Gruppetime uke 3
IN1000

Plan for timen

- Enkle lister
- Nøstede lister
- Mengder
- Ordbøker
- Egenjobbing

Oppvarming

- Hva er en liste? Hva slags ting gjør du med for eksempel en handleliste eller gjøremålsliste?

Enkle lister

Hva er en liste

- Hittil har vi laget variabler med kun en verdi
- Med lister kan man lagre mer enn en verdi i samme variabel
- Dynamisk størrelse
  - Kan legge til eller ta vekk elementer (verdier) fra listen
  - Et element i listen kan være for eksempel heltall (Integer eller int) eller tekst (String eller str)
- Eksempler:
  - tallListe = [2, 4, 6, 7]
  - navneListe = ["Anne", "Per", "Lisa"]
  - tomListe = []

Oppgave

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"]
```

Løsning

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
liste5 = []
# 0/tom
```

Indeksering i lister

- Hvert element i en liste har en indeks
  - Første element i listen har indeks 0 - altså man starter å telle fra 0 og ikke 1
  - Andre element i listen har indeks 1, tredje indeks 2 osv
- Hente ut en verdi fra en liste basert på indeks
  - Eksempel:
    - minListe = [2, 8, -1]
    - minListe[0] vil gi 2
    - minListe[1] vil gi 8
    - minListe[2] vil gi -1

Oppgave

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.

```
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

Løsning

2.1

liste3 [4] > "F"

liste1 [0] > 1

liste4 [1] > ingenting

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

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

Nyttige listeoperasjoner

```
len(<liste>)                    # gir lengden på listen
liste.insert(index, element)     # sette inn element på angitt indeks
liste.append(element)           # setter inn nytt element på slutten av listen
liste.remove(element)           # fjerne angitt element fra listen
liste.pop(index)                # fjerner og returnerer elem. på angitt indeks
```

Oppgave

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.

Fjern “Martin” fra listen.

Skriv ut alle navnene på listen.

Bonus: hva blir utskriften?

Løsning

Løsning

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

0	“Ola”
1	“Martin”
2	“Selma”
3	“Kari”

```
print(navn[1])

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

Hva er nøstede lister?

- Eksempel: En koffert
  - toalettsaker = ["tannborste", "haarborste", "tannkrem"]
  - klaer = ["jakke,", "bukse", "sokker", "undertoy"]
  - div = ["kamera", "lommebok", "pass", "mobillader"]
- Og alt dette ligger i en koffert...
  - koffert = [toalettsaker, klaer, div]

Oppgave

Oppgave 4

Vi legger dem inn i listen “koffert”, hvordan vil utskriften av denne se ut? Og hvordan kan vi tegne opp denne?

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

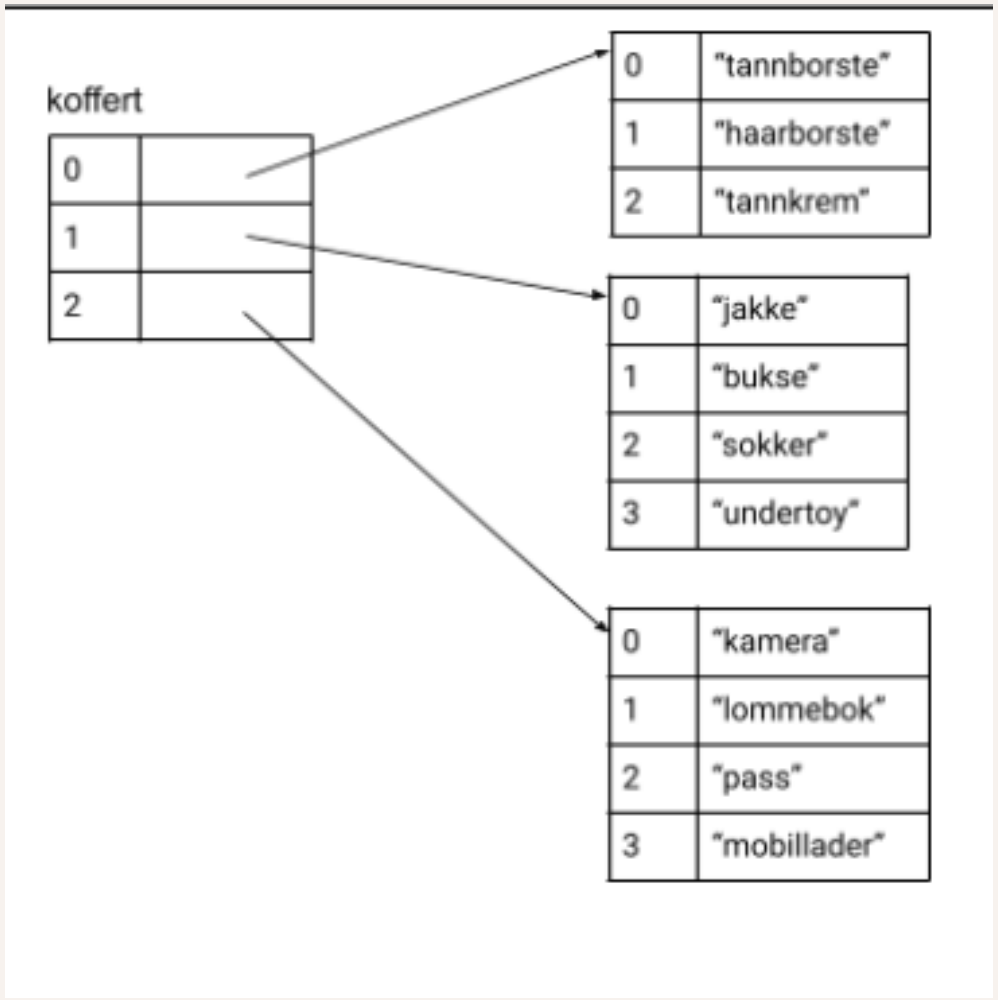
Løsning

Løsning

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

Hvordan få tak I et element fra en nøstet liste

- koffert[0] gir oss mulighet til å søke I listen som ligger på indeks 0 i koffert, altså toalettsaker
- koffert[0][1] gir oss tilgang til elementet på indeks 1 inne i tolettsaker
- Eksempel:
  - koffert[0][2] vil gi "tannkrem"
  - koffert[2][1] vil gi "lommebok"
  - koffert[1][3] vil gi "undertoy"



Mengder

Hva er en mengde i Python?

- Definisjon: En mengde er en samling av elementer der innbyrdes rekkefølge og antall forekomster av hvert element ignoreres
- Altså det finnes ingen spesifikk rekkefølge på elementene og ingen indeksering – i motsetning til lister
- I Python kalles mengder for set
- Syntaks:
  - mengde = {elemnt1, element2, element3}
- Eksempel:
  - navn = {"Kari", "Per", "Ola"}

Hvorfor bruke mengder?

mengder/set	**lister**
navn = {“Kari”, “Per”, “Ola”} navn2 = {“Per”, “Ola”, “Kari”} if navn == navn2 : …. #evalueres til **sant** fordi begge inneholder de samme elementene, sett er uavhengig av rekkefølgen.	navn = [“Kari”, “Per”, “Ola”] navn2 = [“Per”, “Ola”, “Kari”] if navn == navn2 : …. #evalueres til **usant** selv om begge inneholder de samme elementene, så er de i ulik rekkefølge og dermed ikke like lister.

- Operasjoner blir utført raskere enn på lister
- Kan brukes til å fjerne doble forekomster av elementer
- Vi kan enkelt sjekke om innholdet til to sett er like hvis vi ikke bryr oss om **rekkefølgen**

Nyttige operasjoner man kan bruke på set (mengder)

- Konvertere liste til mengde:
  - navn = ["Kari", "Per", "Ola"]
  - navn_mengde = set(navn)
- Andre operasjoner:
  - mengde.add(element) - legge til nytt element
  - mengde.discard(element) - fjerner element, gjør ingentin dersom elementet ikke finnes i mengden
  - mengde.remove(element) - fjerner element, får ett unntak dersom elementet ikke finnes i mengden
  - mengde.clear() - fjerner alle elementer i mengden
  - len(mengde) - finnes antall elementer i mengden

Ordbøker/dictionary

Hva er en ordbok i Python

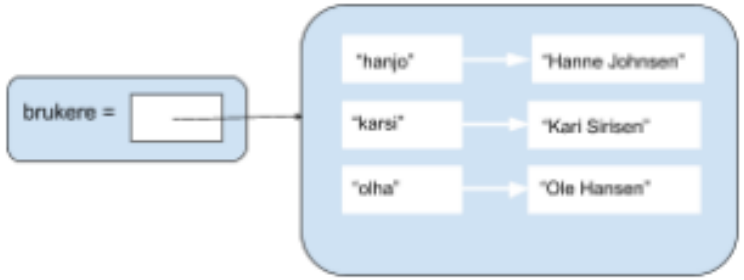
- Definisjon: En ordbok er en beholder som holder på par av nøkler og verdier. Hver nøkkel i ordboken er knyttet til en verdi.
- Syntaks:
  - ordbok = {nøkkel: verdi, nøkkel: verdi2}
- Eksempel:
  - kontakter = {"Kari": 46272839, "Ola": 84927284}

Bruk av ordbøker

- Brukes når man vil knytte to ting sammen, som et navn og et telefonnummer
- I stede for å bruke indeksering som lister, brukes nøkler til å hente ut verdier
- Nyttige operasjoner for ordbøker:
  - ordbok[nokkel] = verdi – legge til nytt element i ordboken ELLER endre på eksisterende
    - Eksempel:
      - Kontakter["Per"] = 74823984 – dersom "Per" allerede finnes i ordboken vil verdien til Per endres, hvis ikke vil Per legges til i ordboken
  - ordbok.pop(nokkel) - fjerne nøkkelen og verdien
    - Eksempel:
      - kontaker.pop("Per")
  - print(ordbok[nokkel]) - printer ut verdien til nøkkelen som er gitt

Oppgave

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).

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.

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.remove("hanjo")
```

Løsning

Jobbe med innlevering/Trix

- Ukens pensum:
  - Trix oppgave 03.01 - 03.16 https://trix.ifi.uio.no/course/8?tags=uke3
  - Innlevering 3 – ikke obligatorisk, men sterkt anbefalt

