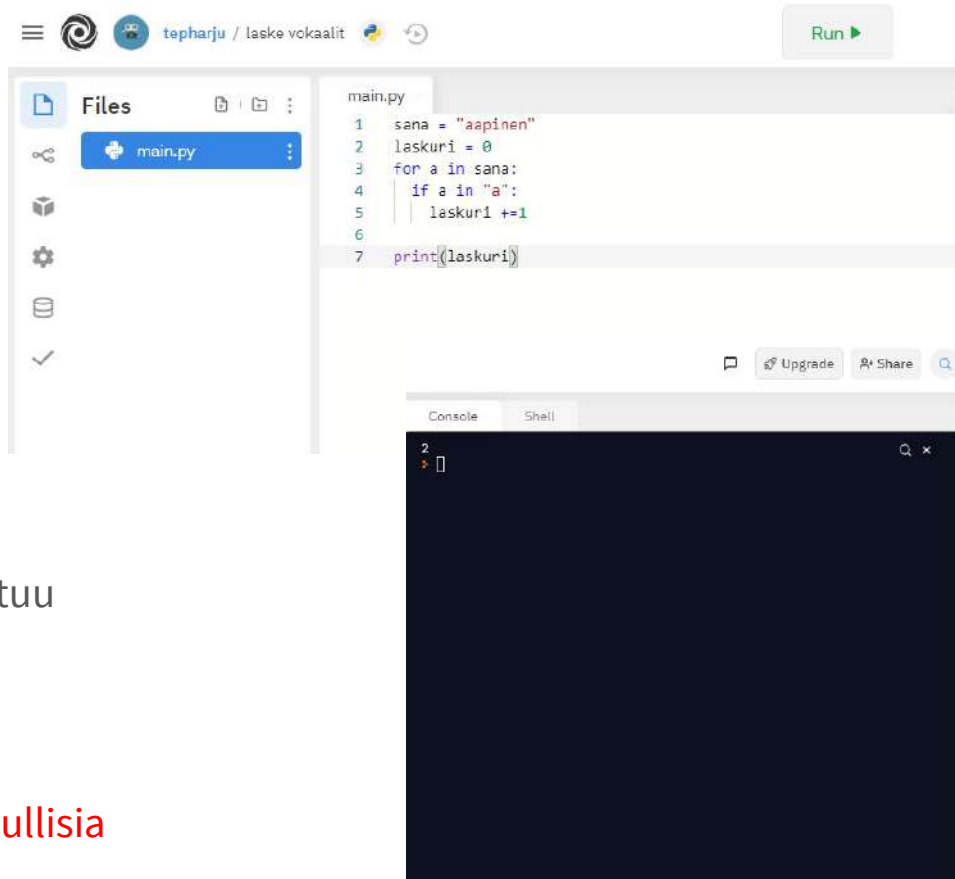
Tämän jälkeen koodissa on kolme `print`-komentoa, joista jokainen tulostaa rivin tekstiä. Teksti on annettu lainausmerkkien sisällä, mikä tarkoittaa, että kone tulostaa tekstin sellaisenaan. Näiden komentojen jälkeen koodissa on tyhjä rivi, jonka kone jättää huomiotta. Voimme laittaa koodiin tyhjiä rivejä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



# replit.com

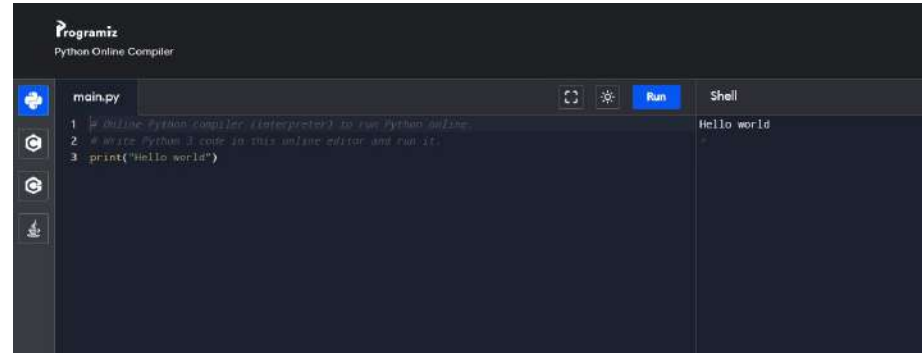
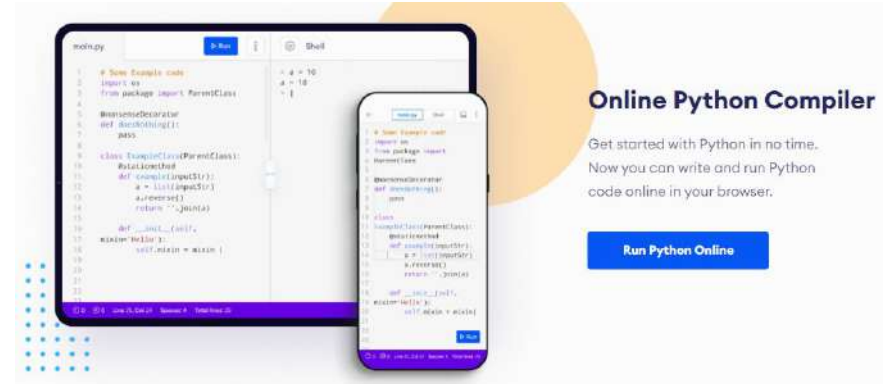
- Ei asennettavia ohjelmia
- Ilmainen (for now...)
- Selkeä ja helppokäyttöinen
- Koodit saa talteen
- Kirjaston yms. lisääminen onnistuu
- Myös mm. C++, Ruby, Javascript, onnistuu
- Vaatii tunnuksen
- Joskus “hidas”
- Lisäominaisuudet (ryhmät, yms.) maksullisia

<https://replit.com/>



# Programiz

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



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

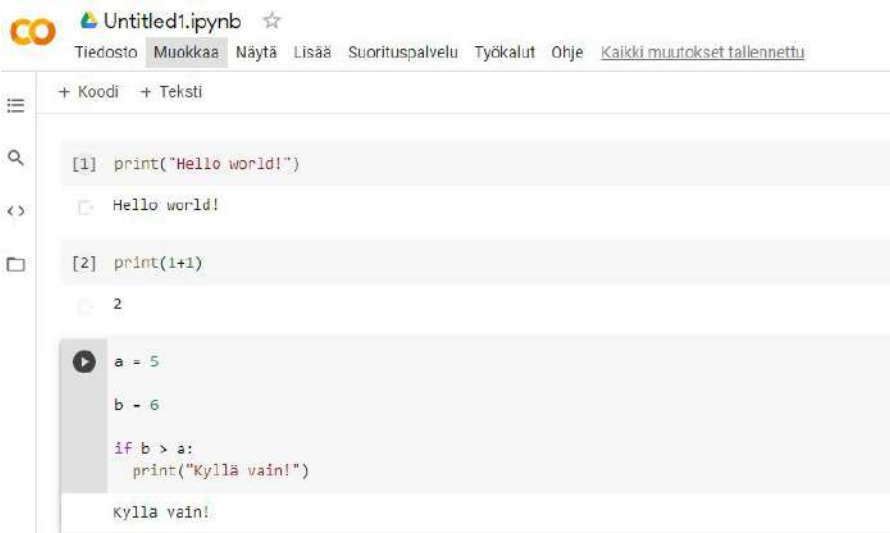


# Google Colab

- Kirjoita ja suorita koodia selaimessa
- Dokumenttiin voi lisätä myös tekstiä ja dokumentteja voi jakaa google docs -tyyliin.

[colab.research.google.com](https://colab.research.google.com)

<https://jupyter.org/>



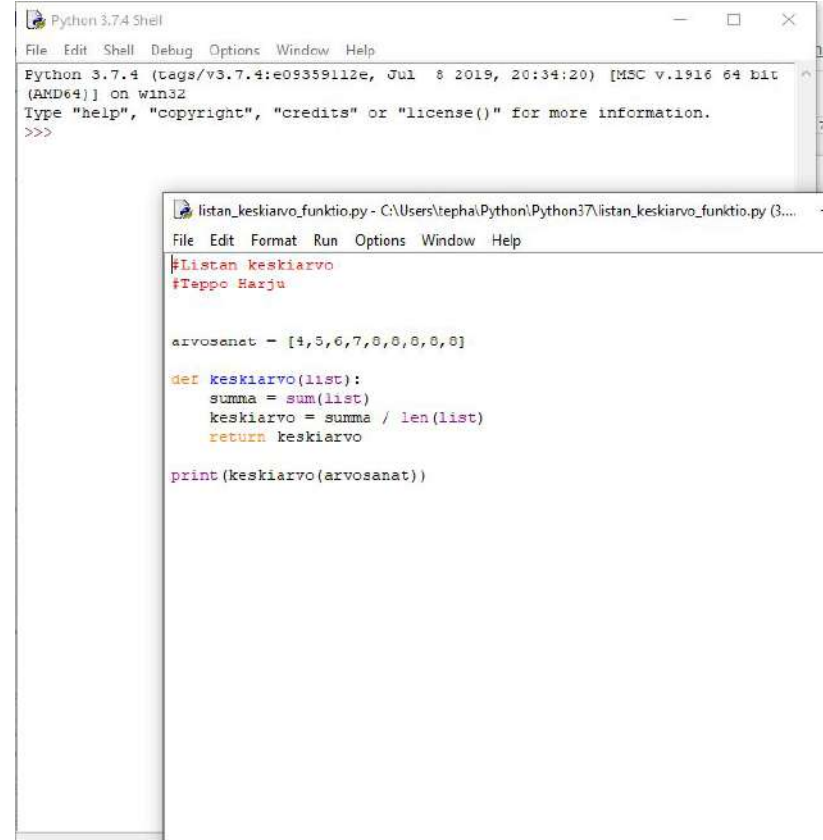
The screenshot shows the Google Colab web interface. At the top, there's a title bar with the Colab logo, the filename 'Untitled1.ipynb', and a star icon. Below the title bar is a menu bar with options: 'Tiedosto', 'Muokkaa', 'Näytä', 'Lisää', 'Suorituspalvelu', 'Työkalut', 'Ohje', and 'Kaikki muutokset tallennettu'. The main area is divided into two tabs: '+ Koodi' (selected) and '+ Teksti'. On the left side, there's a sidebar with icons for file explorer, search, and code editor. The code editor shows three code cells. The first cell contains `[1] print("Hello world!")` and its output is 'Hello world!'. The second cell contains `[2] print(1+1)` and its output is '2'. The third cell contains `a = 5`, `b = 6`, and `if b > a: print("Kyllä vain!")`. The output of the third cell is 'kyllä vain!'.



# 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/v3.7.4:09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32'. The bottom window is a text editor titled 'listan\_keskiarvo\_funktio.py' with a menu bar (File, Edit, Format, Run, Options, Window, Help). It contains a Python script that defines a function to calculate the average of a list.

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:09359112e, 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
#Teppe Harju

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

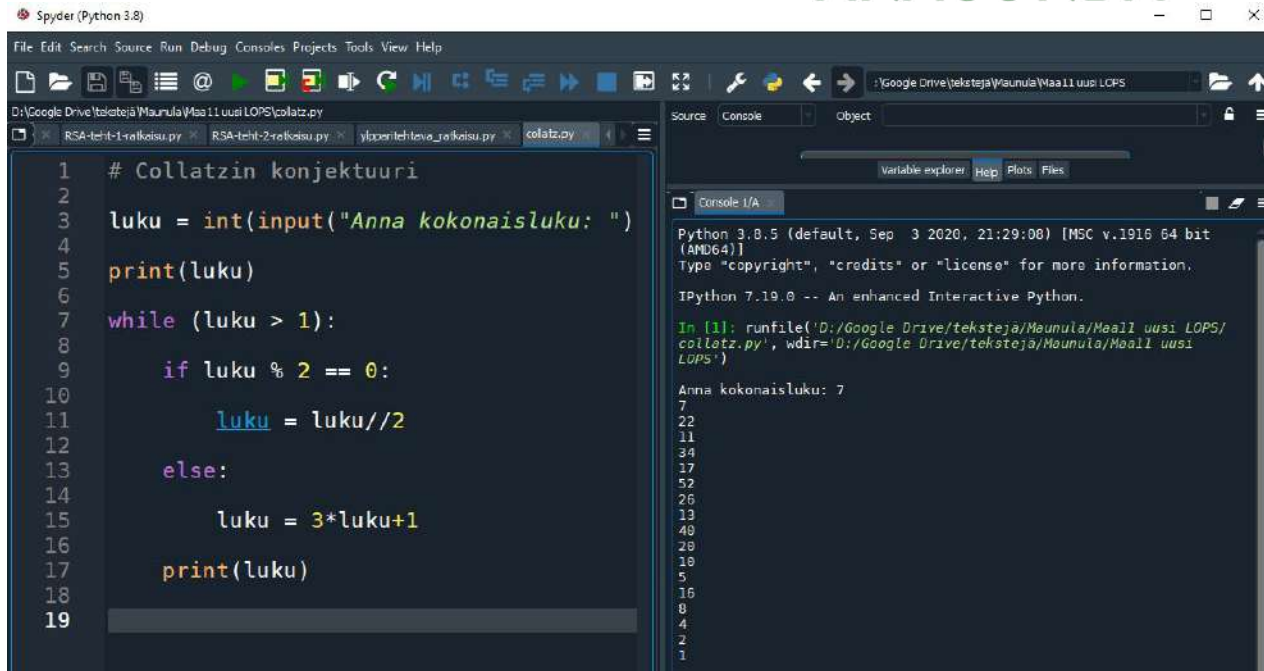
def keskiarvo(list):
    summa = sum(list)
    keskiarvo = summa / len(list)
    return keskiarvo

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    else:
11        luku = 3*luku+1
12
13    print(luku)
14
15
16
17
18
19
```

The right-hand pane shows the IPython console with the following output:

```
Python 3.8.5 (default, Sep  3 2020, 21:29:00) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more information.
IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/Google Drive/teksteja/Maunula/Maa11 uusi LOPS/
collatz.py', wdir='D:/Google Drive/teksteja/Maunula/Maa11 uusi
LOPS')

Anna kokonaisluku: 7
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
```

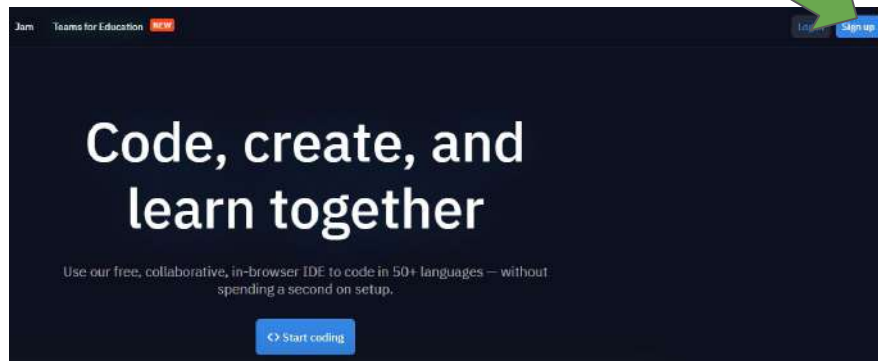
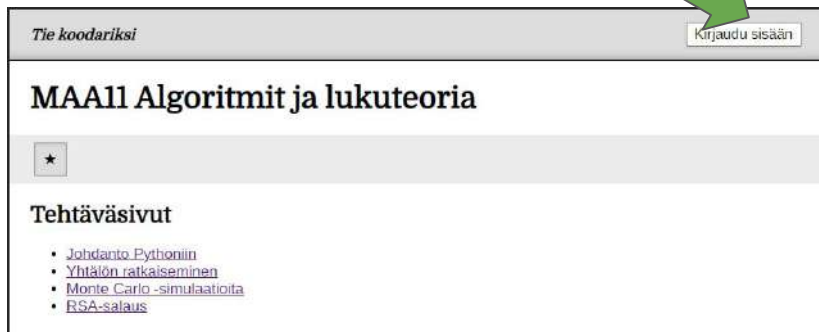
<https://www.anaconda.com/products/individual>

<https://www.spyder-ide.org/>



# Tänään käytettävät ympäristöt

- <https://tie.koodariksi.fi/maa11> ← Valmiita tehtäviä
  - <https://replit.com> ← Avoimet tehtävät ja omat kokeilut, Google-kirjautuminen kätevä
- Tee nyt tili molemmille sivustolle.**



# Pythonin perusteet





# 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 rivi kerrallaan 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  
❏
```



# Tee nyt:

<https://tie.koodariksi.fi/Maa11>

-> Johdanto Pythoniin:

-> Tehtävät 1 - 6

Kirjaudu sisään ja luo tunnus,  
niin koodit säilyvät tallessa

Tie koodariksi

Kirjaudu sisään

MAA11 Algoritmit ja lukuteoria



Tehtäväsivut

- [Johdanto Pythoniin](#)
- [Yhtälön ratkaiseminen](#)
- [Monte Carlo -simulaatioita](#)
- [Liukuluvuista](#)
- [Järjestämisalgoritmit](#)
- [RSA-salaus](#)



# 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 (yksi tapa)

```
nimi = "Matti"
```

```
ika = 78
```

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

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

pilkku luo välilyönnin

↑  
muuttuja

↑  
tekstiä

↑  
muuttuja

↑  
tekstiä





# 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



# math-moduuli

- *math*-moduuli sisältää hyödyllisiä matemaattisia funktioita ja vakioita
- Ensin *math*-moduuli täytyy tuoda mukaan **import**-käskyllä

```
import math
```

Tuodaan koko *math*-moduuli käyttöön

```
print(math.exp(4))      # exp(x) -> Eksponenttifunktio e^x
print(math.sqrt(9))     # sqrt(x) -> Luvun x neliöjuuri (kuten x**(1/2))
print(math.pi)          # pi -> pii (ei ole funktio vaan vakio)
print(math.log(10,2))   # log_2(10)
```



# Oman funktion luonti

```
def f(x):  
    return x**3 + 1
```

```
print( f(2) )
```



# 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!

Materiaalikansio: [bit.ly/ohjelmointimaol](https://bit.ly/ohjelmointimaol)

Harjoitustehtäviä osa 1. (seuraava kalvo)

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

*How to actually learn any new programming concept*



*Essential*

Changing Stuff and  
Seeing What Happens



# Harjoitustehtäviä osa 1 , perustaso

1. Tee ohjelma, joka kysyy käyttäjän nimeä ja ikää. Ohjelma tulostaa tervehdyksen ja laskee käyttäjän syntymävuoden ja elinajan sekunneissa. (*Vihje: input*)
2. Suorakulmaisen kolmion hypotenuusan pituus. Tee ohjelma joka kysyy kolmion sivujen a ja b pituuksia ja laskee hypotenuusan pituuden. (*Vihje: math*)
3. Tee ohjelma, joka kysyy kahta kokonaislukua. Ohjelma tulostaa lukujen summan, tulon, erotuksen ja osamäärän.
4. Ruokakustannukset. Tee ohjelma, joka arvioi viikottaisia ruokakustannuksia. Ohjelma kysyy työpaikkaruokailun hintaa, määrää ja muiden viikottaisten ruokaostosten hintaa. Ohjelma kertoo keskimääräiset ruokamenot päivässä ja viikossa.



# Kysyttävää?





**Ei käytetyt diat ->**



# 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.

# Lisää tulostamisesta (käytetään vain print(n,"teksti"))

```
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 ->

