

Sammendrag

Lese fra fil

Når vi skal lese inn en fil, er det lurt å bruke `with` for å sørge for at fila lukkes hvis det oppstår en feil.

For å åpne en fil bruker vi `open()`. Filer med innhold på norsk bør åpnes med `encoding="utf-8"` eller `encoding="utf-8-sig"`.

Når vi åpner en fil, får vi et filobjekt. Vi kan hente ut alt innholdet med metoden `read()`.

```
filnavn = "MikkelRev.txt"
```



```
with open(filnavn, encoding="utf-8") as fil:
    innhold = fil.read()

print(innhold)
```

Filobjektet oppfører seg også som en beholder, som vi kan gå gjennom linje for linje:

```
filnavn = "MikkelRev.txt"
```



```
with open(filnavn, encoding="utf-8") as fil:
    for linje in fil:
        print(linje)
```

Skrive til fil

Når vi skal skrive til fil, åpner vi den på samme måte som når vi skal lese fra den, men vi legger til `"w"` når vi åpner fila.

For å skrive til fila bruker vi metoden `write()`.

```
with open(filnavn, "w") as fil:  
    fil.write("Hei!")
```



Med "w" (for «write») overskriver vi alt innhold i fila. For å legge til tekst i en eksisterende fil kan vi i stedet bruke "a" (for «append»):

```
with open(filnavn, "a") as fil:  
    fil.write("Hei\n")  
    fil.write("på\n")  
    fil.write("deg.")
```



Da kan det også være nyttig å kjenne til spesialtegn som «\n» (linjeskift, eng. «newline») og «\t» (tabulator).

Nyttige stringmetoder

Metoden `split()` lar oss dele opp en tekst ved et angitt tegn eller en tekst. Vi får en liste med de nye verdiene i retur.

Metodene `lstrip()`, `rstrip()` og `strip()` kan brukes til å fjerne overflødige mellomrom eller linjeskift på henholdsvis venstre side, høyre side og begge sider av en tekst.

CSV-filer

Vi leser CSV-filer på samme måte som andre filer, men vi bruker biblioteket «csv» for å gjøre det enklere å håndtere filformatet.

Metoden `reader()` gir oss «leseobjekt». Vi kan da lese neste linje med metoden `next()`, og vi kan lese alle linjene med en `for...in`-løkke.

```
import csv
```



```
filnavn = "Befolkning_1951-2022.csv"
```

```
with open(filnavn, encoding="utf-8-sig") as fil:  
    filinnhold = csv.reader(fil, delimiter=";")
```

```
    overskrifter = next(filinnhold)  
    print(overskrifter)
```

```
    for rad in filinnhold:  
        print(rad)
```

JSON-filer

Vi leser JSON-filer på samme måte som andre filer, men vi bruker biblioteket «json» for å gjøre det enklere å håndtere filformatet.

Når vi jobber med en JSON-fil, bruker vi metoden `load()` for å hente inn hele fila i form av et datasett organisert i lister og ordbøker.

```
import json
```



```
filnavn = "skandinavia.json"
```

```
with open(filnavn, encoding="utf-8") as fil:  
    data = json.load(fil)
```

```
print(data)
```

Slike datasett er ofte vanskelige å lese for oss mennesker. Det blir lettere om vi får formatert datasettet med linjeskift og innrykk. Det kan vi få til med metoden `dumps()`.

```
data_formatert = json.dumps(data, indent=2)
```



```
print(data_formatert)
```

