

# forelesning1

October 13, 2025

## 1 Forelesning 1: Programmering i kjemi

### 1.1 Periodesystemet og periodiske egenskaper

```
[ ]: # Installere biblioteker i Jupyter Notebook
     #!pip install mendeleev chemlib pandas seaborn
```

```
[ ]: # Klasser og objekter
     liste = [] # Objekt av klassen "list"
     liste.append(1) # Funksjoner som brukes på objekter, kalles "metoder"
```

```
[ ]: from mendeleev import element

     Z = 10
     grunnstoff = element(Z)

     navn = grunnstoff.name
     symbol = grunnstoff.symbol
     gruppe = grunnstoff.group_id

     print("Atomnummer:", Z, "--Navn:", navn, "--Symbol:", symbol, "--Gruppe:",
           ↪gruppe)
```

- Oppgave: Modifiser programmet slik at det skriver ut informasjonen (navn, symbol og elektronegativitet) om de 18 letteste grunnstoffene.

```
[ ]: for Z in range(1,19):
     grunnstoff = element(Z)
     navn = grunnstoff.name
     symbol = grunnstoff.symbol
     elneg = grunnstoff.electronegativity()
     print(f"Navn: {navn}, Symbol: {symbol}, Elektronegativitet: {elneg}")
```

```
[ ]: import matplotlib.pyplot as plt
     import numpy as np
```

```
[ ]: atomnummer = []
     elektronegativitet = []
```

```
for Z in range(3,11):
    grunnstoff = element(Z)
    atomnummer.append(Z)
    elektronegativitet.append(grunnstoff.electronegativity())
```

- Oppgave: Plott listene mot hverandre. Modifiser programmet slik at det kun plotter grunnstoffene i andre periode. Forklar trenden du ser.

```
[ ]: plt.scatter(atomnummer, elektronegativitet, color = "hotpink")
plt.xlabel("Atomnummer")
plt.ylabel("Elektronegativitet")
plt.show()
```

### 1.1.1 Elektronegativitet i gruppe 1

Oppgave: Plott elektronegativitet som funksjon av atomnummer for grunnstoffene i gruppe 1.

```
[ ]: from mendeleeve import element
import matplotlib.pyplot as plt

atomnummer = []
elektronegativitet = []
gruppe1 = ["H", "Li", "Na", "K", "Rb", "Cs"]
for atom in gruppe1:
    grunnstoff = element(atom)
    atomnummer.append(grunnstoff.atomic_number)
    elektronegativitet.append(grunnstoff.electronegativity())

plt.scatter(atomnummer, elektronegativitet)
plt.xlabel("Atomnummer")
plt.xticks(atomnummer, gruppe1)
plt.ylabel("Elektronegativitet")
plt.show()
```

## 1.2 Støkiometriske beregninger

### 1.2.1 Stoffmengdeberegninger

```
[ ]: from chemlib import Compound

butanol = Compound("C4H9OH") # Definerer forbindelsen
# Regner fra gram til mol og molekyler
print(butanol.get_amounts(grams=2))
# Finner prosentvis masse hydrogen i ammoniakk
print(butanol.percentage_by_mass("H"))

natriumsulfat = Compound("Na2SO4")
```

```
# Fra mol til gram og formelenheter
print(natriumsulfat.get_amounts(moles=1))

ammoniakk = Compound("NH3")
# Fra molekyler til mol og gram
print(ammoniakk.get_amounts(molecules=1E24))
```

```
[ ]: from chemlib import Compound, Reaction
```

```
[ ]: H2 = Compound("H2")
      I2 = Compound("I2")
      HI = Compound("HI")
```

```
[ ]: reaksjon = Reaction([H2, I2], [HI])
      reaksjon.is_balanced
```

```
[ ]: reaksjon.balance()
```

```
[ ]: reaksjon.is_balanced
```

```
[ ]: reaksjon.formula
```

- Oppgave: Bruk chemlib til å balansere ufullstendig forbrenning av benzen (vi får vann og CO)

```
[ ]: from chemlib import Galvanic_Cell

celle = Galvanic_Cell("Cu", "Zn")
egenskaper = celle.properties
print(egenskaper)                # Ulike egenskaper
print(egenskaper["Cell"])        # Cellediagram
print(egenskaper["Anode"])       # Anodemateriale
print(egenskaper["Cathode"])     # Katodemateriale
print(egenskaper["Cell Potential"]) # Cellepotensial
celle.draw()
```

```
[ ]: from chemlib import electrolysis

n_e = 2 # mol elektroner
strømstyrke = 5.0 # A
tid = 2*60*60 # s
e = electrolysis("Cu", n_e, amps = strømstyrke, seconds = tid)
print(e)
```