

# Kemiantekniikan ohjelmointikurssi CHEM-A2600 (5 op)

Antti Karttunen

**Kesäkurssin 2019 aloitusluento 3.6.2019**

Avaa luennon alussa web-sivu: [premo.aalto.fi/a2600](https://premo.aalto.fi/a2600)

# Osa 1: Kurssin kuvaus



# Kurssin opettajat (kesä 2019)



Antti Karttunen  
(lomalla 1.-28.7.)



Kim Eklund

# Kesäkurssin 2019 yhteenveto

- Kurssi kestää kaksitoista viikkoa (3.6.–25.8.)
- Kaksi viikkoa = yksi **kierros**
  - Jokaisella kierroksella on omat aiheensa ja niihin liittyviä ohjelmointitehtäviä
  - Kurssin arviointi perustuu ohjelmointitehtäviin
- Ohjelmointitehtävät on toteutettu MyCoursesin Quiz-toiminnolla
  - CodeRunner-niminen ohjelmisto tarkistaa tehtävät automaattisesti
- Kurssilla pidettäneen myös ohjattuja harjoituksia
  - Ajankohdat päätetään kurssin alussa opiskelijakyselyn perusteella
  - Ei perinteinen "laskuharjoitus" vaan "päivystys", jolloin tarjolla ohjausta
  - Harjoituksissa voi keskustella opettajien ja muiden opiskelijoiden kanssa
- Koska ohjelmointitehtävät voi tehdä millä tahansa tietokoneella ja tehtävien tarkastus on automatisoitu, kurssin **voi suorittaa etäopiskeluna**
  - Tarvittavien ohjelmointityökalujen asennus on ohjeistettu alla (Anaconda)

# Kurssikalenteri

Kierros	Päättyy	Teema
<b>Kurssin aloitusluento maanantaina 3.6. klo 10.15 (Ke2 / A304)</b>		
1	16.6.	Ohjelmoinnin peruskäsitteitä ja Pythonin perusteet (tulkki, IDE). Tulostaminen, syötteen lukeminen Muuttujat, operaattorit ja kontrollirakenteet: if-elif-else, while, for
2	30.6.	Funktiot, moduulit
3	14.7.	Tietorakenteet (listat, monikot, sanakirjat)
4	28.7.	Numpy ja Matplotlib
5	11.8.	Tiedostojen käsittely, virheenkäsittely
6	25.8.	Sovellukset, Scipy, olio-ohjelmoinnin perusteita

**Kaikki tehtäväkierrokset avautuvat kesäkuun aikana!  
(tieto päivitetty 2019-06-07)**

# Ohjatut harjoitukset

- Ensimmäiset ohjatut harjoitukset tiistaina 4.6. klo 14:00–16:00 tilassa **C330b**.
- Muiden harjoitusten ajankohta ja paikka ilmoitetaan myöhemmin opiskelijakyselyn perusteella
  - Vastaa kyselyyn osoitteessa <https://doodle.com/poll/i8gm8myyg8i478dx>
- Tarvitset harjoituksissa **oman läppäriin** (PC/Mac)!
  - Harjoituksissa saattaa olla käytettävissä pari lainaläppäriä

# Tehtävät ja arvostelu

- Kurssin arvostelu perustuu ohjelmointitehtäviin
- Uusi kierros alkaa aina ma klo 8.00 ja on auki seuraavan viikon su klo 23:59 asti
- Kannattaa katsoa kierroksen oppimateriaali läpi jo ennen tehtävien aloittamista!
  
- Jokainen kierros sisältää kahdentyyppisiä tehtäviä: A- ja B-tehtävät
  - A-tehtävät ovat perustason tehtäviä
  - B-tehtävät ovat syventäviä tehtäviä
  
- Kurssin arvostelussa sekä A-tehtävien että B-tehtävien painoarvo on 50%
- Esimerkkejä siitä, miten tietyn arvosanan voi saavuttaa:
  - **1:** vähintään 50% A-tehtävien pisteistä (ja vähintään 25% jokaiselta kierrokselta)
  - **2:** vähintään 75% A-tehtävien pisteistä ja noin 25% B-tehtävien pisteistä
  - **3:** vähintään 80% A-tehtävien pisteistä ja noin 50% B-tehtävien pisteistä
  - **4:** vähintään 80% A-tehtävien pisteistä ja noin 80% B-tehtävien pisteistä
  - **5:** vähintään 90% A-tehtävien pisteistä ja noin 90% B-tehtävien pisteistä
- Lopulliset pisterajat saattavat joustaa alaspäin

# Ajankäytöstä

- 5 op = 5 x 27 h = 135 h
- Kontaktiopetus (0–26 h):
  - Aloitusluento (2 h)
  - 24 h ohjattuja tietokoneharjoituksia (12 x 2 h)
  - *Kurssin saa suorittaa myös etänä ilman osallistumista kontaktiopetukseen*
- Oma työskentely (109–135 h):
  - Harjoitustehtävät 60 h (6 x 10 h)
  - Kurssin verkkomateriaaleihin perustuva omatoiminen opiskelu 49-75 h
- Kurssin läpäiseminen vaatii aktiivista omatoimista työskentelyä
  - ***Ohjelmoimaan oppii vain ohjelmoimalla!***
  - Aloita ohjelmointitehtävien tekeminen aina ajoissa
- Ohjelmointitehtävien vaativuus ja kurssin vaatima ajankäyttö kasvaa loppua kohti!



# Ohjelmointitehtävistä

- Ohjelmointitehtävät palautetaan yksilösuorituksina
- CodeRunner-ohjelma pisteyttää ohjelmointitehtävien palautukset automaattisesti
- Tehtävistä on mahdollista keskustella opettajien kanssa ohjatuissa harjoituksissa
- Kurssilla on myös Slack-työtila. Ohjeet ks. dia 23.
- Näin isolla kurssilla opettajilla ei ole mahdollisuutta vastata yksittäisiin sähköpostikysymyksiin tehtävistä.
  - Jos kuitenkin löydät mielestäsi virheen tehtävästä, ole yhteydessä opettajiin (Huom! Antti on lomalla heinäkuun ajan)
- Tehtävistä saa keskustella myös kaverin kanssa, varsinkin harjoituksissa
  - Valmiita ratkaisuja **ei saa jakaa** (miksi veisit kaverilta oivaltamisen ilon?)
  - Kurssilla on käytössä automaattinen vastausten samankaltaisuuden arviointi ("ohjelmakoodin Turnitin"), josta lisää seuraavalla sivulla

# Palautusten samankaltaisuuden arviointi

- Ohjelmointitehtävät palautetaan **yksilösuorituksina**
- Kaikki kurssin tehtäväpalautukset tallentuvat tietokantaan
- Opiskelijoiden palautuksia vertaillaan toisiinsa automaattisesti jokaisella kierroksella
  - Palautuksia vertaillaan myös vuosien 2017-2018 palautuksiin
- Yksinkertaisissa tehtävissä ratkaisut ovat luonnollisesti usein samankaltaisia
- Laajemmissa tehtävissä samankaltaisuuksia ei pitäisi esiintyä, koska meillä kaikilla on hieman erilainen tapa ratkaista ohjelmointiongelmia
- Jos automaattinen tarkistus löytää palautuksista samankaltaisuuksia:
  - Käymme palautukset tarkemmin läpi ja selvitämme tilanteen opiskelijoiden kanssa

## **OHJELMOINTIKURSSIN HUONEENTAULU**

**Kun kaverisi on jo ratkaissut ohjelmointitehtävän, jota itse olet vielä tekemässä:**

**ON SALLITTUA, että näytät koodiasi kaverille, joka neuvoo sinua ongelmatilanteessa  
EI OLE SALLITTUA katsoa kaverin valmista ratkaisua**

# Osa 2:

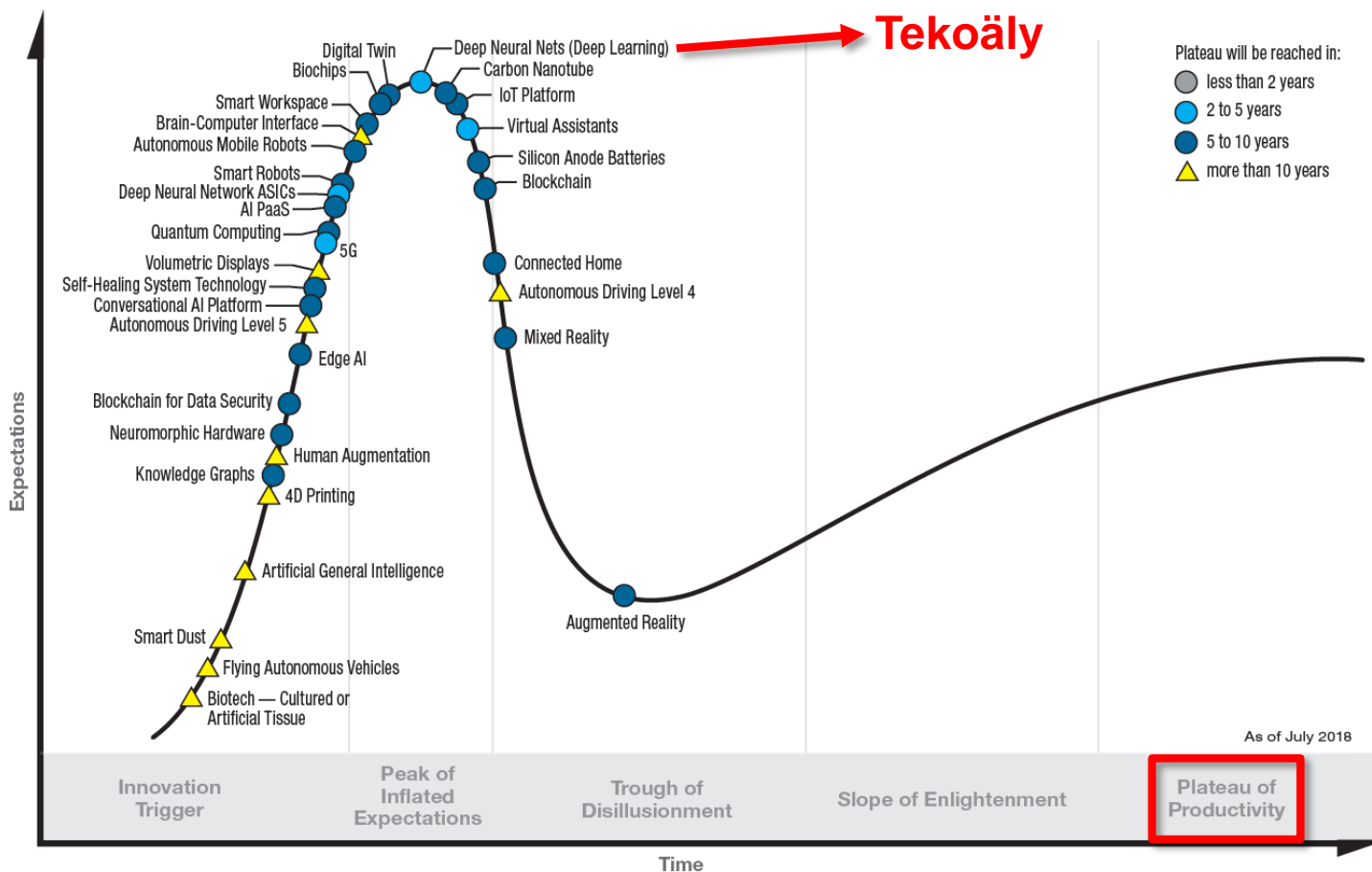
## Yleistä asiaa ohjelmoinnista



# Ohjelmointitaitojen merkitys

- Tyypillinen uutinen 2018: yhä useammat työtehtävät tullaan automatisoimaan
  - Ohjelmointitaito yhä tärkeämpää, mutta mediassa myös liioittelua

## Hype Cycle for Emerging Technologies, 2018

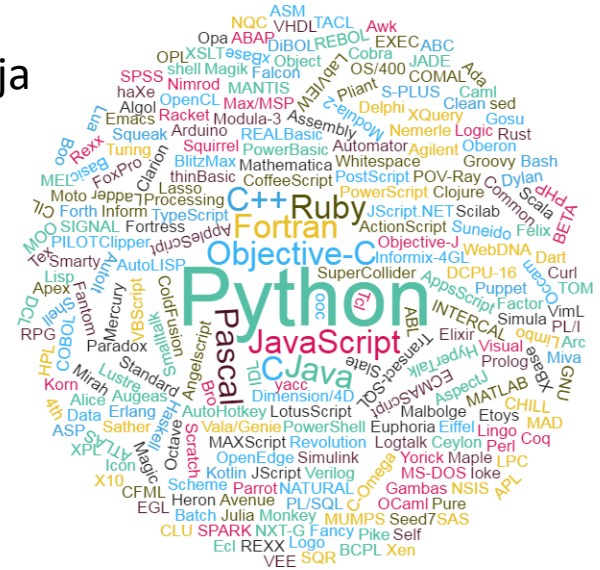


# Ohjelmointi ja kemian tekniikka

- **Kemian tekniikassa** ohjelmoinnilla on merkittävä rooli. Esimerkkejä:
  - Kemian teollisuuden prosessien **ohjaus**
  - Prosessien simulointi (**mallinnus**): kemiallinen prosessi mallinnetaan ensin tietokoneella ja siirrytään vasta sitten kalliimpaan kokeelliseen työhön
  - Erilaisten kemiallisten **tietokantojen** tehokas käyttö vaatii ohjelmointia: materiaali- ja kemikaalitietokannat, rakenne-ominaisuustietokannat
  - **Mittausdatan** edistynyt käsittely vaatii usein ohjelmointia (tai ainakin se on huomattavasti tehokkaampaa ohjelmoimalla kuin käsityönä Excelillä)
  - *Keksitkö lisää esimerkkejä?*
- **Kemiallisissa tieteissä** ohjelmoinnilla keskeinen rooli monilla osa-alueilla
  - Laskennallinen kemia, mittausdatan käsittely, spektroskopia, tiedonlouhinta (*data mining*), koneoppiminen (*machine learning*), ...

# Mitä ohjelmointi on?

- Ohjelmoinnissa luodaan tarkka toimintaohje, eli ”ohjelma”
  - Ohjelman käskyjen avulla tietokone suorittaa tarvittavan tehtävän
- Ohjelma täytyy kirjoittaa **ohjelmointikielellä**, joita on satoja
  - Alkuaikoina ohjelmat kirjoitettiin keskusprosessorin konekielellä (esim. bittijonoina 011001...)
  - Nykyisin käytetään pääasiassa korkeamman tason lausekieliä (kuten Python, Java, C++)
- **Minkä ohjelmointikielen opettelen?**
  - Mikä tahansa moderni lausekieli on hyvä valinta
  - Tärkeintä on oppia ohjelmoinnissa tarvittava **ajattelutapa**.
  - Ajattelutavan opittuasi uusien ohjelmointikielten oppiminen on varsin helppoa
- Tällä kurssilla käytetään **Python**-ohjelmointikieltä



Kuva: AJK/wordclouds.com

# Python-ohjelmointikielen etuja

- Selkeä "kielioppi" (*syntax*). Pythonilla kirjoitettu ohjelma, "koodi", on siis varsin luettavaa, ainakin jos hyviä ohjelmointikäytäntöjä on noudatettu
- Erittäin matala aloituskynnys. Aloittelijakin saa heti aikaan näkyviä tuloksia
- Python-tulkki itsessään on **vapaa** (*free*) ohjelmisto, eli sen lähdekoodi on kaikkien saatavilla. Tämän seurauksena sen voi hankkia ilmaiseksi ja kuka tahansa voi ehdottaa siihen parannuksia.
- Pythonille on saatavissa valtava määrä vapaita **kirjastoja** (*library*) joita voi hyödyntää omissa ohjelmissa
- Erittäin vahva ja aktiivinen [yhteisö](#) kehittää Pythonia koko ajan



# Mihin Python soveltuu?

- Python soveltuu sekä pieniin että suuriin ohjelmointiprojekteihin
- Pythonilla voi toteuttaa myös suurta laskentatehoa vaativia tieteellisiä ohjelmistoja
  - Tällöin raskas laskenta on syytä hoitaa tieteellisten Python-kirjastojen avulla
  - Esim. NumPy ja SciPy, joita hyödynnetään tällä kurssilla
- Python on ns. **tulkattu** kieli
  - Tulkki (*interpreter*) muuntaa lausekielisen ohjelman ajon aikana konekielelle
  - Tulkatun kielen vastakohta on **käännetty** kieli (kuten C tai C++)





# Tietotekniikka sivuaineena?

- Tietotekniikka on erinomainen sivuaine kemian tekniikan opiskelijalle
  - <https://into.aalto.fi/pages/viewpage.action?pageId=4854511>
- CHEM-A2600 vastaa sisällöltään suurelta osin kurssia Ohjelmoinnin peruskurssi Y1 (CS-A1111)
- Seuraava kurssi tietotekniikan sivuaineen “aloittelijan opintopolulla” on Ohjelmoinnin peruskurssi Y2 (CS-A1121)
  - Y2-kurssilla on esitietotesti, jolla varmistetaan Y1-kurssin tasoinen osaaminen
  - CHEM-A2600 –kurssilta täytyy olla vähintään arvosana **3**, jotta Y2-kurssille jatkamista voi harkita
- Jos haluat suorittaa tietotekniikan sivuaineen ja korvata Y1-kurssin kurssilla CHEM-A2600, ole yhteydessä ensin Anttiin niin selvitetään asia
- Kurssien Y1 ja Y2 vastuuoopettajat 2018-2020:
  - Kerttu Pollari-Malmi (Y1)
  - Vesa Hirvisalo (Y2)

# Osa 3:

## Kurssilla käytettävät työkalut

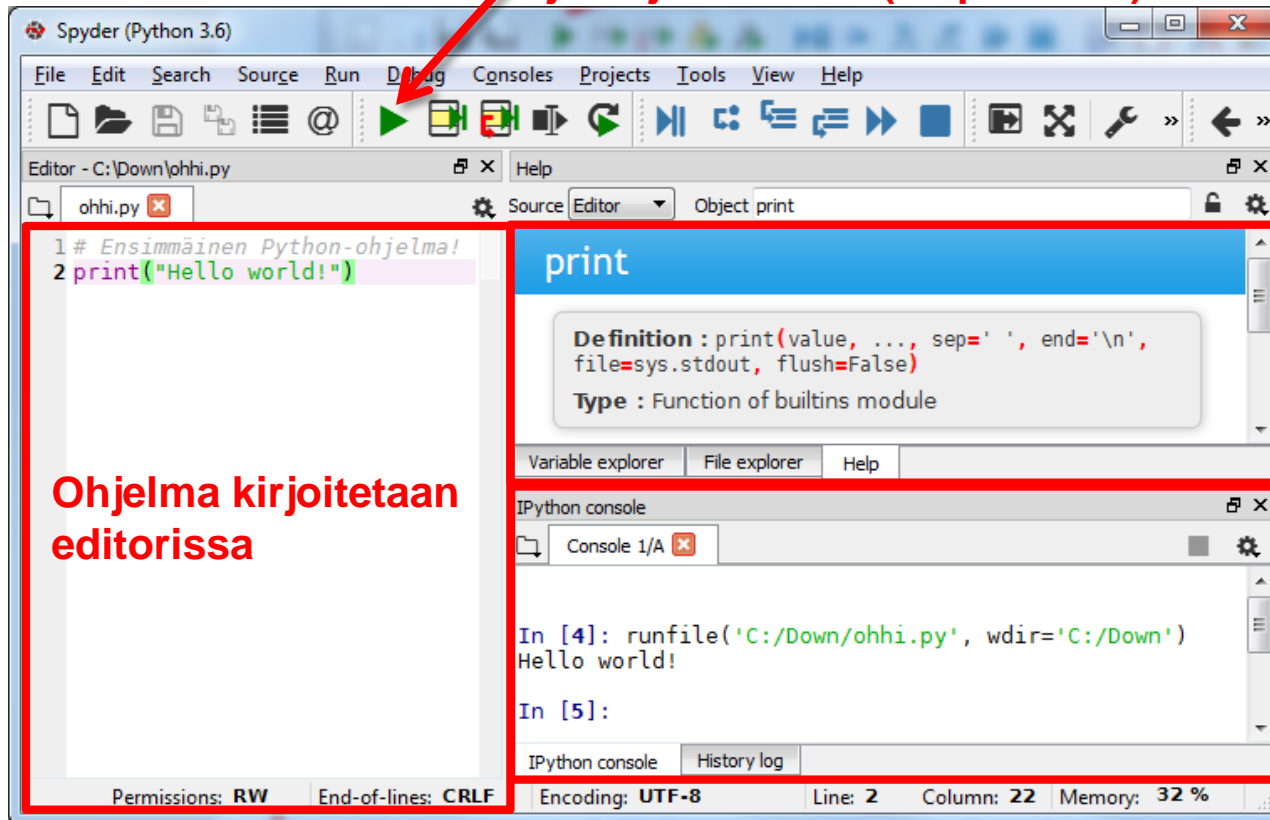


# Anaconda

- Kurssilla hyödynnämme **Anaconda**-nimistä Python-jakelupakettia (**distribution**)
- Paketti on asennettuna kaikkiin CHEM-rakennuksen mikroluokkiin
- Jakelupaketti sisältää
  - Python-**tulkin** ohjelmien ajamista varten
  - Valtavan määrän **kirjastoja** joita voi hyödyntää omissa ohjelmissa
  - Erityisen kattavan kirjastokokoelman **tieteellistä ohjelmointia** varten
  - Graafisen **Spyder**-ohjelmointiympäristön, joka helpottaa ohjelmien luomista
- Anacondan perusversio on ilmainen (Windows-, Linux-, OSX)
  - Paketin voi asentaa helposti myös **omalle tietokoneelle**
  - <https://www.anaconda.com/download/>
  - Asennusohjeita oppimateriaalissa:
  - <https://mycourses.aalto.fi/mod/book/view.php?id=442395&chapterid=2492>
  - Tällä kurssilla kelpaavat Pythonin versiot **3.7** ja **3.6**

# Spyder-ohjelmointiympäristö (*IDE*)

Aja ohjelma tästä (tai paina F5)



Ohjelma kirjoitetaan editorissa

Help: paina editorissa Ctrl+I funktion nimen päällä ja Spyder näyttää ohjeen

Ohjelma pyörii "konsolissa"  
Voit myös kirjoittaa komentoja suoraan konsoliin

- **Huom!** Spyder käynnistyy aika hitaasti, joten odottele rauhassa (jopa minuutti)
- Kun ajat ohjelman, Spyder tallentaa sen ennen ajamista (*nimi.py*)
- Kannattaa koota jokaisen kierroksen materiaalit omaan alihakemistoonsa

# Oppimateriaali

- MyCourses-sivulta löytyy kurssin [Python-oppimateriaali](#)
  - Toteutettu MyCourses-kirjana. Myös PDF saatavilla kurssin etusivulla.
  - Sisältää kaikki kurssilla käsiteltävät asiat
  - Linkkejä lisämateriaaliin löytyy tämän esityksen lopusta

## Python-oppimateriaali (CHEM-A2600)



Next: [1.1. Tulostaminen \(print\) ja syötteen lukeminen \(input\)](#) ►

### Kierros 1

Kurssin ensimmäisellä kierroksella tutustutaan ohjelmoinnin peruskäsitteisiin ja Python-ohjelmointikielen perusteisiin.

### Oppaan lukuohje

Kun oppaassa esitetään Python-koodia, se näyttää tältä:

```
print("Nyt lasketaan!")  
print("11*11 on", 11*11)
```

Kun oppaassa näytetään, mitä Python-koodi tulostaa, se näyttää tältä:

```
Nyt lasketaan!  
11 * 11 on 121
```

### TABLE OF CONTENTS

- 1. Kierros 1
  - 1.1. Tulostaminen (print) ja syötteen lukeminen (input)
  - 1.2. Muuttujat
  - 1.3. Tyypimuunnokset
  - 1.4. Kokonaisluvut, liukuluvut ja pyöristäminen
  - 1.5. Matemaattiset perusoperaattorit
  - 1.6. if-elif-else ja vertailuoperaattorit
  - 1.7. Totuusmuuttujat
  - 1.8. Loogiset operaattorit
  - 1.9. Laskujärjestyksestä
  - 1.10. while-silmukka
  - 1.11. for-silmukka
- 2. Kierros 2
  - 2.1. Funktiot
  - 2.2. Erilaisia funktioita
  - 2.3. Muotoiltu tulostaminen

# Coderunner-tehtävät (MyCourses)

**Voit halutessasi harjoitella CodeRunnerin käyttöä ensin testitehtävien avulla**

## *CodeRunner-testitehtäviä*

Näiden testitehtävien avulla voit harjoitella CodeRunner-ohjelmointitehtävien tekemistä. Näitä tehtäviä ei arvioida.

## Kierros 1: Perusteet, muuttujat, kontrollirakenteet

[Linkki kierroksen oppimateriaaliin](#)

### *Kierros 1: Harjoitusryhmän valinta*

#### *Kierros 1, A-tehtävät*

#### *Kierros 1, B-tehtävät*

**Vastattuasi kaikkiin Quiz-sivun tehtäviin klikkaa alalaidasta**

Finish attempt ...

Question 1 Edit question Flag question Marked out of 1.00 Not complete

Kirjoita funktio *ainemaara*, joka ottaa parametreinä yhdisteen massan (g) ja moolimassan (g/mol) ja palauttaa yhdisteen ainemäärän (mol). Jos funktiota kutsutaan epäfysikaalisella parametrilla, sen pitää palauttaa arvo -1.

Vinkki:  $n = \frac{m}{M}$

Answer: (penalty regime: 0, 0, ... %)

1
---

**Kopioi Spyderissä kirjoittamasi ja testaamasi koodi tähän ja klikkaa *Check* (tai *Tarkista*)**

**1. kierroksen tehtävissä virheellisistä palautuksista ei tule miinusta. Tämä muuttuu myöhemmin.**

Check

# Slack-työtila

- Kurssille on luotu oma Slack-työtila **chempython.slack.com**
- Slackiä voi käyttää monella eri tavalla:
  - Mobiilisovelluksella (ilmainen, löytyy puhelimen sovelluskaupasta)
  - Työpöytäsovelluksella (<https://slack.com/downloads/>)
  - Web-selaimella (<https://chempython.slack.com>)
- Kirjaudu kurssin Slack-työtilaan @aalto.fi –sähköpostiosoitteellasi
- **Slack-keskustelujen säännöt**
  1. Käytä omaa etu- ja sukunimeäsi, ei nimimerkkiä.
  2. Kesällä 2019 kurssin keskustelut käydään kanavalla **#kesa2019**
  3. Keskusteluetiketistä: kyseessä on kurssin julkinen foorumi samaan tapaan kuin luentosali tai luokkahuone, joten pidetään keskustelu asiallisena.
  4. Kysymysten esittäminen ohjelmointitehtävistä on sallittua, mutta kokonaisten ratkaisujen tai ratkaisuyritysten lähettäminen kanavalle ei ole sallittua.
  5. Kesällä 2019 Slack on kurssin virallinen ohjausmuoto (harjoitusten lisäksi). Huomaa kuitenkin, ettei Slack-päivystys ole ympärivuorokautinen, vaan keskittyy suurin piirtein aikoihin ma-pe klo 8-16.

# Tärkeitä Python-resursseja

- <https://docs.python.org/3/>
  - Pythonin virallinen dokumentaatio
  - <https://docs.python.org/3/tutorial/index.html> (tutorial)
  - <https://docs.python.org/3/reference/index.html> ("kielioppi")
- <https://www.tutorialspoint.com/python3/>
  - Varsin selkeä tutoriaali
- Google
  - Valtavan käyttäjämäärän ansiosta netti on täynnä koodikysymyksiä ja vastauksia niihin (esim. <http://stackoverflow.com/>)