

Gruppetime IN1000

Uke 3



Agenda

- Hvordan gikk det med innlevering 2?
- Innlevering 2: oppgave 3 og 4
- Temaer for denne uken:
 - Enkle lister
 - Nøstede lister
 - Mengder
 - Ordbøker
- Oppgaver om ukens temaer og gjennomgang

Oppgave 3: Kodeflyt

```
def print_prosa():
    print("Melding til alle gaardeiere:")
    print("Antall dyr paa gaarden: ")

antall_dyr = 4
print_prosa()
print(antall_dyr)
antall_nye_dyr = int(input("Hvor mange nye dyr kommer til gaarden: "))
antall_dyr = antall_dyr + antall_nye_dyr
print_prosa()
print(antall_dyr)

if antall_dyr > 12:
    print("Det er mer enn ett dusin dyr paa gaarden!")
elif antall_dyr == 12:
    print("Det er ett dusin dyr paa gaarden!")
else:
    print("Det er mindre enn ett dusin dyr paa gaarden!")
```

Oppgave 4: Feil

```
a = input("Tast inn et heltall! ")  
b = int(a)  
if b < 10:  
    print (b + "Hei!")
```

Hva er en liste?

Diskuter!

Lister i Python

- Flere verdier i samme variabel!
- “Størrelsen” på listen endrer seg automatisk når du legger til og sletter elementer
 - “Dynamisk” størrelse
- Kan ha lister med forskjellige elementer:
 - `tallListe = [2, 4, 6, 7]`
 - `navneListe = ["Anne", "Per", "Lisa"]`
 - `tomListe = []`
- Kan ha elementer med forskjellige typer::
 - `liste = [1, "to", 3.0]`

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 1

Hvor mange elementer er det i listene?

liste = [0] # 1

liste1 = [1, 2, 3] # 3

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

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

liste4 = ["mange elementer i denne listen"] # 1

liste5 = [] # 0/tom

Indeksering

- Indeksering er måten listen er nummerert på
- I Python har det første elementet i listen indeks 0
- For å aksessere en indeks skriver vi liste[indeks]
- Eksempel:
 - liste = [1, 2, 3, 4]
 - print(liste[0]) # gir 1

Oppgave 2.1

Hva ligger på indexene?

liste3[4]

liste1[0]

liste4[1]

liste3[5]

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 2.1 løsningsforslag

Hva ligger på indexene?

```
liste3[4] # "F"
```

```
liste1[0] # 1
```

```
liste4[1] # feil
```

```
liste3[5] # feil
```

(viser med
livekoding)

```
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 2.2

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 = []
```

Oppgave 2.2 løsningsforslag

På hvilken index ligger:

5 (i liste2): # 2

“BC” (i liste3) : # 1

1 (i liste1): # 0

“E” (i liste3): # 3

```
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 2.3

2.3 Tegn opp hvordan liste3 ser ut (tabell med index | verdi).

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

Oppgave 2.3 løsningsforslag

2.3 Tegn opp hvordan liste3 ser ut (tabell med index | verdi).

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

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

Nyttige operasjoner på lister

<code>len(liste)</code>	# Gir lengde (antall elementer - int)
<code>liste.insert(index, element)</code>	# Setter inn element på gitt indeks
<code>liste.append(element)</code>	# Setter inn element på slutten av listen
<code>liste.remove(element)</code>	# Fjerner første forekomst av element fra listen
<code>liste.pop(index)</code>	# Fjerner og returnerer element på gitt indeks

CAT LIST!!!

```
cat_list = ["katt1", "katt2", "katt3", "katt4"]
```



“katt1”



“katt2”



“katt3”



“katt4”

CAT LIST!!!

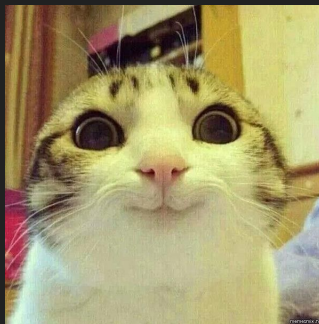
```
cat_list.pop(3)
```



“katt1”



“katt2”



“katt3”

CAT LIST!!!

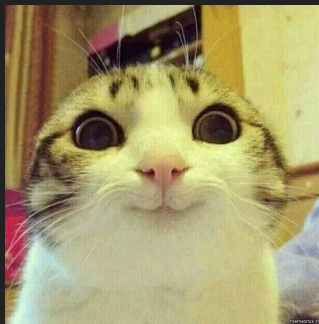
```
cat_list.append("katt5")
```



“katt1”



“katt2”



“katt3”



“katt5”

CAT LIST!!!

```
cat_list.remove("katt2")
```



“katt1”



“katt3”



“katt5”

CAT LIST!!!

```
cat_list.insert(1, "katt2")
```



"katt1"



"katt2"



"katt3"



"katt5"

CAT LIST!!!

`cat_list.pop(0)`



“katt2”



“katt3”



“katt5”

CAT LIST!!!

`cat_list.pop(0)`



“katt3”



“katt5”

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 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?

Oppgave 3 - Løsningsforslag

```
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

- Nøstede lister er lister som *inneholder andre lister*
- En nøstet liste kan også inneholde en annen nøstet liste
- Eksempel:
- `[[1, 2], [en, to]]`

Nøstede lister

Lister i lister

Eksempel (koffert):

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 de små baggene når vi skal på tur legger vi dem i en koffert.

```
koffert = [toalettsaker, klaer, div]
```

Oppgave 4

Vi legger dem inn i listen “koffert”, hvordan vil utskriften av denne se ut?

Og hvordan kan vi tegne opp denne?

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

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

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

```
koffert = [toalettsaker, klaer, div]
```

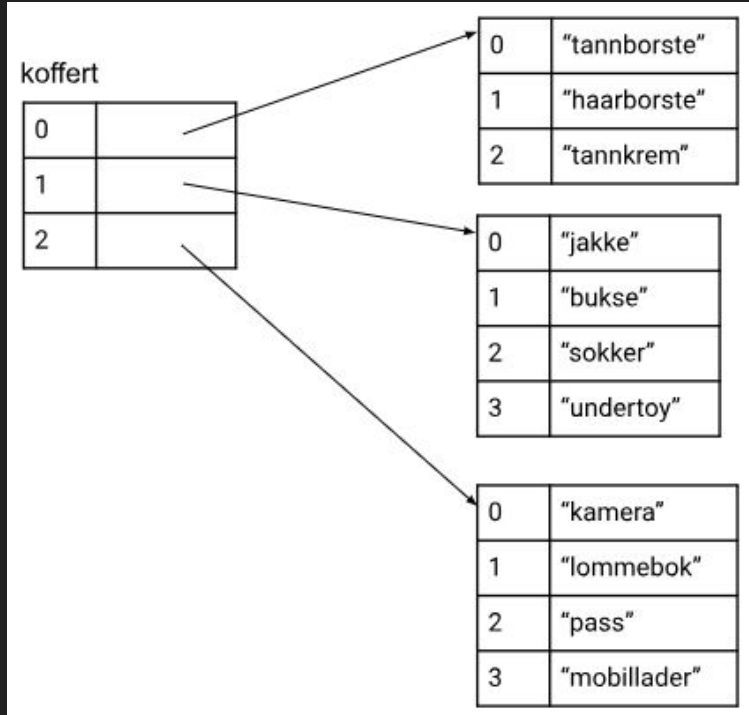
```
print(koffert)
```

Oppgave 4 - løsningsforslag

Utskrift:

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

Tegning:



Oppgave 4 - fortsettelse

Hvordan får vi så tak i ett element?

Hva er...

`koffert[1][1]`

`koffert[2][0]`

Hvordan “henter man frem”

“mobillader”

“sokker”

“haarborste”

Oppgave 4 - løsningsforslag

Hvordan får vi så tak i ett element?

koffert[0][2] > tannkrem

koffert[2][1] > lommebok

Hva er...

koffert[1][1]

“bukse”

koffert[2][0]

“kamera”

Hvordan “henter man frem”

“mobillader”

koffert[2][3]

“sokker”

koffert[1][2]

“haarborste”

koffert[0][1]

Mengder

Syntaks:

$\text{mengde} = \{\text{element}, \text{element2}, \text{element3}, \dots\}$

Eksempel:

$\text{navn} = \{\text{"Kari"}, \text{"Per"}, \text{"Ola"}\}$

- Mengder har ingen spesifikk rekkefølge på elementene
- Ingen indeksering
- Like elementer telles bare en gang

Mengder

Tom mengde lages slik:

```
mengde = set()
```

IKKE: mengde = {}

Nyttige operasjoner:

```
set(listenavn)           # konverterer en liste til en mengde
menge.add(element)       # legge til nytt element i mengden
mengde.discard(element)  # fjerner element fra listen #gir ingen unntak hvis
                           elementet ikke finnes
mengde.remove(element)   # fjerner element fra listen #gir unntak
mengde.clear()           # fjerner alle elementer i mengden
len(<mengde>)            # finner antall elementer i mengden
```

Minioppgaver

1. Hva er output her?

```
mengde = {1, 1, 1, 2}  
print(len(mengde))
```

2. Hva er output her?

```
mengde1 = {1, 1, 1, 2}
```

```
mengde2 = {1, 2, 2}
```

```
print(mengde1==mengde2)
```

3. Hva er output her?

```
mengde1 = {1, 1, 1, 2}
```

```
mengde1.remove(1)
```

```
print(len(mengde1))
```

Ordbok

Hva er en ordbok? Diskuter!

Ordbøker i Python

- En beholder som holder på *par av nøkler og verdier*
- Vi kan slå opp på en nøkkel og få tilbake verdien
- En nøkkel kan bare være knyttet til én verdi
- Forskjellige nøkler kan ha like verdier

Ordbøker: Syntaks

Definere en ordbok:

```
ordbok = {nøkkel1 : verdi1, nøkkel2 : verdi2 ...}
```

```
telefonbok = {"Kari" : 47302430, "Ola" :  
53243235, "Martin" : 99534532}
```

```
tomOrdbok = {}
```

Bruk:

- Vi slår opp som i lister, men i stedet for å bruke indeks bruker vi nøkkelen:
- ```
print("Kari sitt telefonnummer er",
telefonbok["Kari"])
```
- Husk at som mengder har ordbøker ingen definert "rekkefølge" på verdiene.

# Ordbøker : Noen nyttige operasjoner

`ordbok[nøkkel] = verdi`

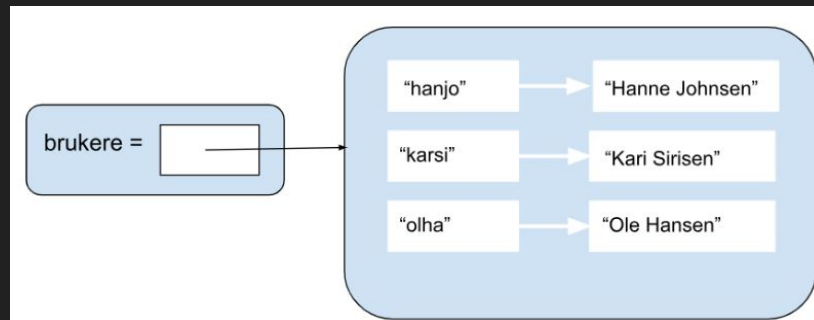
# Legger til ny verdi og nøkkel eller endrer på  
eksisterende verdi og nøkkel

`ordbok.pop(nøkkel)`

# fjerner nøkkel og verdi fra ordbok (dersom de  
eksisterer)

# 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

Registrer 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 - løsningsforslag

## 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")
```

# Neste uke!

While-løkker

For-løkker med teller

For-løkker på lister