

Oppgavesett dag 2

Avansert kurs ved Folkeuniversitetet

I denne seksjonen finner du oppgaver som hører til dag 2 av Folkeuniversitetet og Simula Learnings kurs Programmering i Python - Avansert. Tema for andre dag er tekstbehandling, ordlister, csv og excel, og pandas. Dersom du står fast er det bare å spørre. God koding!

Tekstbehandling

Oppgave 1 *Måneder med r i navnet*

Mila har hørt at hun bør få i seg ekstra D-vitamin i måneder som inneholder “r”. Så hun har lyst til å bruke Python til å løkke gjennom alle månednavnene og skrive ut kun de som inneholder bokstaven “r”.

- a) Opprett en liste, månednavn som inneholder navnet på alle 12 måneder
- b) Bruk en **for**-løkke til å løkke igjennom disse månedene og skriv ut hvert månednavn til terminalen
- c) Modifiser programmet til å bruke en betingelse for å kun skrive ut de månedene som inneholder bokstaven r. (**Hint**: du kan bruke **in**-operatoren for å se om en tekststreng er inneholdt i en annen tekststreng)

Oppgave 2 *Navn og alder*

Husk at du kan hente ut deler av en tekststreng ved å skrive for eksempel

```
1 tekst_del1 = tekst[5:]  
2 enkel_bokstav = tekst[4]
```

```
3 tekst_del2 = tekst[:4]
```

Her skal vi bruke dette til å hente ut litt nyttig informasjon fra en setning.

- a) Opprett en variabel *setning* som inneholder «Hei, jeg heter Emilie og er 25 år gammel.» Du kan bytte ut navn og alder med dine egne.
- b) Bruk indeksering til å hente ut navnet, slik at du får en variabel *navn* som har verdien «Emilie». Skriv den ut for å sjekke at det blir riktig.
- c) Bruk indeksering til å hente ut alderen, slik at du får en variabel *alder* som har verdien «25». Skriv den ut for å sjekke om det blir riktig.
- d) Endre navn og alder i setningen. Fungerer programmet ditt fortsatt? Er det noe du må endre på? Tenk gjerne på antall bokstaver det er i navnet, og antall sifre i alderen.

(Her er det meningen å bare fikle litt, ikke å lage en løsning som fungerer for alle navn og aldre.)

Oppgave 3 *Statusbeskjeder*

I denne oppgaven skal du bruke *replace*-funksjonen til å fjerne unødvendig tekst fra en tekststreng. *Replace*-syntaksen ser slik ut:

```
1 tekst = "Tekststreng du ønsker å endre".replace("å  
   endre", "det du vil endre til")  
2 print(tekst)
```

```
Tekststreng du ønsker det du vil endre til
```

Monica har en liste med status-beskjeder som hun har lest inn fra en tekst fil. Listen ser slik ut:

```
1 beskjeder = ['STATUS - alt ok', 'STATUS - trenger  
   oppdatering', 'STATUS - alt ok', 'STATUS - fatal  
   feil, trenger restart']
```

Hun ønsker å lage en ny liste, *beskjeder_kun_info* som inneholder hver beskjed, men uten 'STATUS -' biten.

- a) Lim lista over inn i en Python fil
- b) Bruk en **for**-løkke til å løkke igjennom hver beskjed i listen, beskjeder. Skriv ut beskjedene inne i løkka
- c) Opprett en tom liste, `beskjeder_kun_info` over koden for **for**-løkka.
- d) Inne i løkka: Bruk `replace` til å bytte ut '**STATUS -**' med en tom streng `' '` for hver beskjed i lista. Bruk `.append` til å lagre den modifiserte beskjeden i `beskjeder_kun_info`.
- e) Skriv ut `beskjeder_kun_info` til terminalen og dobbeltsjekk at det ble riktig.

Ordlister

Oppgave 4 *Telefonbok*

I denne oppgaven skal vi lage et oppslagsverk for å ta vare på telefonnummer. En måte å opprette et oppslagsverk er å bruke følgende syntaks:

```
1 oppslagsverk = {  
2     nøkkelnavn1 : verdi1,  
3     nøkkelnavn2 : verdi2  
4 }
```

Hvor `nøkkelnavn` er nøklene du bruker til å slå opp med og verdiene er det som skal slås opp.

Denne datatypen kan vi bruke til å lage en telefonbok hvor nøklene er navnet på vennene våre som en *tekststreng* og verdiene er telefonnummere som *heltall*

- a) Lag et oppslagsverk, `telefonbok` med følgende navn og telefonnummer:

Navn	Telefonnummer
'Hedda'	12345678
'Ejura'	11235813
'Henrik'	31415926
'Antoni'	99999999

Tabell 1: Oversikt over plottestil-tekststrenger

For å slå opp i et oppslagsverk kan du bruke *klammeindeksering* og skrive nøkkelen til det du vil slå opp mellom klammene. Eksempel: `oppslagsverk[nøkkelnavn]`

- b) Skriv kode som slår opp i telefonboken din og skriver ut telefonnummeret til 'Ejura' til terminalen.

Nå skal vi utvide programmet til å be en bruker om navn og skrive ut tilhørende telefonnummer dersom det ligger i telefonboken. For å sjekke om en nøkkel finnes i et oppslagsverk kan du bruke `in`. For eksempel for å sjekke om 'Hedda' finnes i telefonbok kan du bruke betingelsen `if 'Hedda' in telefonbok`.

- c) Bruk `input` til å be en bruker om å skrive inn et navn og lagre navnet i en variabel, `navn`.
- d) Bruk `if` og en betingelse til å sjekke om navn finnes i telefonboken din. Dersom navnet finnes i telefonboken skal du skrive ut telefonnummeret til terminalen. Dersom navnet ikke ligger i telefonboken, skriv ut en beskjed om at navnet ikke finnes.

Oppgave 5 *Antall bokstaver*

I denne oppgaven skal vi telle litt – vi skal se på en tekststreng, og analysere denne. Til slutt vil du ha laget et program som kan analysere en tekststreng og si noe om hvilke bokstaver som forekommer oftest.

- a) Opprett en variabel som inneholder en tekststreng – for eksempel teksten «supercalifragilisticexpialidocious». Bruk en løkke til å skrive ut hver

bokstav i denne teksten. Her blir utskriften noe slikt som

```
s
u
p
e
r
( . . . )
```

- b) Opprett et tomt *oppslagsverk*, kall denne for `antall_forekomster`. Denne skal vi lagre antall forekomster av hver bokstav i, med bokstaver som nøkler og tall som elementer.

Modifiser løkken du skrev i forrige deloppgave, slik at den *først* sjekker om bokstaven allerede finnes i `antall_forekomster` eller ikke. Hvis den finnes, skal vi telle oppover – da har vi allerede sett bokstaven før. Oppdater telleren ved å bruke `+=`-operatoren. Hvis den ikke finnes, skal vi starte å telle – sett elementet tilsvarende nøkkelen til å være `1`.

- c) Skriv en løkke for å skrive ut innholdet i oppslagsverket ditt. For hver bokstav, skriv ut en setning om at «Bokstaven [bokstav] forekommer [n] gang(er) i teksten.»

- d) Utvid programmet ditt videre, og lag et nytt oppslagsverk med tall som nøkler og en liste av bokstaver som element, der hver nøkkel angir hvor mange ganger disse bokstavene forekommer i teksten.

Lag en ny løkke som skriver ut for hver nøkkel «Bokstaven(e) [liste over bokstaver] forekommer alle [antall forekomster] gang(er) i teksten.».

- e) Utvid programmet ditt enda en gang for å finne (alle) bokstave(er) som forekommer *flest* antall ganger. I teksten over er det «i», som forekommer 7 ganger. Bruk gjerne `max`-operatoren kombinert med en liste.

- f) Prøv deg til slutt frem med et par ulike tekststrenger – gjerne noen lange. Bruk f.eks. `input` for å lese inn teksten fra kommandolinjen. Hvilken bokstav vil du tro er den vanligste i norsk språk? Prøv deg f.eks. frem med tekster fra avisartikler, meldinger og lignende.

Pandas

Oppgave 6 *Analyse av COVID-19*

I denne oppgaven skal vi analysere en oversikt over COVID-19 tilfeller i verden med statistikkbiblioteket Pandas.

- a) Last ned filen *covid-19.csv* og lag et nytt Python-program. Sørg for at begge filene ligger i samme mappe. Ta en titt på csv-filen ved å åpne den og se hvordan den ser ut. Vi kan nå begynne å analysere dataene i filen. Bruk Pandas til å lese inn csv-filen til en dataframe.
- b) Analyser filen ved å printe ut informasjon om den. Bruk `head()`, `info()`, `describe()`, `dtypes`, `shape` og lignende til å få en oversikt over dataene som er lest inn. Bruk gjerne dokumentasjonen til Pandas for å se hvordan for eksempel `head()` fungerer.
- c) Hvor mange tilfeller er det totalt i verden?
- d) Hvor mange tilfeller er det gjennomsnittlig i verden?
- e) Hvor mange tilfeller er det i median i verden?
- f) Hvilket land har flest døde per million innbygger?
- g) Sorter datasettet på antall mennesker og plot i barplot det landet sine antall tilfeller og antall døde.

Oppgave 7 *Netflix med Pandas*

I denne oppgaven skal vi analysere Netflix sitt bibliotek med statistikkbiblioteket Pandas.

- a) Last ned filen *netflix.xls* og lag et nytt Python-program. Sørg for at begge filene ligger i samme mappe. Ta en titt på xls-filen ved å åpne den og se hvordan den ser ut. Vi kan nå begynne å analysere dataene i filen. Bruk Pandas til å lese inn excel-filen til en dataframe.
- b) Analyser filen ved å printe ut informasjon om den. Bruk `head()`, `info()`, `describe()`, `dtypes`, `shape`, `isnull().mean()` og lignende til å få en oversikt over dataene som er lest inn. Bruk gjerne dokumentasjonen

til Pandas for å se hvordan for eksempel `head()` fungerer.

- c) Plot hvor mange serier og filmer det er i et barplot. Du får tilgang på dette gjennom kolonnen **type**. Bruk `dataframe["type"].value_counts()` for å finne verdiene som skal plottes. Deretter finnes det en innebygd plot-funksjon i det som returneres fra `value_counts()`.
- d) Del filen i to mellom serier og filmer. Skriv til ny fil *netflix_movies.xls* og *netflix_shows.xls*. Du kan lage en separat dataframe for filmer ved å skrive `dataframe[dataframe["type"] == "Movie"]`. Gjør tilsvarende for serier og skriv ut de enkelte dataframene til fil.
- e) Fortsett gjerne ved å undersøke filen ved bruk av Pandas. Det er mye forskjellig informasjon å hente ut. For eksempel er det interessant å vite hva medianen er for utgivelsesåret eller hvor mange filmer Steven Spielberg har regissert.