

# Mer lister

Progintro dag 4

# Loope gjennom en liste

```
programmer = [  
    'Dagsrevyen',  
    'Fantorangen',  
    'Nytt på nytt',  
    'Fra bølge til bestevenn']
```

```
for program in programmer:  
    print(program)
```

# Kodeblokk og indentering

- Gjøre mer inni en løkke vs å gå ut av løkka
- Indenteringsfeil:
  - Glemme innrykk
  - Glemme å rykke inn alle linjene som skal med
    - Logiske feil
  - Unødvendig innrykk utenfor og inni for-løkka
- Glemme kolon

# Oppgaver

## 4.1 Pizza

- Tenk på 3 pizzaer du liker og lagre pizzanavnene i en liste. Bruk en for-løkke for å skrive ut hver pizza
- Gjør om for-løkka til å skrive ut en setning om hver pizza i stedet for å skrive ut bare navnet. Feks 'Jeg liker <pizzanavn>'
- Legg på en linje på slutten av programmet , utenfor for-løkka som uttrykker hvor mye du liker pizza, f.eks 'Jeg elsker virkelig pizza!'

# Oppgaver

## 4.2 Dyr

- Tenk på minst tre forskjellige dyr som har noe til felles. Lagre navnene i en lista og bruk en for-løkke til å skrive ut navnet på hvert dyr.
- Endre programmet så det skriver ut en setning om hvert dyr, f.eks 'En hund er et bra kjæledyr'
- Legg til en linje på slutten av programmet som sier hva disse dyrene har til felles. F.eks 'Alle disse dyrene er bra kjæledyr'

# Sekvenser av tall

```
for i in range(1, 5):  
    print(i)
```

- Halvåpent intervall: får med 1, men ikke 5

```
for i in range (2, 11, 2):  
    print(i)
```

- Det siste argumentet er hvor mye man hopper for hvert tall (default 1)

# Lage liste med tall

For å lage en faktisk liste med tall må man bruke list()

```
tall = list(range(1, 5))  
print(tall)
```

# Statistikk-funksjoner

```
tall = [1, 4, 5, 3, 9, 12, 34, 6, 44]
```

```
min(tall)
```

```
max(tall)
```

```
sum(tall)
```



# List comprehension

Kombinerer for-løkke og det å legge til elementer i en liste i en linje

Vanlig:

```
kvadrater = []  
for i in range(1, 11):  
    kvadrater.append(i**2)
```

List comprehension:

```
kvadrater = [i**2 for i in range(1, 11)]
```

# Oppgaver

**4.3 Telle til 20:** Lag en for-løkke som skriver ut alle tallene fra 1 **til og med** 20

**4.4 En million:** Lag en liste av tallene fra 1 til en million, og bruk deretter en for-løkke til å skrive ut tallene. (Om det tar lang tid før programmet blir ferdig, avbryt med ctrl-c)

**4.5 Summere en million:** Lag en liste av tallene fra 1 til en million. Bruk funksjonene min() og max() til å forsikre deg om at lista starter på 1 og slutter på en million. Gjør sum() og se hvor raskt Python kan legge sammen en million tall

**4.6 Oddetall:** Bruk tredje argument i range()-funksjonen til å lage en liste av oddetallene fra 1 til 20. Bruk en for-løkke til skrive ut tallene.

# Oppgaver

**4.7 Tretall:** Lag en liste av multipler av 3 fra 3 til 30. Bruk en for-løkke til å skrive ut tallene.

**4.8 Kubikktall:** F.eks er kubikktallet til 2,  $2 ** 3$ . Lag en liste av kubikktallene fra 1 til 10, og bruk en for-løkke til å skrive ut verdien av hver kubikk

**4.9 Kubikk comprehension:** Bruk list comprehension til å generere en liste av de første 10 kubikkene.

# Deler av en liste

```
kanaler = ['NRK1', 'NRK2', 'NRK3', 'NRKSUPER']
```

```
kanaler[0:2]
```

```
kanaler[1:]
```

```
kanaler[:3]
```

```
kanaler[:]
```

```
kanaler[-3: -2] # for spesielt interesserte
```

```
for kanal in kanaler[:2]:
```

```
    print(kanal)
```

# Kopiere en liste

Lage en helt ny liste basert på en annen.

```
kanaler = ['NRK1', 'NRK2', 'NRK3', 'NRKSUPER']
```

```
kanaler_extra = kanaler # Kopierer ikke lista - forteller kun at begge variabler peker til samme liste
```

```
kanaler_extra = kanaler[:]
```

```
kanaler.append('NRK4')
```

```
kanaler_extra.append('NRKW')
```

```
print(kanaler)
```

```
print(kanaler_extra)
```

# Oppgaver

**4.10** Ta utgangspunkt i koden du har skrevet tidligere i kapittelet, legg til noen linjer på slutten og gjør følgende:

- Print “*De tre første elementene i lista er:*” - så benytter du *slice* for å printe ut de tre første elementene i lista.
- Print “*Tre av de midterste elementene i lista er:*” - benytt *slice* for å printe ut tre elementene fra midten av lista.
- Print “*De tre siste elementene i lista er:*” - benytt *slice* for å printe ut de tre siste elementene i lista.

# Oppgaver

**4.11** Mine pizzaer, dine pizzaer: Benytt koden fra oppgave 4.1 (side 56). Lag en kopi av lista med pizzaer og kall den “friends\_pizzas”. Gjør deretter følgende:

- Legg til en ny pizza i den originale lista
- Legg til en annen pizza i lista *friends\_pizza*.
- Bevis at du har to separate lister. Print ut “*Mine favorittpizzaer er:*” - så benytt en for-løkke for å printe ut den originale lista.

Print ut “*Min venns favorittpizzaer er:*” - så benytt en for-løkke for å printe ut den andre lista. Sørg for at hver nye pizza blir lagret i de tilegnede listene.

# Tupler

- Lister man ikke kan endre på (immutable)
- () i stedet for []

```
kanaler = ('NRK1', 'NRK2', 'NRK3')  
print(kanaler[1])
```

```
kanaler[1] = 'NRK1+' # gir feil
```

```
kanaler = ('NRK1', 'NRK1+', 'NRK3') # er lov
```



# Oppgaver

## 4.13 Buffet: Lag et tuppel med 5 typiske buffet-retter

- Lag en for-løkke som går gjennom og skriver ut hver rett
- Prøve å endre noe i tuppelet og sjekk at Python protesterer
- Bytt ut to av mattypene med noe annet ved å overskrive tuppelet.
- Lag en ny for-løkke og skriv ut rettene. Se at det er de nye rettene som skrives ut

# Jobb med og diskuter oppgavene fra Github

<https://github.com/nrkno/prog-intro/blob/master/Uke4/uke4.pdf>