



Aalto-yliopisto
Kemian tekniikan
korkeakoulu

Kemiantekniikan ohjelmointikurssi CHEM-A2600 (5 op)

Antti Karttunen

Kevään 2024 aloitusluento 2024-04-22

Alkukysely: premo.aalto.fi/a2600

Osa 1:

Kurssin kuvaus



Kurssin opettajat (kevät 2024)



Antti Karttunen



Jonna Järvelin

Kevään 2024 kurssin yhteenveto

- Keväällä 2024 kurssi järjestetään sekä lähi- että etäopetuksena.
 - Ohjausta on tarjolla kurssin Zulip-työtilassa.
 - Kurssilla on myös ohjattuja harjoituksia Kemistintie 1 B tietokoneluokissa.
- 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 22.4. klo 10.15–11.00

Kierros	Sulkeutuu (klo 17.00)	Teema
1	Ma 6.5.	Pythonin perusteet. Tulostaminen, syötteen lukeminen, muuttujat, operaattorit ja kontrollirakenteet (if-elif-else, while, for)
2	To 16.5.	Funktiot, moduulit
3	Ke 29.5.	Tietorakenteet (listat, monikot, sanakirjat)
4	Pe 7.6.	Numpy ja Matplotlib
5	Ke 19.6.	Tiedostojen käsittely, virheenkäsittely
6	Pe 28.6.	Sovellukset, Scipy, olio-ohjelmoinnin perusteita

**Kaikki tehtäväpaketit avautuvat 20.5. mennessä ja
kurssin voi suorittaa nopeammassa tahdissa kuin yllä.**

Ohjatut harjoitukset ja Zulip

- Ensimmäiset lähiharjoitukset **tiistaina 23.4. klo 13.00–14:30** (tietokoneluokka B018)
Muut harjoitusajat MyCourses-työtilan etusivulla.
 - Harjoitukseen voi tulla milloin vain, eli kyseessä on siis "päivystystyyppinen" tilaisuus.
 - Ohjausta esimerkiksi kurssilla tarvittavien ohjelmistojen asennukseen ja ohjelmointitehtävien tekemiseen.
- Kurssilla on käytössä Zulip-työtila, jossa on tarjolla ohjausta koko kurssin ajan
 - Ohjeet MyCourses -> Zulip-työtila.
 - Kysymyksiä voi lähettää milloin tahansa, mutta päivystys keskittyy arkipäiviin klo 9–17.
 - Tehtäväkierrosten deadline-päivät ovat yleensä ruuhkaisia, joten on kysy neuvoja hyvissä ajoin ennen deadline-päivää!

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% = $0.50 \times A\% + 0.50 \times B\%$**
 - 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 60% A-pisteistä ja 20% 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
- Tehtävistä saa keskustella myös kaverin kanssa
 - Valmiita ratkaisuja **ei saa jakaa** (miksi voisit 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 composite of three screenshots from the MyCourses interface, connected by red arrows to illustrate the steps to upload a profile picture.

- Top-left screenshot:** Shows the top navigation bar with a search icon, a bell icon, a chat icon, and a user profile icon. A red arrow points from the text "MyCourses, oikea yläkulma" to the user profile icon.
- Bottom-left screenshot:** Shows the user profile page for "Antti Karttunen". The left sidebar contains a menu with "Profile", "Personal preferences", "Calendar", and "Grades". A red arrow points from the "Profile" menu item to the "Edit profile" button.
- Right screenshot:** Shows the "Edit profile" form. At the top, there is a section titled "User picture" with a dropdown arrow. Below it, there is a "Current picture" section with a "Delete picture" button. A red arrow points from the "User picture" dropdown to the "Current picture" section. Below that, there is a "New picture" section with a "Files" button. A red arrow points from the "Files" button to a red-bordered box containing the text "Vedä kuva tähän" (Drag picture here) and a download icon. Below this box is an "Update profile" button.

Osa 2:

Yleistä asiaa ohjelmoinnista

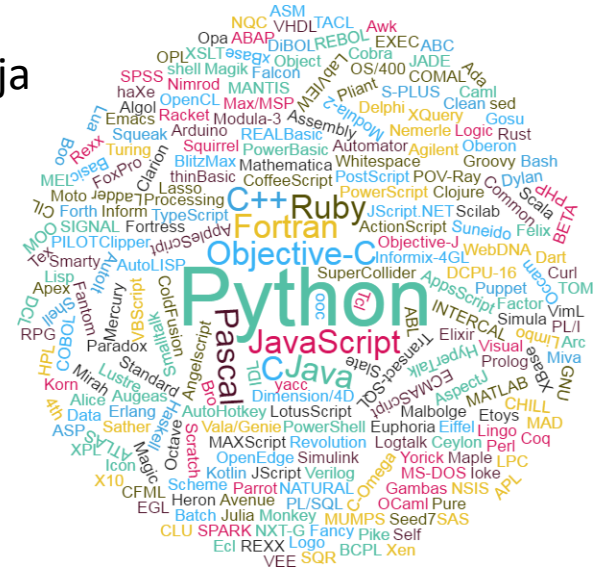


Ohjelmointi ja kemian tekniikka

- **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 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://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 tietokone luokkien koneisiin, joita voi tarvittaessa myös etäkäyttää vdi.aalto.fi –palvelun avulla.
- Anaconda-jakelupaketti sisältää
 - Python-**tulkin** ohjelmien ajamista varten
 - Valtavan määrän **kirjastoja** joita voi hyödyntää omissa ohjelmissa
 - Erityisen 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)