

Oppgavesett dag 3

Introkurs ved Folkeuniversitetet

Innlesing av filer

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 `tekstfillinje.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?

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.

Filen 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 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!

Oppgave 3 *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?
 - Hva er den høyeste lønningen?
 - Hvor gammel er den yngste ansatte?

Plotting med lister

Oppgave 4 *Plotte x- og y-koordinater*

Koden under importerer plot og show fra pylab og oppretter en liste med x-koordinater og en liste med y-koordinater:

```
1 from pylab import plot, show
2 x_koordinater = [1, 2, 3, 4]
3 y_koordinater = [3, 2, 1.5, 2]
```

- Kopier koden over inn i et Python-program.
- Bruk plot funksjonen til å lage et plot med x_koordinater på x-aksen og y_koordinater på y-aksen.
- Kall på show i slutten av programmet slik at plottet vises frem.

Oppgave 5 *Plotte populasjonstall*

Befolkningstall i Norge for årene 2010-2020 er gitt i tabellen under (tall fra Statistisk Sentralbyrå):

År	2015	2016	2017	2018	2019	2020
Befolkning.	5 165 802	5 213 985	5 258 317	5 295 619	5 328 212	5 367 580

- Lag en liste befolkningstall som inneholder befolkningstall og en liste tidspunkt som inneholder året befolkningstallet ble registrert for.
- Bruk plot-funksjonen i pylab til å lage et plot med år på x-aksen og befolkningstall på y-aksen.
- Bruk xlabel- og ylabel-funksjonene i pylab til å legge merkelapper på x- og y-aksen.
- Importer savefig funksjonen fra pylab og bruk den til å lagre plottet som en png-fil.