

Oppgavesett dag 3

Introkurs ved Folkeuniversitetet

Oppgave 1 *Behandle bestillinger fra fil*

- a) Last ned fila “bestillinger_mat.txt” og lagre den i samme mappe som du har Python koden din.
- b) Åpne fila i Python og lagre fil-objektet i en variabel, `tekstfil`, med kommandoen `with open(bestillinger_mat. txt) as tekstfil:`
OBS: hvis du får `FileNotFoundError` bør du dobbeltsjekke at fila ligger i samme mappe som Pythonfila di
- c) Les inn tekstlinjene i fila med `tekstfil.readlines()` og lagre resultatet i en variabel, `tekstfillinjer`.
- d) Skriv ut tekstfillinjer til terminalvinduet. Hva slags type variabel er dette? Hva inneholder den?
- e) Bruk en løkke til å løkke deg igjennom hver linje i tekstfillinjer og skriv ut linjene til terminalen
- f) Bruk `tekstfillinjer.split(',')` til å splitte tekstlinja inn i en liste av kommaseparerte elementer. Lagre denne lista i en variabel, `ord_i_linje`
- g) Det første elementet i `ord_i_linje` er *navnet på varen*. Det andre er *prisen*. Du kan bruke en *flettestreng* sammen med `print` til å skrive ut navnet og prisen pent i samme linje. Prøv å gjenskape følgende output:

```
Bestilling: Hummus, Pris: 60,-  
Bestilling: Scampi wok, Pris: 179,-
```

```
Bestilling: Gulrotkake, Pris: 79,-
Bestilling: Veggie Burger, Pris: 138,-
...
```

- h) Bruk en tellevariabel til å summere alle prisene i fila og skriv ut totalprisen i slutten av koden. Hvor mye kostet alle bestillingene til sammen?
- i) Kjør programmet ditt med fila "bestillinger_møbler.txt" som inputfil istedenfor. Hvor mye kostet disse bestillingene til sammen?

Løsning oppgave 1 *Behandle bestillinger fra fil*

- a) oving4/oppgave1/bestillinger_mat.txt
oving4/oppgave1/bestillinger_møbler.txt

b)

```
1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
```

c)

```
1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
2     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
```

d)

```
1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
2     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
3     print(tekstfillinjer)
```

e)

```

1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
2     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
3     print(tekstfillinjer)
4     for linje in tekstfillinjer:
5         print(linje)

```

f)

```

1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
2     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
3     print(tekstfillinjer)
4     for linje in tekstfillinjer:
5         print(linje)
6         ord_i_linje = linje.split(',')

```

g)

```

1 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
2     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
3     print(tekstfillinjer)
4     for linje in tekstfillinjer:
5         print(linje)
6         ord_i_linje = linje.split(',')
7         print(f'Bestilling: {ord_i_linje[0]}, Pris
            : {ord_i_linje[1]},-')

```

h)

```

1 _sum = int()
2 with open("bestillinger_mat.txt", mode="r",
    encoding='utf-8') as tekstfil:
3     tekstfillinjer = map(str.strip, tekstfil.
        readlines())
4     print(tekstfillinjer)

```

```

5     for linje in tekstfillinjer:
6         print(linje)
7         ord_i_linje = linje.split(',')
8         print(f'Bestilling: {ord_i_linje[0]}, Pris
          : {ord_i_linje[1]},-')
9         _sum += int(ord_i_linje[1])
10    print(f'Total kost for bestillinger: {_sum},-')
    )

```

i)

```

1  with open("bestillinger_mat.txt", mode="r",
2      encoding='utf-8') as tekstfil:
3      tekstfillinjer = map(str.strip, tekstfil.
4          readlines())
5      print(tekstfillinjer)
6      for linje in tekstfillinjer:
7          print(linje)
8          ord_i_linje = linje.split(',')

```

```
Total kost for bestillinger: 1551,-
```

j)

```
Total kost for bestillinger: 6672,-
```

Oppgave 2 *Lese inn middagsforslag*

I denne oppgaven skal vi lese inn middagsforslag fra en vedlagt fil, stokke om på rekkefølgen slik at alle elementene kommer i tilfeldig rekkefølge.

- a) Last ned filen *middagsforslag.txt* og lag et nytt Python-program. Sørg for at begge filene ligger i samme mappe.

Filene inneholder en liste over 31 middagsforslag. La programmet ditt *åpne* filen, *lese* innholdet og *lagre* det som en liste.

- b) Iterer gjennom elementene i listen, og bruk `enumerate` eller en teller for å skrive ut hvilket nummer i listen hvert element er, samt elementet

(middagsforslaget) i seg selv.

- c) Modifiser løkken i forrige oppgave slik at for en valgt måned, si mai, skriver programmet ut en dato samt middagsforslag for den dagen, noe slikt som

```
Den 1. mai skal vi ha laks og ovnsbakte grønnsaker
.
Den 2. mai skal vi ha gulrotsuppe.
Den 3. mai skal vi ha kyllingwok.
(...)
```

- d) Bruk funksjonen `shuffle` fra modulen `random` til å stokke om elementene i listen, før løkken, slik at middagsforslagene kommer i tilfeldig rekkefølge.

Nå har du laget et program som kan generere middagsforslag for en hel måned!

Løsning oppgave 2 *Lese inn middagsforslag*

a)

```
1 f_in = open("middagsforslag.txt", "r")
2 alle_middager = f_in.read().splitlines()
3 f_in.close()
```

b)

```
1 for (dag, middag) in enumerate(alle_middager):
2     print(dag, middag)
```

c)

```
1 for (dag, middag) in enumerate(alle_middager):
2     print(f"Den {dag + 1}. mai skal vi ha {middag}
           til middag.")
```

d) Fullstendig program:

```

1  import random
2
3  f_in = open("middagsforslag.txt", "r")    # "r" =
      read
4  alle_middager = f_in.read().splitlines()
5  f_in.close()
6
7  random.shuffle(alle_middager)
8
9  for (dag, middag) in enumerate(alle_middager):
10     print(f"Den {dag + 1}. mai skal vi ha {middag}
        til middag.")

```

Oppgave 3 *Verdens 100 høyeste fjell*

Visste du at samtlige av verdens 100 høyeste fjell ligger i Asia? I denne oppgaven skal vi se litt nærmere på disse; vi skal øve på å lese inn informasjon fra en fil og etterbehandle denne informasjonen til å skrive ut informasjon av interesse.

I denne oppgaven er det i stor grad lagt opp til at du selv skal finne ut *hvordan* du skal sette opp programmet ditt; i oppgaveteksten under følger bare spørsmålene vi vil finne ut av. Her er det meningen å lese inn innholdet fra en fil, legge dette i en hensiktsmessig datastruktur (liste eller oppslagsverk), og bruke informasjonen til å svare på hver av deloppgavene. Det kan være lurt å dele opp programmet ditt i flere funksjoner!

Filen `hundrehoyestefjell.csv` inneholder en oversikt over verdens 100 høyeste fjell (hentet fra «List of highest mountains on Earth» på Wikipedia; vi har bare tatt med ett navn der det er flere mulige, og ikke tatt hensyn til eventuelle uenigheter om grenser og lignende). Hver linje angir navnet på fjellet, høyde i meter over havet samt hvilke land fjellet ligger i, altså slik:

```

Abi Gamin ,7355 ,India ,Kina
Annapurna I ,8091 ,Nepal
...

```

Hver opplysning om samme fjell er altså gitt på samme linje, delt opp ved kommaer – og hvis et fjell ligger i flere land får vi altså en lengre linje (målt

i antall kommaer/opplysninger).

Skriv et program som leser inn denne filen, og som svarer på følgende oppgaver:

- a) Hvilke er verdens 10 høyeste fjelltopper? Skriv ut en oversikt i rekkefølge.
- b) Hvor mange fjelltopper ligger i 2 land? Og hvor mange ligger i 3?
- c) Hvor mange fjell ligger i hvert av landene? Skriv ut en oversikt over hvert land og hvor mange fjelltopper som ligger i dette landet.
- d) Av landene som er gitt, hvilken topp er høyest i hvert land? Skriv ut en oversikt over hvert land og den tilhørende høyeste fjelltoppen.

Løsning oppgave 3 *Verdens 100 høyeste fjell*

```
1 def les_inn_fil(filnavn):
2     """
3
4     Henter inn opplysninger i filen, og legger disse i
5     en liste slik at vi har
6     posisjon 0 : navn på fjellet
7     posisjon 1 : høyde
8     posisjon 2 : liste over hvilke land fjellet
9     ligger i
10
11     """
12     fin = open(filnavn)
13     opplysninger = fin.read().splitlines()
14
15     fjelltoppinfo = []
16
17     for linje in opplysninger:
18         linje_info = linje.split(",")
19         fjelltoppinfo.append([linje_info[0],
20                               linje_info[1], linje_info[2:]])
```

```

19     return fjelltoppinfo
20
21
22
23 def finn_10_hoyeste(fjelltoppinfo):
24     """
25
26     Sorterer informasjonen om navn og høyde etter hø
27     yde; bruker dette
28     til å skrive ut de 10 første.
29
30     """
31     # lag et oppslagsverk etter høyde
32
33     etter_hoyde = {}
34
35     for info in fjelltoppinfo:
36         navn = info[0]
37         hoyde = info[1]
38
39         etter_hoyde[hoyde] = navn
40
41     # sorter etter høyde
42
43     hoyeste = list(etter_hoyde.keys())
44     hoyeste.sort()
45
46     print("Verdens 10 høyeste fjell er: ")
47
48     for i in range(10):
49         hoyde = hoyeste[i]
50         print(f"- {etter_hoyde[hoyde]} ({hoyde} moh
51             .)")
52
53 def finn_antall_i_n_land(fjelltoppinfo, n):
54     """
55
56     Går igjennom listen over hvilke land hver

```



```

57     fjelltopp ligger i
    og teller alle tilfeller der denne listen har
    lengden n.
58
59     """
60
61     antall_totalt = 0
62
63     for info in fjelltoppinfo:
64         alle_land = info[2]
65
66         if len(alle_land)==n:
67             antall_totalt += 1
68
69     print(f"Det er {antall_totalt} fjelltopper som
    ligger i {n} land.")
70
71
72 def finn_antall_fjell_pr_land(fjelltoppinfo):
73     """
74
75     Går igjennom fjelltoppinfo og teller opp antall
    forekomster av hvert
76     land. Her vil alle fjelltopper som er i flere land
    telles en gang
77     pr land det ligger i.
78
79     """
80
81     antall_fjell = {}
82
83     for info in fjelltoppinfo:
84         navn = info[0]
85         hoyde = info[1]
86         alle_land = info[2]
87
88         for land in alle_land:
89             if land in antall_fjell.keys():
90                 antall_fjell[land] += 1
91             else:

```

```

192         antall_fjell[land] = 1
193
194     print("Av verdens 100 høyeste fjell ligger ...")
195
196     for land in antall_fjell.keys():
197         print(f"- {antall_fjell[land]} i {land}")
198
199
200 def finn_hoyeste_i_hvert_land(fjelltoppinfo):
201     """
202
203     Lagrer informasjonen i en dictionary, med
204     midlertidige variable for
205     høyde og tilhørende navn. Hvis vi kommer til et
206     fjell som er høyere
207     for et gitt tall oppdaterer vi både høyden og
208     navnet.
209
210     """
211
212     hoyeste = {}
213
214     for info in fjelltoppinfo:
215         navn = info[0]
216         hoyde = info[1]
217         alle_land = info[2]
218
219         for land in alle_land:
220             if land in hoyeste.keys():
221                 if hoyde > hoyeste[land]["høyde"]:
222                     hoyeste[land]["høyde"] = hoyde
223                     hoyeste[land]["navn"] = navn
224             else:
225                 hoyeste[land] = {}
226                 hoyeste[land]["høyde"] = hoyde
227                 hoyeste[land]["navn"] = navn
228
229     print("Det høyeste fjellet i ...")
230
231     for land in hoyeste.keys():

```

```

129         print(f"- {land} er {hoeyeste[land]['navn']}
        ({hoeyeste[land]['høyde']} moh.).")
130
131
132     filnavn = "hundrehoyestefjell.csv"
133     fjelltoppinfo = les_inn_fil(filnavn)
134
135     print("a)")
136     finn_10_hoyeste(fjelltoppinfo)
137     print("\nb)")
138     finn_antall_i_n_land(fjelltoppinfo, 2)
139     finn_antall_i_n_land(fjelltoppinfo, 3)
140     print("\nc)")
141     finn_antall_fjell_pr_land(fjelltoppinfo)
142     print("\nd)")
143     finn_hoyeste_i_hvert_land(fjelltoppinfo)

```

Oppgave 4 *Ansatte*

I denne oppgaven ser vi hvordan vi kan få bruk for programmering på en litt mer praktisk måte. Vi skal undersøke en bedrift som har veldig mange ansatte og se at programmering effektiviserer og løser oppgavene mye raskere enn vi kunne gjort for hånd.

- a) Last ned filen *ansatte.txt* og lag et nytt Python-program. Sørg for at begge filene ligger i samme mappe.
- b) Ta en titt på txt-filen ved å åpne den (dobbeltklikk på den) og se hvordan den ser ut. Da ser vi at den øverste delen av filen ser slik ut:

```

1 navn,alder,inntekt,kontor
2 Monica,27,470303,Stavanger
3 Marianne,46,485394,Oslo
4 Erik,44,713323,Stavanger
5 ....

```

Her ser vi at listen inneholder informasjon om de ansatte i en veldig stor bedrift. Den første linjen forteller oss hvordan dataene i filen er ordnet.

Vi ser altså at hver linje forteller navnet på den ansatte, alder, inntekt og hvilket kontor de jobber ved.

- c) Vi kan nå begynne å analysere dataene i filen. Les inn alle linjene i filen til en liste i Python. Kall listen for `ansatte`.
- d) Ved hjelp av en for-løkke kan vi gå gjennom alle elementene i `ansatte`. Print ut hver linje for å sjekke at du har fått lest inn filen riktig.
- e) Hvor mange ansatte er det i bedriften? (Hint: Hvor mange linjer er det i fila?)
- f) På hver linje har vi **teksten** eller strengen `"Monica,27,470303,Stavanger"`. For å dele denne strengen opp i den informasjonen den inneholder kan vi bruke kommandoen `split(",")` som deler opp denne linjen med tekst ved hvert komma. Da får vi en liste for den ansatte der det første elementet er navnet, den neste alderen også videre. Skriv nå bare ut navnet på den ansatte for hver ansatt i bedriften.
- g) Nå som vi har klart å hente ut de fire bitene med informasjon vi har på hver ansatt kan vi løse de følgende spørsmålene ved programmering. Velg et par av de du synes er mest interessant og prøv å finne svaret. Vanskelighetsgraden øker nedover lista.
 - Hva er gjennomsnittsalderen på de ansatte?
 - Hvor mye bruker bedriften på lønninger?
 - Hvem tjener mest?
 - Hvilken ansatt er yngst?
 - Ved hvilket kontor sitter det flest ansatte?
 - Hva er det mest populære navnet i bedriften? (Hint: Bruk et oppslagsverk (dictionary).)

Løsning oppgave 4 *Ansatte*

a)

```
1 fil = open("ansatte.txt")
2 ansatte = fil.readlines()
```

```
3 fil.close()
```

b)

```
1 fil = open("ansatte.txt")
2 ansatte = fil.readlines()
3 fil.close()
4
5 for ansatt in ansatte:
6     print(ansatt)
```

c)

```
1 fil = open("ansatte.txt")
2 ansatte = fil.readlines()
3 fil.close()
4
5 antall_ansatte = -1 #starter på -1 for å ikke
   telle med headeren i tekst-filen, kunne også
   trukket fra 1 i etterkant.
6
7 for ansatt in ansatte:
8     antall_ansatte += 1
9
10 print(f"Det er {antall_ansatte} ansatte i
    bedriften. Det ville det tatt lang tid å finne
    ut av for hånd!")
```

Alternativt:

```
1 fil = open("ansatte.txt")
2 ansatte = fil.readlines()
3 fil.close()
4
5 antall_ansatte = len(ansatte) - 1
6
7 print(f"Det er {antall_ansatte} ansatte i
    bedriften. Det ville det tatt lang tid å finne
    ut av for hånd!")
```

Det er 1265 ansatte i bedriften. Det ville det tatt lang tid å finne ut av for hånd!

Husk å ikke telle med header-en i fila!

d)

```
1 fil = open("ansatte.txt")
2 ansatte = fil.read().splitlines()
3 fil.close()
4
5 for ansatt in ansatte:
6     ansatt_informasjon = ansatt.split(",")
7     print(ansatt_informasjon[0])
```

e) Her er et løsningsforslag til gjennomsnittsalderen.

```
1 fil = open("ansatte.txt")
2 ansatte = fil.readlines()
3 fil.close()
4
5 alder = 0
6 antall_ansatte = 0
7 for ansatt in ansatte[1:]:
8     ansatt_informasjon = ansatt.split(",")
9     alder += int(ansatt_informasjon[1])
10    antall_ansatte += 1
11
12 gjsnitt = alder/antall_ansatte
13 print(f"Gjennomsnittsalderen i bedriften er {
    gjsnitt:.2f} år.")
```

Gjennomsnittsalderen i bedriften er 44.35 år.

Svaret på den andre spørsmålene er henholdsvis:

- Gjennomsnittsalder: 44.35
- Lønninger: 568 millioner
- Mest populære kontor: Stavanger

- Størst lønning: Rolf med 744521 kr
- Yngste ansatt: Rolf på 14 år, ja alderene er tilfeldig generert, men noen genier finnes det vel?
- Vanligste navn: Stian, 22 ganger