

Velkommen

IN 1000, Gruppe 16

Hvem skal jeg kontakte?

Spørsmål om faget:

kmbjorem@uio.no(meg) eller foreleser.

Studieinfo: spørsmål, klager, utsettelse av frister

<https://www.mn.uio.no/ifi/studier/kontakt/>

UiO forvei: bekymringer, noen å snakke med

<https://www.mn.uio.no/studier/forvei/>

SiO: fysisk og psykisk helsehjelp

<https://www.sio.no/helse>

<https://www.sio.no/helse/noen-%C3%A5-snakke-med>



Praktisk

- Gruppbytte:
 - Send mail til gruppelærer i den gruppe du vil starte i.
 - Vent på svar om det er ledig plass.
 - Gå på ny gruppe.
- Fredags-Python:
 - Frivilligt opplegg på fredager med gruppelærer tilstede.
-





Opprop!

Uke 3

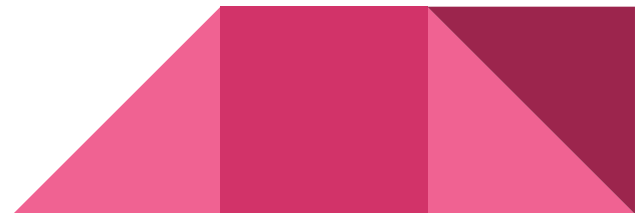
Lister

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

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

```
tomListe = []
```

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



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 til høyer..

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

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

```
liste = [0]
```

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

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

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

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

```
liste5 = []
```



Jobbe to og to

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



Operasjoner på lister

`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)` # fjerner angitt element fra listen

`liste.pop(index)` # fjerner (og returnerer) element på angitt indeks (`liste.pop(-1)`: sletter alltid det siste elementet(!))

`print("Liste: ", liste)` # skriver ut> Liste: 1, 2, 3



Oppgave som løses 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? skriv på ark!
 - `print(koffert)`

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

Oppgave løses hver for dere, diskuter etterpå

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/sets

Rekkefølge og antall forekomster er irrelevant.

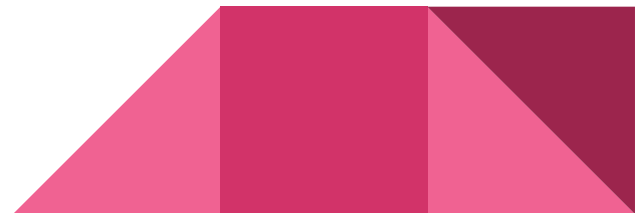
```
tomMengde = set()
```

```
navnMengde = {"Per", "Pål", "Espen"}
```

```
tallMengde = {1, 2, 3}
```

Da 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}
```



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()` eller `mengde_av_liste = set(liste)`

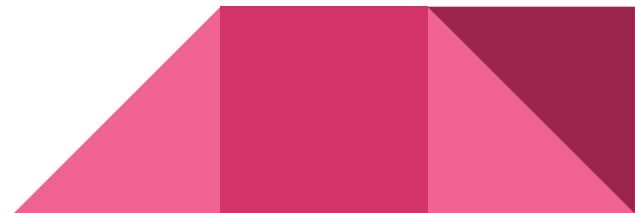
`menge.add(element)` # legge til nytt element i mengden

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

`mengde.remove(element)` # fjerner element fra listen

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

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



Ordbøker/dictionaries

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"}



Oppgave 5

Gitt følgende ordbok/dictionary:

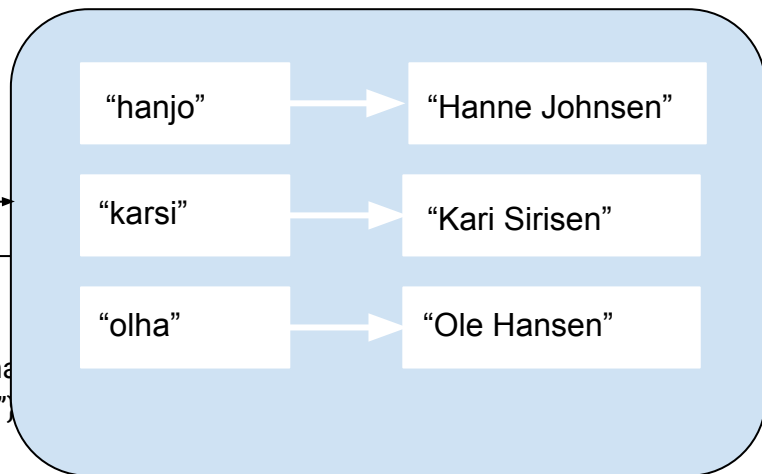
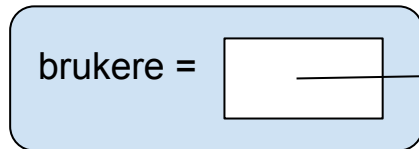
5.1 Definer en ordbok/dictionary som heter brukere, hvor du legger inn følgende par av nøkler og verdier (brukernavn som nøkler og navn som verdi).

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.

5.4 Fjern Hanne Johansen fra ordboken.



Løsning

5.1

```
brukere = {"hanjo": "Hanne Johnsen",  
           "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")
```

Jobbe to og to

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



Operasjoner på ordbøker/dictionaries

tomOrdbok = {}

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.

ordok.pop(nøkkel) # fjerne nøkkelen og verdien.

Eks: kontakter.pop("Per")



Diskuter hva som må til for å fikse denne koden

Oppgave 6

Her er et eksempel på en uleselig/uryddig kode, hva slags endringer ville du gjort med dem?

```
def f1() :  
    i = float(input("input"))  
    print(i*i)
```

```
def f2() :  
    i = input()  
    print("velkommen til in1000 ", i)
```

```
def f3( ) :  
    print ( "Hei!" )
```

```
f3()  
f2()  
f1()
```

Løsning

generelt dårlig funksjonsnavn og variabelnavn, lite beskrivende

metoden kalt "multipliser" > forklarer hva metoden gjør.

Variabelnavn feks kalt "tall" i stede for "i". Gi bedre beskjed i terminalen til bruker om hva input skal være (at det skal være ett tall).

Kalle metoden noe annet, feks. velkommen().

Variabelnavn kalt "navn". Ingen beskjed til bruker om hva man ønsker at input skal være.

Mange unødvendige mellomrom inni parantesen, disse syns ikke i utskriften uansett.

Jobb med oppgaver