

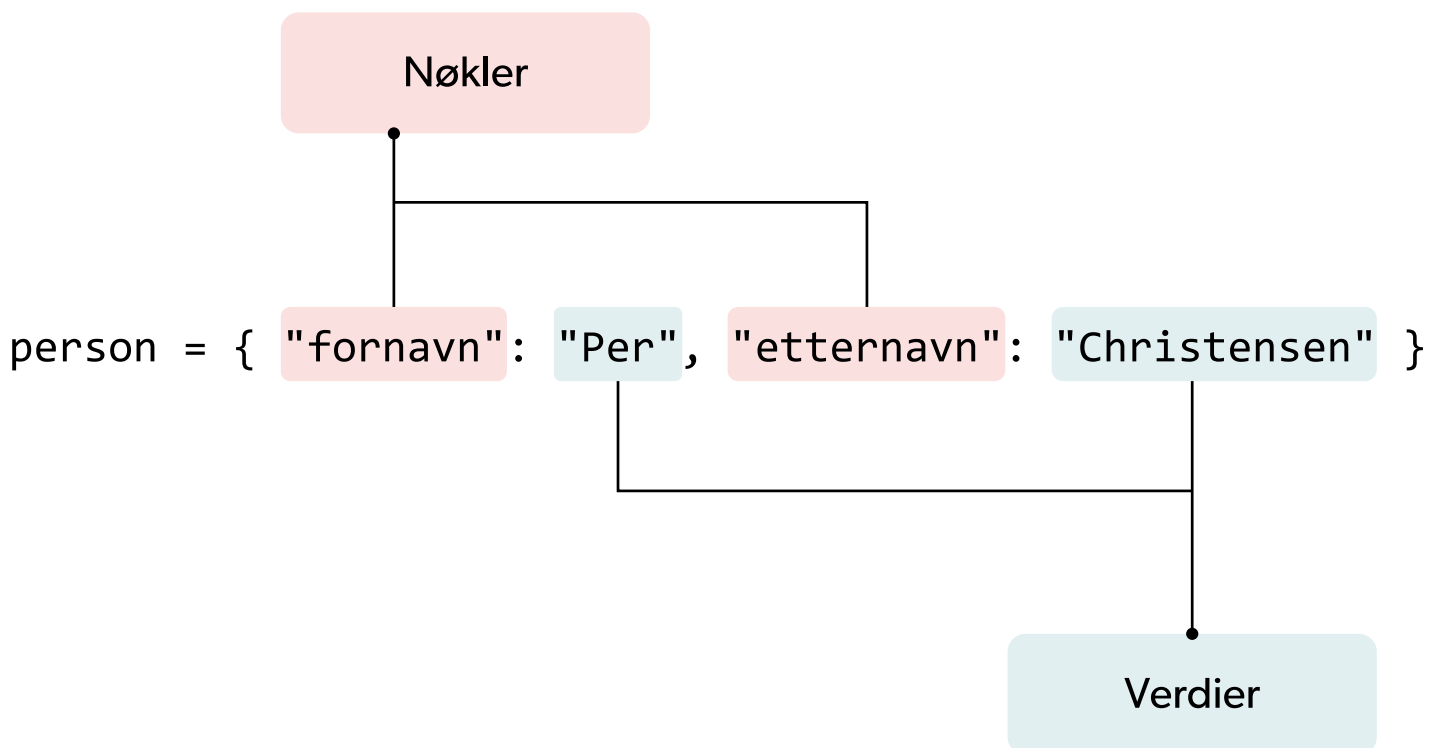
1E ORDBØKER

Introduksjon

Se for deg at du vil lagre en del informasjon som hører naturlig sammen, for eksempel informasjon om en person. Da er kanskje ikke en liste best egnet, for der er det ingen sammenheng mellom verdiene. Det er ingenting som viser oss at flere av verdiene hører til én person. Det er heller ingen informasjon om *hva* de forskjellige verdiene er. For å lagre slik informasjon kan det være bedre å bruke en *ordbok* (eng. dictionary).

En ordbok fungerer på et vis som en samling av variabler som hører naturlig sammen. Vi kan se på et eksempel der vi lagrer en ordbok med informasjon om en person:

```
person = { "fornavn": "Per", "etternavn": "Christensen" }
```



Vi bruker sløyfeparenteser (`{` og `}`) for å lage en ordbok. Innholdet i ordboka består av par av *nøkler* og *verdier*. Vi bruker kolon mellom en nøkkel og en verdi og komma mellom hvert nøkkel-verdi-par. I eksemplet ovenfor er "fornavn" og "etternavn" eksempler på nøkler, mens

"Per" og "Christensen" er eksempler på verdier.

Her har vi brukt tekster som nøkler, men vi kunne også ha brukt tall.

Vi kan også dele opp nøkkel-verdi-parene over flere linjer. Det gjør ofte koden lettere å lese:

```
person = {  
    "fornavn": "Per",  
    "etternavn": "Christensen"  
}
```



Ordbøker egner seg svært godt til å lagre informasjon, spesielt når vi får store og kompliserte datasett.

Hente ut verdier

Vi henter ut en verdi fra en ordbok omtrent på samme måte som vi henter en verdi fra en liste. Forskjellen er at vi bruker en nøkkel for å hente ut en verdi, i stedet for en indeks. Koden

```
print(person["fornavn"])
```



vil for eksempel gi utskriften "Per".

Skrivemåten minner om den vi bruker med lister, men her angir vi nøkkelen vi er ute etter, i stedet for en indeks.

Hvis vi vil skrive ut noe mer rundt en slik verdi fra en ordbok, må vi ta hensyn til at vi bruker anførselstegn. Fra «1A Tall og tekster» husker du kanskje at denne koden vil gi problemer:

```
print(f"Personens fornavn er {person["fornavn"]}." )
```



En enkel løsning i denne situasjonen er å bruke apostrofer rundt nøkkelen:

```
print(f"Personens fornavn er {person['fornavn']}.")
```



Legge til nye nøkler og verdier

Hvis vi ønsker å legge til mer informasjon i en ordbok, kan vi angi en verdi sammen med en ny nøkkel som vi vil legge til:

```
person["foedselsaar"] = 1972
```



Hvis vi nå skriver ut hele ordboka med `print(person)`, får vi denne utskriften:

```
{'fornavn': 'Per', 'etternavn': 'Christensen', 'foedselsaar': 1972}
```



Endre eller slette

Vi kan endre verdier i en ordbok ved å henvise til nøkkelen og så tilordne en ny verdi, slik:

```
person["foedselsaar"] = 1975
```



I dette tilfellet blir verdien som er knyttet til nøkkelen `"foedselsaar"`, endret til `1975` hvis nøkkelen finnes fra før. Hvis nøkkelen ikke finnes fra før, blir det laget en ny nøkkel med tilhørende verdi `1975`.

Vi kan slette nøkler med tilhørende verdier ved å bruke `pop()`:

```
person.pop("fornavn")
```



Da forsvinner nøkkelen `"fornavn"` med sin verdi (`"Per"`) fra ordboka.

Hvis vi prøver å fjerne en nøkkel som ikke finnes i ordboka, får vi en feilmelding. Det er derfor lurt å undersøke om nøkkelen faktisk finnes i ordboka, før vi fjerner den:

```
if "fornavn" in person:  
    person.pop("fornavn")
```



Oppgaver

- 1 Lag en ordbok som den i eksemplene ovenfor, men med informasjon om deg selv. Skriv ut informasjonen på en passende måte.
- 2 Lag en tom ordbok med dagens ukedag som navn. Gå inn på yr.no og finn stedet der du bor.
 - a Legg til nøkler og verdier i ordboka med dagens dato, dagens maksimumstemperatur, dagens minimumstemperatur, dagens nedbørsmengde og dagens gjennomsnittlige vindstyrke.
 - b Skriv ut en tekst med en kort værmelding om dagen i dag.
- 3 Lag en Python-ordbok som minner om en ekte ordbok, der du forklarer begrepene variabel, datatype, operator og løkke.
- 4 Vi kan bruke en ordbok til å lage en svært enkel krypteringsmaskin.
 - a Lag en ordbok som følger dette mønsteret:

```
krypter = { "a": "c", "b": "d", "c": "e", ... }
```

 (Alle nøklene gir en bokstav som ligger to bokstaver etter i alfabetet.) For bokstavene ø og å kan du bruke verdiene "a" og "b".
 - b Lag en tekstvariabel med en tekst du ønsker å kryptere.
 - c Gå gjennom hver av bokstavene i teksten som skal krypteres, og gjør den om ved hjelp av ordboka du laget i a.
 - d Lag en tilsvarende «dekrypteringsordbok» som kan gi den originale teksten fra den krypterte teksten.