

Outline

- **Lese, bruke og skrive data i filer**
- Eksempel: kombinere data fra ulike filer
- Mer om funksjoner: parametre og skop for variabler

Innlesing fra fil

- Å hente data fra filer er gøy!
 - Man kan jobbe på mye større og mer spennende data enn fra tastatur
 - Man slipper å taste det inn hver gang man kjører
- Innlesing fra tekstfiler er veldig rett frem i Python
 - En tekstfil åpnet i Python er faktisk en *samling* av linjer
 - Man kan dermed iterere gjennom linjer i filen vha for-løkke
 - (Det finnes også mange andre måter å gjøre det på)

Hvordan lese inn fra fil

- Først åpne en fil (*biblioteksfunksjonen `open` lager et fil-objekt*):
 - `min_fil = open("mittFilNavn.txt")`
- Deretter iterere gjennom hver linje i filen:
 - `min_fil = open("mittFilNavn.txt")`
`for linje in min_fil:`
- Inni for-løkken kan man gjøre noe med linjen
 - `min_fil = open("mittFilNavn.txt")`
`for linje in min_fil:`
`print("Her fant jeg: " + linje)`
- [les_fra_fil.py]

Organisere informasjon innad på linjer

- Det er ofte nyttig å organisere informasjon på bestemte måter i en fil:
 - Kan gruppere: én gruppe på hver linje, og alt tilhørende gruppen inndelt langs linjen
 - Representerer en tabell: hver linje er en rad, og hver linje er videre inndelt i verdier per kolonne (som i et regneark)
- Slike filer er ofte i *tabulært format*, hvor man bruker et spesifikt tegn (f.eks. komma) for å skille verdier innad i linje

Lese tabulære filer i python

- En tabulær fil er en helt vanlig tekstfil, hvor man kan iterere gjennom linjer
 - `for linje in open("min_tabular_fil.csv"):`
- Hver linje kan splittes til en liste (*basert på separator-tegn*)
 - `biter = linje.split(",")` #Splitter på hvert komma
 - `biter = linje.split()` #Splitter på blanke tegn
- Man kan hente ut bestemte biter (kolonner):
 - `navn = biter[0]`
`alder = int(biter[1])`
- Man kan også iterere gjennom bitene:
 - `for bit in biter:`
`print("Jeg fant: " + bit)`

Bygge samling fra innhold i fil

- Vi har tidligere laget to versjoner av hovedstad-program:
 - I uke 1 sjekket vi land med if-setninger og printet hovedstad
 - I uke 3 la vi land og hovedsted inn i en ordbok og slo opp land
- Nå vil vi bygge ordboken fra innholdet i en fil
 - Vi trenger informasjon som kobler étt land med én hovedstad
 - Bruker tabulær fil hvor hver linje har land og tilhørende hovedstad
 - Går gjennom hver linje i fil, splitter, og legger inn i ordbok
- [hovedstad.py]

Oppsummer data gruppert på linjer

- Vi vil finne hvilken måned hadde høyest dagsnedbør
 - Har tabulær fil med nedbørsmengde per dag gjennom året
 - Dagsnedbør er gruppert på én linje for hver måned
- Fremgangsmåte: nøstet løkke inkludert splitting
 - Itererer gjennom måneder (linjer)
 - Splitter hver linje til liste over dagsnedbør
 - Itererer gjennom dagsnedbør og finner høyeste
 - Skriver ut høyeste verdi for denne måneden
- [regn.py]

Oppgave

- Gitt en tekstfil av typen vist under (navn og alder), skriv et program som leser filen og skriver ut navnet på den eldste personen i fila

alder.csv

```
odlaug:76  
oluf:65  
gunda:74  
malfrid:80  
godtfred:68
```


En mulig løsning

- [alder.py]

Hvordan skrive til tekstfil

- Først åpne en fil for skriving:
 - `min_fil = open("mittUtFilNavn.txt", "w")`
- Deretter skrive tekst (en streng-verdi) til slutten av filen:
 - `min_fil = open("mittUtFilNavn.txt", "w")`
`min_fil.write("Dette havner i filen")`
- Man vil typisk ha linjeskift i det man skriver ut:
 - `min_fil.write("Dette havner i filen\n")`
`min_fil.write("Dette havner på\n ulike\n linjer\n")`
- Når man er ferdig bør man lukke filen:
 - `min_fil.close()`
- [elefanter.py]