



# Lister og indekser

## Introduksjon

Denne oppgaven handler om *lister*, altså å samle flere ting i en og sar sammen i Python, bør du se på [løkker i repetisjonsoppgaven](#) dersom c

## Hvordan lage lister?

Hver ting i en liste kalles et *element*. En liste lages ved å skrive ele

```
>>> lst = ['egg', 'ham', 'spam']  
>>> lst  
['egg', 'ham', 'spam']
```

Vi har nå ei liste som inneholder ordene `'egg'`, `'ham'` og `'spam'` lister:

```
>>> lst = []  
>>> lst  
[]
```

Ei liste kan inneholde alt mulig - tall, strenger og også andre lister:

```
>>> lst = [ 3, 'komma', [1, 4] ]  
>>> lst  
[3, 'komma', [1, 4]]
```

# Legge til og fjerne elementer

Hva om vi ønsker å legge til eller fjerne elementene fra lista vi har? Vi kan bruke `lst.append(elm)` og `lst.remove(elm)`, der `lst` er lista og `elm` er elementet. `lst.append(elm)` legger til `elm` på slutten av `lst`, slik som illustrert:

```
>>> lst = []
>>> lst.append('Per')
>>> lst
['Per']
>>> lst.append('Ada')
>>> lst.append('Kim')
>>> lst
['Per', 'Ada', 'Kim']
```

`lst.remove(elm)` sletter det første elementet `elm` fra `lst`. Det vil si at det bare er det første elementet lik `elm`:

```
>>> lst = ['Per', 'Ada', 'Kim', 'Ada']
>>> lst.remove("Ada")
>>> lst
['Per', 'Kim', 'Ada']
```

Vi skal nå lage et handleliste-program. Programmet skal be brukeren om å skrive inn en gjenstand når `ferdig` skrives inn. Programmet skal fungere slik:

```
>>>
Skriv inn en gjenstand: ost
Skriv inn en gjenstand: melk
Skriv inn en gjenstand: brød
Skriv inn en gjenstand: ferdig
```

ost  
melk  
brød

Dette må du gjøre:

- ☐ Lag ei tom liste.
- ☐ Be om input.
- ☐ Så lenge input ikke er lik **ferdig**, legg til det nye elementet i lis  
**Hint:** Hva slags løkke kan vi bruke her?
- ☐ Skriv ut hvert hvert element i lista.  
**Hint:** Hva slags løkke kan vi bruke her?

# Indekser

Tenk deg at vi har ei liste, og ønsker å hente ut det andre elementet i lista. Dette kalles *indeks*. Indeks er plassen til elementet og skrives mellom **[]**. Her er et eksempel på en liste med tall:

```
>>> lst = [1, 2, 3, 4, 5]
>>> lst[1]
2
```

Du la kanskje merke til at vi skrev **1**, men fikk ut det andre elementet. Dermed har det første elementet i lista indeks **0**, og det andre har indeks **1**. Du husker kanskje at det samme skjer når vi bruker **range()**?

```
>>> list(range(5))
```

```
[0, 1, 2, 3, 4]
```

Til nå har vi brukt `for element in lst` for å gå igjennom element praktisk å telle hvor langt vi er kommet i lista. Til dette kan vi bruk indeksen:

```
>>> lst = ['Per', 'Kim', 'Ada']  
>>> for i, value in enumerate(lst):  
    print(i, value)
```

```
0 Per  
1 Kim  
2 Ada
```

I eksempelet over får `i` verdien av indeksen, og `value` får verdier løkke, men vi får indeksen i tillegg.

Modifiser nå programmet fra forrige oppgave til å skrive ut indekser v fungere:

```
>>>  
Skriv inn en gjenstand: ost  
Skriv inn en gjenstand: melk  
Skriv inn en gjenstand: brød  
Skriv inn en gjenstand: ferdig  
0 ost  
1 melk  
2 brød
```

Dette må du gjøre:

- ☐ Bruk programmet fra oppgaven over.
- ☐ Bruk `enumerate` til for å få indeksen til hver element.
- ☐ Skriv ut indeksen på samme linje som elementet i lista.

# Indekstrening

Vi vil nå la brukeren selv velge hvor mange gjenstander som skal s

```
>>>
```

```
Skriv inn en gjenstand: ost
```

```
Skriv inn en gjenstand: melk
```

```
Skriv inn en gjenstand: brød
```

```
Skriv inn en gjenstand: ferdig
```

```
Hvor mange gjenstander vil du skrive ut? 2
```

```
0 ost
```

```
1 melk
```

Dette må du gjøre:

- ☐ Begynn med programmet du allerede har.
- ☐ Før gjenstandene skrives ut, spør om hvor mye som skal skri
- ☐ Avbryt utskriften når antallet er lik det brukeren ba om.

## Strenger og indekser

Vi kan også bruke indekser på strenger. For eksempel:

```
>>> s = "Ada"  
>>> s[0]  
'A'
```

Vi skal nå skrive et program som henter input fra brukeren og skriver

```
>>>
```

```
Skriv inn en setning: Hei på deg!
```

```
H  
i  
p  
  
e  
!
```

Dette må du gjøre:

- ☐ Hent input fra brukeren.
- ☐ Bruk en løkke for å hente ut hver bokstav og dens indeks.
- ☐ Hvis indeksen er et partall, skriv ut bokstaven.

**Hint:** `tall%2` er *resten* av `tall` delt på `2`, hva gir `tall%2` når

**Lisens:** [CC BY-SA 4.0](#) **Forfatter:** Ole Kristian Pedersen, Kodeklu