

Løsningsforslag

Runde 1

1.6 Filer

1.6.1

- a) Ta utgangspunkt i oppgave 1.5.7. I stedet for å legge heltallene inn i en liste, skal de lagres på hver sin linje i en `.txt`-fil som heter `heltall.txt`. Legg inn mellom 10 og 15 verdier. Kall programfilen `skriv_til_fil.py`.

Skriv `notepad heltall.txt` i Terminalvinduet i VS Code og sjekk innholdet i filen med Notisblokk-programmet.

```
data = []
fortsett = True
while fortsett:
    tekst_inn = input('Skriv inn et heltall: ')
    if tekst_inn == '': fortsett = False
    if tekst_inn.isnumeric():
        data.append(tekst_inn)
with open('heltall.txt', mode='w') as fil:
    fil.write('\n'.join(data))
```

```
Skriv inn et heltall: 3
Skriv inn et heltall: 6
Skriv inn et heltall: 12
Skriv inn et heltall: 9
Skriv inn et heltall: 4
Skriv inn et heltall: 11
Skriv inn et heltall: 31
Skriv inn et heltall: 18
Skriv inn et heltall: 8
Skriv inn et heltall: 7
Skriv inn et heltall: 1
Skriv inn et heltall:
```

- b) Bytt `heltall.txt` med en annen elev. Skriv et program som leser innholdet i filen og
- Skriver ut hvor mange tall filen inneholder
 - Skriver ut summen av tallene
 - Skriver ut alle tallene på én linje.

Kall programfilen `les_fra_fil.py`.

```
with open('heltall.txt') as fil:
    lines = fil.readlines()
    print(f'Filen inneholder {len(lines)} tall.')
    print(f'Summen av tallene er {sum(int(line) for line in lines)}.')
    print('Tallene er ', end='')
    for line in lines: print(f'{int(line):5}', end='')
```

```
Filen inneholder 11 tall.
Summen av tallene er 110.
Tallene er      3      6     12      9      4     11     31     18      8      7      1
```

Smidig IT-2

1.6.2

- a) De tre filene `fil_1.xxx`, `fil_2.xxx` og `fil_3.xxx` har hver sin ukjente filtype, som er enten `.txt` (tekstfil), `.jpg` (bildefil) eller `.mp3` (lydfil). Skriv et program som
- Lar brukeren skrive inn et filnavn
 - Sjekker de første bytene i filen og skriver ut om filsignaturen er `.jpg`, `.mp3` eller ukjent.

Kall programmet `sjekk_signatur.py` og finn filtypen til de tre utleverte filene. Husk å åpne filene i *binary mode* med `mode = 'rb'`

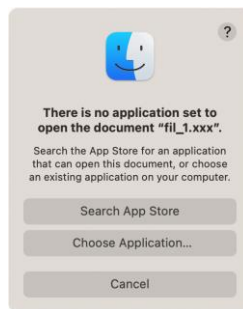
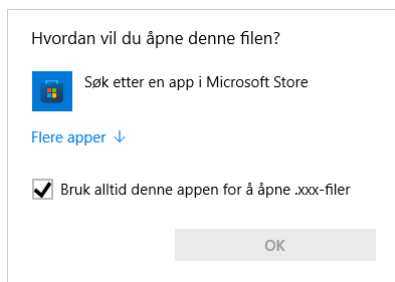
```
print('Sjekker filsignatur')
filnavn = input('Skriv inn filnavn: ')
with open(filnavn, mode='rb') as fil:
    signatur = fil.read(3)
    if signatur[0:2] == b'\xff\xd8':
        print(f'{filnavn} er en .jpg fil.')
    elif signatur == b'ID3':
        print(f'{filnavn} er en .mp3 fil.')
    else:
        print(f'{filnavn} har ukjent filtype.')
```

Skriv inn filnavn: `fil_1.xxx`
`fil_1.xxx` er en .jpg fil.

Skriv inn filnavn: `fil_2.xxx`
`fil_2.xxx` er en .mp3 fil.

Skriv inn filnavn: `fil_3.xxx`
`fil_3.xxx` har ukjent filtype.

- b) Dobbelt-klikk på filene i filutforskeren (macOS: Finder).
Hva skjer? Forklar.



Ingen program er forbundet med filtypen `.xxx` så Windows/macOS spør om hvilket program vil bruke til å åpne slike filer.

Gi filene nytt navn med riktig filtype. Dobbelt-klikk på filene i filutforskeren.
Hva skjer? Forklar.

Windows har forbundet `.jpg` filtypen med Bilder-appen. Når vi dobbelt-klikker på en `.jpg`-fil, startes **Bilder**-appen, som leser inn og viser `.jpg`-filen (Preview i macOS). Tilsvarende for `.mp3`-filer, som er forbundet med **Groove Musikk**-appen (Music i macOS), og `.txt`-filen som er forbundet med **Notisblokk**-programmet (TextEdit i macOS).

Gi filene det gamle navnet med filtype `xxx`. Prøv å åpne filene i en nettleser.
Hva skjer? Forklar.

Dra filene fra filutforskeren/Finder og slipp dem over Chrome. Nettleseren ser ut til å være smartere enn Windows/macOS. Nettleseren viser bildet, spiller av lyden og viser teksten til filene som har feil filtype. Nettleseren klarer å analysere innholdet av filen for å vite hvordan den skal behandles. Dette gjelder for Chrome på PC og Mac. Det fungerer ikke for Safari.

Alle filene inneholder et kodeord.
Hvilken fil inneholder hvilket kodeord?

Smidig IT-2

- fil_1.xxx inneholder kodeordet DIGITALISERINGSDIREKTORATET
- fil_2.xxx inneholder kodeordet KRYPTERING
- fil_3.xxx inneholder kodeordet HACKING