

IN1000 – seminarartime uke 3

Gruppe 14 – Seminarrom Logo

Opprop og div. info

- Er du syk eller har luftveissymptomer? Hold deg hjemme! Finnes en haug med digitale tilbud på IFI
- Kontaktinfo: ahmedws@uio.no, eller huk tak i meg utenom timene
- Spørsmål til pensum, eller problemer dere har møtt på?

Agenda

- Lister
- Nøstede lister
- Mengder (sets i Python)
- Mapper/ordbøger/dictionary

Hva er en liste? Hva slags ting gjør du med en liste til vanlig?

- Diskuter i 1 min, og prøv å se koblingen mot lister i den virkelige verden mot lister i programmering

Lister

- Vi ønsker å lagre flere verdier!
- Dynamisk størrelse (i enkelte språk er lister statiske)
- Lister kan holde rede på ulike elementer og forskjellige datatyper
- Indekser er viktige
- «Slice lister», reversere en liste ganske raskt
- Listeoperasjoner

Oppgave 1

Hvor mange elementer er det i listene?

```
liste = [0]
```

```
liste1 = [1, 2, 3]
```

```
liste2 = [3, 3, 5, 7]
```

```
liste3 = ["A", "BC", "D", "E", "F"]
```

```
liste4 = ["mange elementer i denne listen"]
```

```
liste5 = []
```

Løsning

liste = [0] # 1

liste1 = [1, 2, 3] # 3

liste2 = [3, 3, 5, 7] # 4

liste3 = ["A", "BC", "D", "E", "F"] # 5

liste4 = ["mange elementer..."] # 1

Oppgave 1

Hvor mange elementer er det i listene?

```
liste = [0]
```

```
liste1 = [1, 2, 3]
```

```
liste2 = [3, 3, 5, 7]
```

```
liste3 = ["A", "BC", "D", "E", "F"]
```

```
liste4 = ["mange elementer i denne listen"]
```

```
liste5 = []
```

Oppgave 2

Ta utgangspunkt i listene over.

2.1 Hva ligger på indexene?

liste3 [4]

liste1 [0]

liste4 [1]

liste3 [5]

2.2 På hvilken index ligger:

5 (i liste2):

"BC" (i liste3):

1 (i liste1):

"E" (i liste3):

2.3 Tegn opp hvordan liste3 ser ut.

Løsning

2.1

liste3 [4] > "F"

liste1 [0] > 1

liste4 [1] > "mange elementer i denne listen"

liste3 [5] > ingenting, er kun 5 elementer, på indeksene 0-4.

2.2

5 (i liste2): liste2 [2]

"BC" (i liste3): liste3 [1]

1 (i liste1): liste1 [0]

"E" (i liste3): liste3[3]

2.3

Index element

Index	element
0	"A"
1	"BC"
2	"D"
3	"E"
4	"F"

Oppgave 3

Skriv en liste "navn" hvor du legger inn 3 navn; "Ola", "Martin" og "Selma".
Legg deretter til "Kari" nytt navn bakerst i listen.

Tegn opp hvordan listen ser ut nå.

|

Skriv ut Martin sitt navn ved hjelp av indeksen det ligger på.
Legg til to til navn først i listen, navnene velger du selv.

NB! Husk at når man legger noe inn på en indeks som er i bruk flyttes det som ligger der og alle etterfølgende elementer en indeks opp..

Tegn opp hvordan listen ser ut nå.

Fjern "Martin" fra listen.

Skriv ut alle navnene på listen.

Bonus: hva blir utskriften?

Oppgave 3

Skriv en liste "navn" med 3 navn; "Ola", "Martin" og "Selma".

Legg deretter til "Kari" nytt navn bakerst i listen.

Tegn opp hvordan listen ser ut nå.

Skriv ut Martin sitt navn ved hjelp av indeksen det ligger på.

Legg til to til navn forrest i listen, navnene velger du selv.

Tegn opp hvordan listen ser ut nå.

Løsning

```
navn = ["Ola", "Martin", "Selma"]  
navn.append("Kari")
```

0	"Ola"
1	"Martin"
2	"Selma"
3	"Kari"

```
print(navn[1])
```

Fjern "Martin" fra listen.

Skriv ut alle navnene på listen.

Bonus: hva blir utskriften?

```
navn.insert(0, "Lise")  
navn.insert(0, "Kai")
```

0	"Kai"
1	"Lise"
2	"Ola"
3	"Martin"
4	"Selma"
5	"Kari"

```
navn.remove("Martin")  
# alternativt navn.pop(3)
```

```
print("Alle navn: ", navn)
```

```
> Alle navn: ["Kai", "Lise", "Ola", "Selma",  
"Kari"]
```

Nøstede lister

- Lister i lister
- Kan være nyttig dersom vi ønsker å samle all data på ett sted
- 2D-matriser
- Kan kombinere eksisterende lister til en felles liste
- Livekoding, aksessering av elementer

Oppgave 4

Gitt at vi har de tre listene

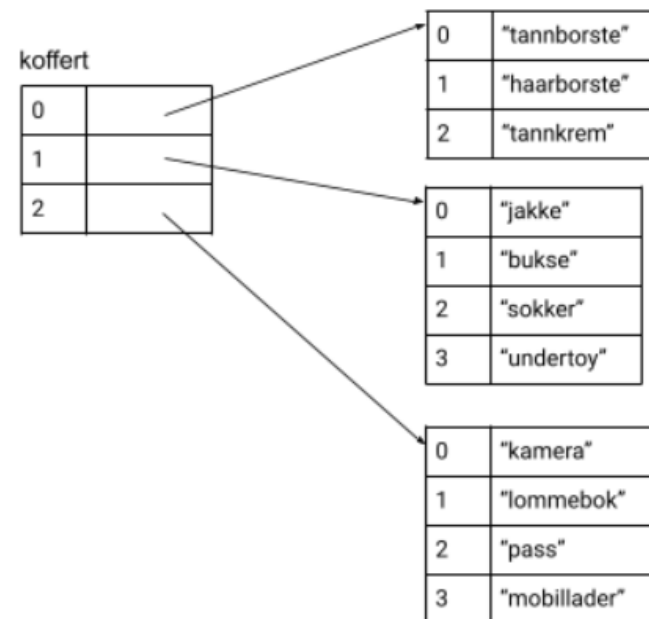
```
toalettsaker = ["tannborste", "haarborste", "tannkrem"]  
klaer = ["jakke", "bukse", "sokker", "undertoy"]  
div = ["kamera", "lommebok", "pass", "mobillader"]
```

Vi legger dem inn i listen "koffert", hvordan vil utskriften av denne se ut? Og hvordan kan vi tegne opp denne (prøv deg fram)?

```
koffert = [toalettsaker, klaer, div]  
print(koffert)
```

Løsning

```
> [ ['tannborste', 'haarborste', 'tannkrem'], ['jakke',  
'bukse', 'sokker', 'undertoy'], ['kamera', 'lommebok',  
'pass', 'mobillader'] ]
```



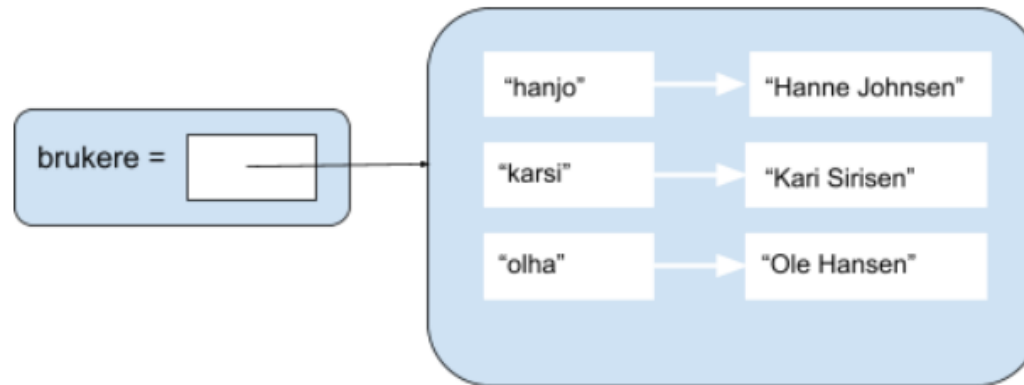
Litt om mengder

- Ingen spesifikk rekkefølge, altså ingen indeksering
- Syntaks: mengde = {element1, element2, element3}
- Tegning
- Fordel: raskere operasjoner enn lister
- Livekoding av hvordan man kan sammenligne mengder, kontra lister
- NB: en tom mengde defineres slik:
 - mengde = set(), IKKE mengde = {}

Ordbøker (dictionaries)

- Beholder som holder på par av nøkler og verdier (nøkkel-verdi)
- En meget kraftig datastruktur
- Ingen spesifikk rekkefølge (et lite unntak..)
- Fremfor indeksering, benyttes nøklene for å hente ut verdier
- Livekoding (legge til elementer, fjerne, overskrive, sjekke om noe er inni)

Oppgave 5, gitt følgende ordbok:



5.1 Definer en ordbok/dictionary som heter brukere, hvor du legger inn følgende par av nøkler og verdier (brukernavn som nøkler og navn som verdi).

5.2 karsi har registrert seg med feil navn, mellomnavnet hennes mangler, endre navnet hennes til "Kari Marie Sirisen".

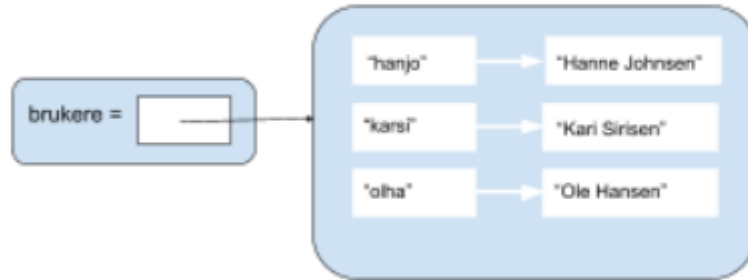
5.3 register deg selv som bruker med et brukernavn og navn.

5.4 Skriv ut hanjo sitt navn.

5.5 Fjern Hanne Johansen fra ordboken.

Oppgave 5

Gitt følgende ordbok/dictionary:



5.1 Definer en ordbok/dictionary som heter brukere, hvor du legger inn følgende par av nøkler og verdier (brukernavn som nøkler og navn som verdi).

Løsning

5.1

```
brukere = {"hanjo" : "Hanne Johansen",  
          "karsi" : "Kari Sirisen", "olha" : "Ole  
Hansen"}
```

5.2

```
brukere["karsi"] = "Kari Marie Sirisen"
```

5.3

```
brukere["kristijb"] = "Kristine Berge"
```

5.4

```
print (brukere["hanjo"])
```

5.5

```
brukere.pop("hanjo")
```