

# ▲ Lister og indekssar

*Skrevet av: Ole Kristian Pedersen, Kodeklubben Trondheim*

*Oversatt av: Stein Olav Romslo*

*Kurs: Python*

*Tema: Tekstbasert*

*Fag: Programmering*

*Klassetrinn: 5.-7. klasse, 8.-10. klasse*

## Introduksjon

Denne oppgåva handlar om *lister*, altså å samle fleire ting i ein og same variabel. Sidan lister og løkker heng tett saman i Python bør du sjå på løkker i repetisjonsoppgåva ([../repetisjon/repetisjon\\_nn.html#l%C3%B8kker](#)) viss du har gløymt korleis løkker fungerer.

## Korleis lage lister?

Kvar ting i ei liste kallast for eit *element*. Me lagar ei liste ved å skrive element inni hakeparentesar, `[]`, med komma, `,`, mellom elementa:

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

No har me laga ei liste som inneheldt orda 'egg', 'ham' og 'spam'. Det er ikkje vanskelegare enn det! Me kan òg lage tomme lister:

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

Ei liste kan innehalde alt mogleg - tal, strengar, eller til og med andre lister!

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

# Leggje til og fjerne element

Kva viss me ynskjer å leggje til eller fjerne elementa frå lista vår? Då kan me bruke dei to funksjonane `lst.append(elm)` og `lst.remove(elm)`, der `lst` er lista og `elm` er elementet me vil leggje til eller fjerne.

Funksjonen `lst.append(elm)` legg til `elm` på slutten av `lst`, slik som du ser i dømet under:

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

Funksjonen `lst.remove(elm)` slettar det fyrste elementet `elm` frå `lst`. Det vil seie at viss `elm` ligg i lista `lst` fleire gonger, så blir berre det fyrste elementet lik `elm` sletta:

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

## Handleliste

No skal me lage eit handlelisteprogram. Programmet skal be brukaren skrive inn matvarer, og så skrive ut matvarene når brukaren skriv ferdig. Programmet skal fungere slik:

```
>>>
Skriv inn ein gjenstand: ost
Skriv inn ein gjenstand: mjølk
Skriv inn ein gjenstand: brød
Skriv inn ein gjenstand: ferdig
ost
mjølk
brød
```

Dette må du gjere:

- ☐ Lag ei tom liste.
- ☐ Be om input.
- ☐ Så lenge input ikkje er lik ferdig skal du leggje til det nye elementet i lista.

**Hint:** Kva type løkke kan me bruke her?

- ☐ Skriv ut kvart element i lista.

**Hint:** Kva type løkke kan me bruke her?

## Indeksar

Tenk deg at me har ei liste, og vil hente ut det andre elementet i lista. Korleis skal me klare det? Då brukar me noko kalla *indeks*. Indeks er plassen til elementet, og blir skrive i klammeparentesar, `[]`, rett etter variabelen: `lst[index]`. Her er eit døme på ei liste med tal:

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

Du la kanskje merke til at me skreiv `1`, men fekk ut det andre elementet i lista? Det er fordi me startar å telje på `0`. Difor får det fyrste elementet i lista indeks `0`, og det andre får indeks `1`. Dette er heilt vanleg. Du hugsar kanskje at det same skjer når me brukar `range()`?

```
>>> list(range(5))
[0, 1, 2, 3, 4]
```

Til no har me brukt `for element in lst` for å gå gjennom elementa i lista, men nokre gonger kan det vere praktisk å telje kor langt me har kome i lista. Til det kan me bruke `enumerate()` som gir oss både verdien og indeksen:

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

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

I dømet over får `i` verdien av indeksen, og `value` får verdien av elementet. Det er nesten som ei vanleg `for`-løkke, men me får indeksen i tillegg.

## Nummerert (indeksert) handleliste

No skal du modifisere programmet frå førre oppgåve til å skrive ut indeksar ved sidan av gjenstandane i handlelista. Det skal fungere slik:

```
>>>
Skriv inn ein gjenstand: ost
Skriv inn ein gjenstand: mjølk
Skriv inn ein gjenstand: brød
Skriv inn ein gjenstand: ferdig
0 ost
1 mjølk
2 brød
```

Dette må du gjere:

- ☐ Bruk programmet du laga tidlegare.
- ☐ Bruk `enumerate` for å få indeksen til kvart element.
- ☐ Skriv ut indeksen på same linje som elementet i lista.

## Indekstrening

No vil me at brukaren sjølv skal velje kor mange gjenstandar som skal skrivast ut, slik som i dømet under:

```
>>>
```

```
Skriv inn ein gjenstand: ost
```

```
Skriv inn ein gjenstand: mjølk
```

```
Skriv inn ein gjenstand: brød
```

```
Skriv inn ein gjenstand: ferdig
```

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

```
0 ost
```

```
1 mjølk
```

Dette må du gjere:



Start med programmet du allereie har.



Før du skriv ut lista, spør kor mykje du skal skrive ut.



Avbryt utskrifta når du har skrive ut så mange element som brukaren ba om.

## Strengar og indeksar

Me kan bruke indeksar på strengar òg. Til dømes:

```
>>> s = "Ada"
```

```
>>> s[0]
```

```
'A'
```



### Skrive ut annankvar bokstav

No skal me skrive eit program som hentar input frå brukaren og skriv ut annankvar bokstav. Det skal fungere slik:

```
>>>
```

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

```
H
```

```
i
```

```
p
```

```
e
```

```
!
```

Dette må du gjøre:

- ☐ Hent input frå brukaren.
- ☐ Bruk ei løkke for å hente ut kvar bokstav og indeksen til bokstaven.
- ☐ Viss indeksen er eit partal skriv du ut bokstaven.

**Hint:**  $\text{tal} \% 2$  er *reisten* av  $\text{tal}$  delt på 2. Kva gir  $\text{tal} \% 2$  når  $\text{tal}$  er eit partal?