

Lab uke 3 - Oppgaver

Oppgave 1) Python list

Egenskap:

- Har en rekkefølge
 - o Indeks som forteller hvor dataen ligger
 - o Teller fra 0 og oppover
 - o Indeks -1 er det siste elementet i lista
- Har mange metoder/tjenester som tilbys, men vi skal se på
 - o `append()`, `insert()`, `count()`, `remove()`, `pop()`

Utforsk: `append()` og `insert()`

Lag en ny python-fil, og legg til følgende linje

```
tall_liste = []
```

Variabelen 'tall_liste' har fått den tomme lista som sin verdi

Skriv verdien til variabelen 'tall_liste' ut i terminalen.

Hva ser du?

`append(elem)` - legger til et nytt element på slutten av lista

Legg til følgende linjer:

```
tall_liste.append(1)
tall_liste.append(2)
print(tall_liste)
```

index	element
0	1
1	3
2	2

Figur 1

Hva ser du?

insert(indeks, elem) – legger til elementet til en gitt posisjon

Legg til følgende linje:

```
tall_liste.insert(1, 3)
```

Skriv tall_liste ut i terminalen. Hva ser du?

Figur 1 over viser strukturen til lista. Merk at vi har indeksering fra 0 og oppover som bestemmer rekkefølgen til elementene.

Legg merke til hvordan metodene/tjenestene brukes:

```
variabel.metode(argumenter)
```

Andre liste-metoder som bør eksperimenteres med:

count(elem) – teller opp hvor mange ganger et element finnes i lista

remove(elem) – fjerner et gitt element fra lista

pop(indeks) – fjerner elementet som ligger på en gitt posisjon

Anta at følgende linjer er lagt til:

```
tall_liste.remove(3)
```

```
tall_liste.pop(0)
```

Hvordan ser tall_liste ut nå?

Bruk litt tid og eksperimenter med disse metodene.

Oppgave 2a) Behandling av data

Lag en tom python-fil og legg til følgende linje:

```
handleliste = ["melk", "tomat", "gulost"]
```

Brukeren lurte på størrelsen på handlelisten. Skriv ut antall elementer handlelisten består av med en passende tekst, `len(handleliste)`

Brukeren har helt glemt hva det første elementet i lista er. Skriv ut det første elementet ut i terminalen med en passende tekst.

Brukeren ombestemte seg og vil kjøpe en parmesan istedenfor en gulost. Forandre på 'handleliste' slik at det tredje elementet blir erstattet med «parmesan».

Oppgave 2b) Interaktivitet og struktur

Be brukeren om to elementer de har lyst å legge til på handlelista, og legg dem til.

Tegn opp strukturen til handleliste, slik illustrert på figur 1.

Tips: `input()` og `append()`

Oppgave 3) Mengder

Egenskap:

- ingen spesifikk rekkefølge
- ingen duplikater
 - o alle elementer er ulike

Legg til følgende kode:

```
min_mengde = {1, 3, 2}
print("Min mengde: ", min_mengde)
print("Består av " + len(min_mengde) + " elementer")
```

Fiks programmet slik at det kjører uten feilmelding

Legg til følgende linje:

```
print(min_mengde == {1, 2, 3, 2, 3})
```

Hva ser du, og hvorfor er verdien slik den er?

Legg til følgende kode:

```
tall_liste = [1, 1, 2, 3, 4, 4]
tall_liste = list(set(tall_liste))
```

Skriv ut `tall_liste` til terminalen. Hva har skjedd?

Oppgave 4) Ordbøker

Egenskap

- har en nøkkelverdi og en innholdsverdi
- nøkkelverdien refererer til en innholdsverdi

Legg til følgende linjer:

```
hovedsteder = {"Norge": "Oslo", "Sverige": "Stockholm",  
               "England": "London"}  
  
print(hovedsteder)
```

Vi sier at landet er nøkkelverdien, og hovedstaden er innholdsverdien.

Oppdater ordboka hvor du legger til hovedstaden til Frankrike.

Slå opp i ordboka og finn hovedstaden til England. Skriv ut i terminalen med passende tekst.

Be brukeren om et land og lagre verdien til en variabel 'land'.

Be brukeren om hovedstaden til det gitte landet og lagre verdien til en variabel 'hovedstad'.

Legg denne informasjonen inn i ordboka.

Skriv ut hovedstaden som variabelen 'land' refererer til med en passende tekst.

Hint:

For å oppdatere ordboka eller legge inn nye elementer, er syntaksen:

```
ordbok[nøkkelverdi] = innholdsverdi
```

For å slå opp i ordboka (gitt nøkkelverdi), er syntaksen:

```
ordbok[nøkkelverdi]
```

Oppgave 5) Nøstede lister

En nøstet liste er en liste av lister.

```
nested = [[1, 2, 3], 4, 5, [[6, 7], 8]]
```

Hva er det første elementet i nested?

Hvordan kan man aksessere sublisten [6, 7] ved å bruke nested?

Ekstra:

Sjekke om et element finnes i en liste?

```
print(elem in liste) -> True dersom elem finnes i liste, False ellers
```

Sjekke om en nøkkelverdi finnes i ei ordbok?

```
print(key in ordbok) -> True dersom nøkkelen finnes, False ellers
```