

UKE 3 SEMINAR

LISTER, MENGDER OG ORDBØKER

LISTER

```
tallListe = [2, 4, 6, 7]
```

```
navneListe = ["Anne", "Per", "Lisa"]
```

```
tomListe = []
```

```
nøstetListe = [[1, 2], [3, 4], [5, 6]]
```

Lister er dynamiske, du kan endre størrelse på dem ved å legge til eller fjerne et element.

JOBBE TO OG TO

Lag et program som bruker alle operasjonene på lister som dere finner på neste slide.

OPERASJONER PÅ LISTER

<code>len(liste)</code>	<code># gir lengden på listen</code>
<code>liste.insert(index, element)</code>	<code># sette inn element på angitt indeks</code>
<code>liste.append(element)</code>	<code># setter inn nytt element på slutten av listen</code>
<code>liste.remove(element)</code>	<code># fjerner angitt element fra listen</code>
<code>liste.pop(index)</code>	<code># fjerner (og returnerer) element på angitt indeks (liste.pop(-1): sletter alltid det siste elementet(!))</code>
<code>print("Liste:", liste)</code>	<code># skriver ut> Liste: 1, 2, 3</code>

LØS HVER FOR DERE, DISKUTER ETTERPÅ

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 = []
```

LØS HVER FOR DERE, DISKUTER ETTERPÅ

Hva ligger på indexene?

- liste3 [4]
- liste1 [0]
- liste4 [1]
- liste3 [5]

På hvilken index ligger:

- 5 (i liste2):
- "BC" (i liste3) :
- 1 (i liste1):
- "E" (i liste3):

```
liste = [0]
```

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

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

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

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

```
liste5 = []
```

LØS HVER FOR DERE, DISKUTER ETTERPÅ

Vi har en mindre bag med “toalettsaker”, vår første liste:

- toalettsaker = [“tannborste”, “haarborste”, “tannkrem”]

I tillegg har vi en mindre bag med klær:

- klaer = [“jakke”, “bukse”, “sokker”, “undertoy”]

Og til slutt en liten bag med diverse:

- div = [“kamera”, “lommebok”, “pass”, “mobillader”]

For å samle alle småbaggene når vi skal på tur legger vi dem i en koffert:

- koffert = [toalettsaker, klaer, div]

Hva printes her?

- print(koffert)

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

LØS HVER FOR DERE, DISKUTER ETTERPÅ

Eksempel:

koffert [0] [2] = «tannkrem»

koffert [2] [1] = «lommebok»

Hva er disse?

koffert [1] [1]

koffert [2] [0]

koffert [0] [0]

Hvordan henter man disse?

“mobillader”

“sokker”

“haarborste”

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


MENGDER (SET)

Rekkefølge og antall forekomster er irrelevant.

Fordi rekkefølge og antall forekomster ikke betyr noe får vi at:

$\{1, 2, 3\} == \{1, 3, 2\} == \{1, 1, 1, 3, 3, 3, 2, 2\} == \{3, 2, 1\}$

mengder/set

```
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.

lister

```
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.

JOBBE TO OG TO

Lag et program som bruker alle operasjonene på mengder som dere finner på neste slide.

OPERASJONER PÅ MENGDER

```
ny_mengde = set()           # ny mengde
mengde_av_liste = set(liste)

menge.add(element)         # legge til nytt element i mengden

mengde.discard(element)    # fjerner element fra mengden, gjør ingenting dersom elementet ikke finnes i mengden.

mengde.remove(element)     # fjerner element fra mengden

mengde.clear()             # fjerner alle elementer i mengden, gjør mengden tom.

len(mengde)                # finner antall elementer i mengden.
```

ORDBØKER (DICTIONARY)

Tenk på en vanlig ordbok, typ norsk-engelsk-ordbok.

Har en nøkkel og en verdi. F.eks. hund: dog

ordbok = {nøkkel : verdi, nøkkel2 : verdi2,}

Kan fylles med alt mulig:

quizSpm = {"Er en banan en frukt eller grønnsak (frukt/grønnsak)" : "frukt", "Flyter bananer i vann? (ja/nei)" : "ja"}

JOBBE TO OG TO

Lag et program som bruker alle operasjonene på ordbøker som dere finner på neste slide.

OPERASJONER PÅ ORDBØKER

tomOrdbok = {} # ny ordbok

ordbok[nokkel] = verdi # legge til nytt element i ordboken ELLER endre på eksisterende.

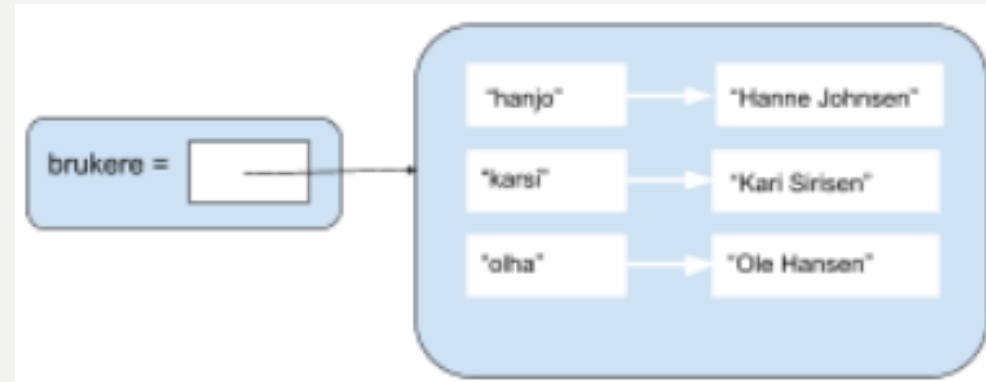
Eks: kontakter["Per"] = 45667990 # dersom "Per" allerede finnes i ordboken vil verdien til per endres.

ordbok.pop(nøkkel) # fjerne nøkkelen og verdien, og returnerer verdien

Eks: kontakter.pop("Per")

LØS HVER FOR DERE, DISKUTER ETTERPÅ

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



- karsi har registrert seg med feil navn, mellomnavnet hennes mangler, endre navnet hennes til "Kari Marie Sirisen".
- Register deg selv som bruker med et brukernavn og navn.
- Skriv ut hanjo sitt navn.
- Fjern Hanne Johansen fra ordboken.

JOBBE MED OPPGAVER