



Aalto-yliopisto  
Kemian tekniikan  
korkeakoulu

# Kemiantekniikan ohjelmointikurssi CHEM-A2600 (5 op)

Antti Karttunen

Syksyn 2023 aloitusluento 2023-09-04

Alkukysely: [premo.aalto.fi/a2600](https://premo.aalto.fi/a2600)

# Osa 1: Kurssin kuvaus



# Kurssin opettajat (syksy 2023)



Antti Karttunen



Lydia Grönholm



Viktoria Malikova



Pinja Salminen

# Syksyn 2023 kurssin yhteenveto

- Syksyllä 2023 kurssi järjestetään lähiopetuksena
  - Kurssilla on ohjattuja harjoituksia Kemistintie 1 B-siiven tietokonehuoneissa.
  - Ohjausta on lisäksi mahdollista saada myös kurssin Zulip-työtilassa.
- Kurssilla on kuusi **kierrosta**
  - 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
  - Tehtäviä voi tehdä omalla tietokoneella missä tahansa ja milloin tahansa

# Kurssikalenteri

**Kurssin aloitusluento maanantaina 4.9. klo 9.15–10.00 (Sali Ke1)**

Kierros	Sulkeutuu (klo 17.00)	Teema
1	Ma 18.9.	Pythonin perusteet. Tulostaminen, syötteen lukeminen, muuttujat, operaattorit ja kontrollirakenteet (if-elif-else, while, for)
2	Pe 29.9.	Funktiot, moduulit
3	Pe 13.10.	Tietorakenteet (listat, monikot, sanakirjat)
4	Pe 27.10.	Numpy ja Matplotlib
5	Pe 10.11.	Tiedostojen käsittely, virheenkäsittely
6	Pe 24.11.	Sovellukset, Scipy, olio-ohjelmoinnin perusteita

**Kaikki tehtäväpaketit avautuvat 30.9. mennessä ja kurssin voi suorittaa nopeammassa tahdissa kuin yllä.**

# Ohjatut harjoitukset

- Kurssilla on tarjolla ohjattuja harjoituksia 4.9.–24.11. seuraavasti:
  - Ma klo 15.15–17.00
  - Ti klo 13.15–15.00
  - To klo 13.15–15.00 (Periodi I) / 10.15–12.00 (Periodi II)
  - Pe klo 8.30–10.00
- Kaikki harjoitukset ovat tietokonehuokassa B016 (Kemistintie 1).
- Harjoitukseen voi liittyä milloin vain, eli kyseessä on siis "päivystystyyppinen" tilaisuus.
- Zulip-työtilan ohjeet: MyCourses -> Zulip-työtila
- Zulip-ohjaus painottuu ylläoleviin harjoitusaikoihin. Kysymyksiä voi esittää Zulipissa myös harjoitusaikojen ulkopuolella, mutta vastaus ei välttämättä tule välittömästi

# Tehtävät ja arvostelu

- Kurssin arvostelu perustuu ohjelmointitehtäviin
- 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%
- Arvosteluesimerkkejä alla olevassa taulukossa
  - **Painotettu% = 0.50 × A% + 0.50 × B%**
  - A% on A-tehtävien suoritusprosentti, eli A\_pisteet / max\_A\_pisteet
  - B% on B-tehtävien suoritusprosentti, eli B\_pisteet / max\_B\_pisteet

Painotettu%	Arvosana	Esimerkkejä siitä, miten arvosanan voi saavuttaa
40%	1	Esim. 80% A-pisteistä ja 0% B-pisteistä tai 70% A-pisteistä ja 10% B-pisteistä
50%	2	Esim. 80% A-pisteistä ja 20% B-pisteistä tai 90% A-pisteistä ja 10% B-pisteistä
65%	3	Esim. 80% A-pisteistä ja 50% B-pisteistä tai 100% A-pisteistä ja 30% B-pisteistä
80%	4	Esim. 90% A-pisteistä ja 70% B-pisteistä tai 100% A-pisteistä ja 60% B-pisteistä
90%	5	Esim. 90% A-pisteistä ja 90% B-pisteistä tai 100% A-pisteistä ja 80% B-pisteistä

# Ajankäytöstä

- 5 op = 5 x 27 h = 135 h
- Ohjattu opetus (0–30 h):
  - Aloitusluento (1 h)
  - Ohjatut harjoitukset (saa käydä vaikka kaikissa, jos siltä tuntuu)
- Oma työskentely (105–135 h):
  - Harjoitustehtävät noin 60 h (6 x 10 h)
  - Kurssin verkkomateriaaleihin perustuva omatoiminen opiskelu
- Kurssin läpäiseminen vaatii aktiivista omatoimista työskentelyä
  - ***Ohjelmoimaan oppii vain ohjelmoimalla!***
  - Aloita ohjelmointitehtävien tekeminen aina ajoissa
- Ohjelmointitehtävien vaativuus ja kurssin vaatima ajankäyttö kasvaa loppua kohti!
  - Aiempien vuosien perusteella vaativimmat kierrokset ovat kierrokset **3** ja **5**.



# Ohjelmointitehtävistä

- Ohjelmointitehtävät palautetaan yksilösuorituksina
- CodeRunner-ohjelma pisteyttää ohjelmointitehtävien palautukset automaattisesti
- Tehtävistä on mahdollista keskustella opettajien kanssa ohjatuissa harjoituksissa ja Zulip-työtilassa
- Näin isolla kurssilla opettajilla ei ole mahdollisuutta vastata yksittäisiin sähköpostikysymyksiin tehtävistä.
  - Jos kuitenkin löydät mielestäsi virheen tehtävästä, voit laittaa sähköpostia [antti.karttunen@aalto.fi](mailto:antti.karttunen@aalto.fi)
- Tehtävistä saa keskustella myös kaverin kanssa
  - Valmiita ratkaisuja **ei saa jakaa** (miksi veisit kaverilta oivaltamisen ilon?)
  - Kurssilla on käytössä automaattinen vastausten samankaltaisuuden arviointi, 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 **aiempien vuosien 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, että katsot kaverin valmista ratkaisua.**

# Tekoälyn käytöstä

- Kehittyneimmät tekoälyjärjestelmät kuten ChatGPT tuottavat varsin sujuvasti lyhyitä Python-koodin pätkiä, kuten yksittäisiä funktioita.
  - Laajempia ohjelmia ne eivät osaa kirjoittaa ja kemiallisten ongelmien kanssa ne ovat yleensä pulassa.
- Koska tämä on ohjelmoinnin peruskurssi, ChatGPT osaa tuottaa kohtuullisia ratkaisuja varsinkin kurssin helpompiin tehtäviin.
  - Edistyneemmissä tehtävissä se on useimmiten pulassa.
- Tämän kurssin tärkein osaamistavoite:
  - Kurssin suoritettuaan **opiskelija osaa kirjoittaa** pieniä ohjelmia kemiantekniikan sovelluksiin, kuten mittausdatan käsittelyyn, analysointiin ja visualisointiin
- Kurssin osaamistavoitteiden näkökulmasta tekoälyn käyttö tällä kurssilla ei ole mielekästä: tarkoitus on oppia ohjelmoinnin perusteet **ohjelmoimalla itse**.
  - Tekoälyjärjestelmän suoraan tuottamaa ohjelmakoodia ei saa esittää opiskelijan itse kirjoittamana vastauksena kurssin tehtäviin.
  - [Aalto-yliopiston ohjeistus tekoälyn käytöstä opetuksessa ja oppimisessa](#).

# Kuva MyCourses-profiiliin

- Opettajien kannalta on mukavaa, jos nimet yhdistyvät kasvoihin
- Olisi hienoa, jos voisit lisätä profiiliisi kasvokuvan!

**MyCourses, oikea yläkulma**

The image shows a user profile page for Antti Karttunen. On the left, a navigation menu includes 'Profile', 'Personal preferences', 'Calendar', and 'Grades'. The main profile area shows the user's name, a 'Message' button, and an 'Edit profile' button. Below this, the 'User details' section lists the email address 'antti.karttunen@aalto.fi'. On the right, the 'User picture' section shows the 'Current picture' and a 'Delete picture' option. Below that is the 'New picture' section, which includes a 'Files' button and a large red-bordered box containing the text 'Vedä kuva tähän' (Drag picture here) and 'You can drag and drop files here'. A red arrow points from the 'Edit profile' button to the 'User picture' section. Another red arrow points from the 'User picture' section to the 'New picture' section. A third red arrow points from the 'New picture' section to the 'Update profile' button at the bottom right.

# Osa 2:

## Yleistä asiaa ohjelmoinnista

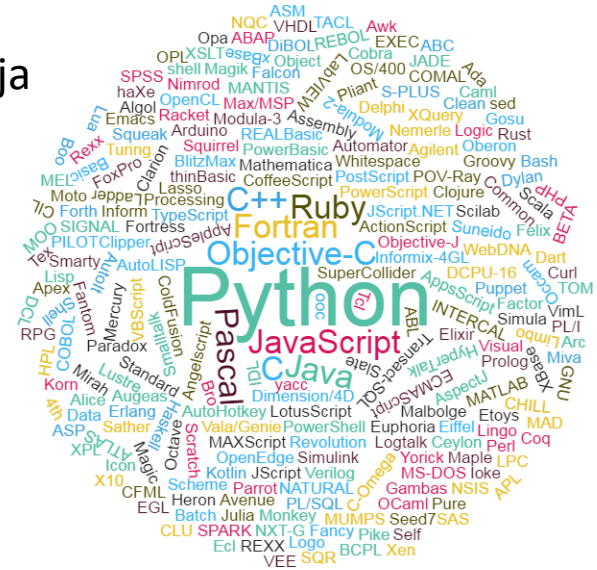


# Ohjelmointi ja kemian tekniikka

- **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 keskusprossessorin 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://www.aalto.fi/fi/sivuaineet/tietotekniikka-sivuaine>
- CHEM-A2600 vastaa sisällöltään suurelta osin kurssia Ohjelmoinnin peruskurssi Y1 (CS-A1111)
- Seuraava kurssi tietotekniikan sivuaineen “aloittelijan opintopolulla” on Ohjelmoinnin peruskurssi Y2 (CS-A1121)
  - Kurssin CHEM-A2600 suorittaneet voivat aloittaa sivuaineen suoraan kurssista Ohjelmoinnin peruskurssi Y2.
  - Y2-kurssilla on esitietotesti, jolla varmistetaan Y1-kurssin tasoinen osaaminen
  - CHEM-A2600 –kurssilta täytyy olla vähintään arvosana **3**, jotta Y2-kurssille jatkamista voi harkita

# Osa 3:

## Kurssilla käytettävät työkalut



# Oppimateriaali

- Kurssin Python-oppimateriaali on toteutettu MyCourses-kirjana
- Oppimateriaali sisältää linkkejä myös muihin Python-oppimateriaaleihin
- Oppimateriaali sisältää paljon pieniä **harjoittelutehtäviä**. Näitä ei pisteytetä, mutta ne tukevat varsinaisten ohjelmointitehtävien tekemistä. Kannattaa kokeilla!

Esimerkki  
oppimateriaalin  
harjoittelutehtävästä:

## Tehtävä 1.6.1

Täydennä if-elif-else -lause vetämällä sanat oikeille paikoilleen

```
paine = float(input("Anna renkaan paine (bar):\n"))
```

```
if paine <= 0.0:  
    print("Virheellinen paine")
```

```
    paine <= 5.0:  
        print("Turvallinen paine")
```

```
    paine <= 7.0:  
        print("Rajoilla ollaan")
```

```
    :  
        print("Rengas räjähti")
```

elif else elif

✓ Check

# Anaconda ja Spyder

- Kurssilla hyödynnämme **Anaconda**-nimistä Python-jakelupakettia (**distribution**)
- Anacondan perusversio on ilmainen (Windows, Mac, Linux)
  - Asennusohjeet löytyvät oppimateriaalista
  - Anaconda on esiasennettu Kemistintie 1 tietokonealuokkien koneisiin, joita voi tarvittaessa myös etäkäyttää [vdi.aalto.fi](http://vdi.aalto.fi) –palvelun avulla.
- Anaconda-jakelupaketti sisältää
  - Python-**tulkin** ohjelmien ajamista varten
  - Valtavan määrän **kirjastoja** joita voi hyödyntää omissa ohjelmissa
  - Eriyisen kattavan kirjastokokoelman **tieteellistä ohjelmointia** varten
  - **Spyder**-ohjelmointiympäristön, jota suosittelemme ohjelmien kirjoittamiseen ja testaamiseen tällä kurssilla
  - Oppimateriaali sisältää lyhyet Spyderin käyttöohjeet
- Kurssilla **saa käyttää** myös muita ohjelmointiympäristöjä kuten PyCharmia
  - Pythonin versio pitää olla vähintään **3.8**

# Coderunner-tehtävät

- Voit halutessasi harjoitella CodeRunnerin käyttöä testitehtävien avulla (MyCourses)

Question 1

Flag question Marked out of 1.00 Not complete

## 1. Tehtävänanto

Kopioi seuraavat kaksi riviä alla olevaan tekstiruutuun ja paina *Tarkista/Kontrollera/Check* niin näet, miten CodeRunner-tarkastusohjelma toimii.

```
print("Moi!")  
print("Hei!")
```

Answer: (penalty regime: 0 %)

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

**3. Kopioi Spyderissä kirjoittamasi ja testaamasi koodi tähän tekstiruutuun ja klikkaa *Tarkista / Kontrollera / Check***

Check

**4. Vastattuasi kaikkiin tehtäviin klikkaa alalaidasta *Lopeta tehtävä / Avsluta test... / Finish Attempt...* (tämä ei ole pakollista, tehtävät palautuvat automaattisesti kun aikaraja umpeutuu)**