

Løkker og lister

Med Einar og Malin

For-løkker

Gjør “noe” for elementene i en liste. Nyttig hvis du har en lang liste eller ikke vet hvor mange elementer du har i lista.

```
for <element> in <liste>:  
    <gjør noe med element>
```

Eksempel:

```
kanaler = ["nrk1", "nrk2", "nrk3", "nrk4", "nrk5", "nrk6"]  
for kanal in kanaler:  
    print(f"{kanal} er en av NRKs kanaler")
```

Indentering

Skiller det som er inne i løkka fra det som er utenfor ved at man bruker “tab” for å sette en kodeblokk et innrykk inn.

```
kanaler = ["nrk1", "nrk2", "nrk3", "nrk4", "nrk5", "nrk6"]
for kanal in kanaler:
    print(f"{kanal} er en av NRKs kanaler")
    print("1. Denne print-setningen blir printet ut flere ganger")
print("2. Denne print-setningen blir printet ut kun en gang")
```

Oppgaveløsning side 56

4.1 - Pizzaer.

- a) Lag en liste over tre av dine favorittpizzaer.
- b) Print ut hver av pizzaene ved hjelp av en for-løkke.

Eksempel på utskrift:

```
mozzarella  
tunfisk  
skinke
```

- c) Endre fra å kun printe ut pizzaene til å printe ut “Jeg liker godt pizza med <pizza>”.

Eksempel på utskrift:

```
Jeg liker godt pizza med mozzarella  
Jeg liker godt pizza med tunfisk  
Jeg liker godt pizza med skinke
```

- d) Skriv ut en linje på slutten av programmet som skriver: “Jeg digger pizza!”

Eksempel på utskrift:

```
Jeg liker godt pizza med mozzarella  
Jeg liker godt pizza med tunfisk  
Jeg liker godt pizza med skinke  
Jeg digger pizza!
```

Bruk av range()

Funksjonen range() lager en liste med tall. Går fra og med første tall, og til (ikke inkludert) siste tall.

```
liste_med_tall = list(range(<start>, <slutt>))
```

Eksempel som vil en liste med elementene 1, 2, 3 og 4:

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

Tar også et valgfritt tredje parameter som angir steget mellom hvert tall.

```
liste_med_tall = list(range(<start>, <slutt>, <intervall>))
```

Eksempel som vil en liste med elementene 1, 3, 5, 7 og 9:

```
tall = list(range(1,10,2))
```

Eksempel på bruk av range i for-løkker:

```
for tall in range(1,5):  
    print(f"Tallet er {tall}")
```

Enkel statistikk med lister av tall

Summerer alle tallene i lista

```
sum(<liste>)
```

Finner det største tallet i en liste

```
max(<liste>)
```

Finner det minste tallet i en liste

```
min(<liste>)
```

Eksempel:

```
utgifter_per_dag = [340, 44, 124, 360, 202, 79, 560]
```

```
totalt_forbruk = sum(utgifter_per_dag)
```

```
max_forbruk = max(utgifter_per_dag)
```

```
min_forbruk = min(utgifter_per_dag)
```

```
print(f"Totalt forbruk på en uke {totalt_forbruk}")
```

```
print(f"Maks forbruk på en dag {max_forbruk}")
```

```
print(f"Minimum forbruk på en dag {min_forbruk}")
```

Du kan også bruke vanlige matematiske operatører (+, -, *, /, **)

List comprehensions

Gjør koden mer kompakt og man kan gjøre flere ting på en linje.

Eksempel på å lage en liste av kvadrater uten list comprehensions:

```
kvadrater = []  
for verdi in range(1,10):  
    kvadrat = verdi**2  
    kvadrater.append(kvadrat)  
print(kvadrater)
```

Eksempel på å lage en liste av kvadrater med list comprehensions:

```
kvadrater = [verdi ** 2 for verdi in range(1,10)]  
print(kvadrater)
```

Oppgaver s. 60

4.3 Bruk en for-loop for å printe ut tallene 1 til 20, inkludert 20.

4.4 Lag en liste fra 1 til en million. Print ut alle tallene. Om det tar for lang tid, avslutt med CTRL+C

4.5 Sjekk at listen du lagde i 4.4. er riktig ved å print ut hva som er min() og max() av lista. Print også ut summen av alle tallene.

4.6 Bruk et tredje argument til range() for å lage en liste med alle oddetallene fra 1 til 20. Bruk en for-loop for å printe ut alle tallene.

4.8 Et nummer opphøyd i tredje er kalt en kube. F.eks. en kube av 2 skrives $2^{**}3$ i Python. Lag en liste av de ti første kubene (1 til 10) og bruk en for-loop for å printe dem.

4.9 Lag kuber av de ti første tallene, men bruk en list comprehension i stedet.

Slicing

Kopierer ut hele eller en del av en liste. Går fra og med start, og til slutt (inkluderer ikke siste indeks).

```
liste[<start>:<slutt>]
```

Eksempel med utgifter per dag hvor indeks 0 representerer mandag, og indeks 6 representerer søndag:

```
utgifter_per_dag = [340, 44, 124, 360, 202, 79, 560]
```

Hent ut verdier fra og med indeks 2 til indeks 5 (onsdag til fredag):

```
utgifter_per_dag[2:5]
```

Hent ut frem til indeks 5 i lista (mandag til fredag):

```
utgifter_per_dag[:5]
```

Hent ut fra indeks 5 fra lista (lørdag og søndag):

```
utgifter_per_dag[5:]
```

Hent ut de to siste dagene fra lista (lørdag og søndag):

```
utgifter_per_dag[-2:]
```

Kopiering av lister

En liste kan kopieres ved å enten ha to variabler som peker på den samme lista, eller ved å lage en liste som inneholder de samme verdiene, men er en egen liste.

```
original_liste = list(range(1,10))
```

```
liste1 = original_liste
```

```
liste2 = original_liste[:]
```



Tupler

Lister som ikke kan endres (ikke muterbart).

```
kanaler = ("nrk1", "nrk2", "nrk3")
```

Kodestil

Viktig hvis andre skal lese koden din.

Husk at kode leses oftere enn den skrives.

Lesbarhet er i mange tilfeller viktigere enn at koden skal være rask og kompakt.

Jobb med og diskuter oppgavene fra Github

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