1 Tekstfil

I reelle programmeringsprosjekter trenger vi ofte å lese data fra filer, behandle dem på en eller annen måte, og deretter kanskje skrive resultatene til nye filer. Å lære å jobbe med filer i Python er derfor en viktig ferdighet for å kunne løse reelle problemer med store mengder data som vi ikke vil skrive inn manuelt.

1.1 Åpne og lese en tekstfil

1.1.1 Åpne

- Vi bruker funksjonen `open()` for å åpne en fil
- Vi bruker "text"-modus (`"t"`) for å open en *tekst* fil (dvs. ikke "binary")
- Vi bruker en `with`-setning for å spesifisere konteksten der filen er åpen. Utenfor `with`-kodeblokken er filen lukket.

1.1.2 Lese

- Vi bruker "read"-modus (`"r"`) for å lese en fil
- Vi bruker fil metoden `read()` for å få tak i en streng som representerer inneholdet i filen.
- Sørg for at du kjører koden i samme mappe som hvor koden og tekstdokumentet ditt ligger.

Syntaks:

```
with open("filnavn.txt", mode="rt") as f:
    # kodeblokken med innrykk her
```

[1]:
```
# Åpne en fil og tilordne filen til variabelen "f"
with open("Askeladden_og_de_gode_hjelperne.txt", mode="rt") as f:
    # Les alle linjene og lagre dem i en liste
    linjene = f.read().split("\n")
    print(f)

# Nå er filen lukket, vi kan ikke lenger bruke "f", men "linjene" er
# fortsatt definert
print(linjene[0])

# Teller hvor mange linjer i teksten
print(f"Teksten har {len(linjene)} linjer.")

# Teller hvor mange tegner i teksten
n_tegn = 0
for linje in linjene:
    n_tegn = n_tegn + len(linje)
print(f"Teksten har {n_tegn} tegner.")
```

```
<_io.TextIOWrapper name='Askeladden_og_de_gode_hjelperne.txt' mode='rt'
encoding='UTF-8'>
Askeladden og de gode hjelperne
Teksten har 296 linjer.
Teksten har 13028 tegner.
```

[2]:
```
# /!\ Det utløser en feilmelding /!\
# utenfor "with"-konteksten er filen lukket!
f.read().split("\n")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[2], line 3
      1 # /!\ Det utløser en feilmelding /!\
      2 # utenfor "with"-konteksten er filen lukket!
----> 3 f.read().split("\n")

ValueError: I/O operation on closed file.
```

1.1.3 Operasjoner på linjene, noen eksempler

[3]:
```
# Skrive ut innholdet i en tekstfil (5 første linjer)
for linje in linjene[:5]:
    print(linje)
```

```
Askeladden og de gode hjelperne

Det var en gang en konge, og den kongen hadde hørt tale om et skip som gikk like
fort til lands som til vanns; så ville han også ha slikt et, og til den som
kunne bygge det, lovte han ut kongsdatteren og halve kongeriket, og det lyste
```

[4]:
```
# Skrive alt med store bokstaver
# Skrive ut innholdet i en tekstfil (5 første linjer)
for i_linje in range(5):
    linjene[i_linje] = linjene[i_linje].upper()
    print(linjene[i_linje])
```

```
ASKELADDEN OG DE GODE HJELPERNE

DET VAR EN GANG EN KONGE, OG DEN KONGEN HADDE HØRT TALE OM ET SKIP SOM GIKK LIKE
FORT TIL LANDS SOM TIL VANNS; SÅ VILLE HAN OGSÅ HA SLIKT ET, OG TIL DEN SOM
KUNNE BYGGE DET, LOVTE HAN UT KONGSDATTEREN OG HALVE KONGERIKET, OG DET LYSTE
```

1.1.4 Skrive tekst i en fil

Skrive i en ny fil eller overskrive innholdet om filen allerede eksisterer

- Filen åpnes i "write"-modus (`"w"`)
- Vi bruker fil metoden `writelines()` for å skrive en streng eller en liste av streng

[5]:
```
with open("store_bokstaver.txt", mode="wt") as f:
    # Skriv alle linjene i filen "store_bokstaver.txt"
    f.writelines(linjene)
```

Skrive i en ny fil eller legge til tekst på slutten av en fil

- Filen åpnes i "append"-modus (`"a"`)

I eksempelet nedenfor er tanken å kjøre koden flere ganger og legge til et nytt navn-adresse par i tekstfilen hver gang vi kjører koden.

[6]:
```python
navn = input("Hva er navnet ditt?")
adressen = input("Hva er adressen din?")

with open("adresser.txt", mode="at") as f:
    print(f"Du heter {navn} og adressen din er: {adressen}")
    # Skriv navnet og adressen i filen"
    f.writelines(f"Navn:     {navn.upper()}\n")
    f.writelines(f"Adressen: {adressen.upper()}\n")
    # Skriv et skilletegn i filen mellom navn-adresse par
    f.writelines("_"*60 + "\n")
```

```
Du heter Vilma og adressen din er: Stavanger, 4005
```

1.2 Operasjoner: flere eksempler

I eksempelet nedenfor prøver vi å finne alle navn-adressepar i filen der adressen er i Bergen.

[7]:
```python
# Filen åpnes i "read"-modus (r) og den forventes å være en tekstfil (t)
with open("adresser.txt", mode="rt") as f:
    linjene = f.read().split("\n")

for i_linje in range(len(linjene)):
    # Finn indeksen til alle linjene som inneholder "BERGEN"
    if "BERGEN" in linjene[i_linje]:
        # Skriv ut navnet
        print(linjene[i_linje-1])
        # Og adressen
        print(linjene[i_linje])
```

```
Navn:     ODIN
Adressen: BERGEN, 5019
Navn:     MORTEN
Adressen: BERGEN, 5026
```