

Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 2 av Kodeskolens kræsjkurs i programmering. Dagens tema er *lister*, *for-løkker* og *tekst-behandling*. Dersom du står fast er det bare å spørre. I tillegg anbefaler vi å lese i kompendiet hvis det er noen temaer du synes er spesielt vanskelige. Oppgaver markert som bonusoppgaver er litt mer utfordrende og du velger selv om du har lyst til å prøve deg på dem. Det er også lov å hoppe over en oppgave dersom den ikke er utfordrende nok. God koding!

Lister

Oppgave 1 *Fruktliste*

I denne oppgaven skal vi øve å opprette lister og hente ut elementer med indeksering

- a) Opprett en liste, frukt, som inneholder elementene 'eple', 'hanningmelon', 'kiwi' og 'appelsin'
- b) Hvilken indeks har 'hanningmelon'? Bruk indeksering til å hente ut dette elementet fra lista og skriv det ut til terminalen
- c) Hvilket element får du dersom du indekserer med [-1]? Skriv ut elementet med indeks -1 og se om du hadde rett.

Løsning oppgave 1 *Fruktliste*

a)

```
1 frukt = ['eple', 'hanningmelon', 'kiwi', 'appelsin']
```

b)

```
1 print(frukt[1])
```

c)

```
1 print(frukt[-1])
```

```
appelsin
```

Oppgave 2 *Månednummer med lister*

Vennen din sliter veldig med å huske hvilken måned som tilhører hvilket nummer og han ber deg om å lage et program som kan hjelpe han med å lære dette. Du skal derfor lage et program som tar et heltall som input og printer ut hvilken måned det er. Her er et eksempel på hvordan programmet kan virke

```
Hvilket månednummer vil du vite navnet på? 5
Måned nummer 5 er mai.
```

- a) Start med å lage en variabel som heter måneder. Denne variabelen skal være en liste som inneholder alle månedsnavnene i rett rekkefølge.
- b) Sjekk at du har husket alle månedene ved å skrive ut måneder og lengden til måneder på skjermen.
- c) Lag en variabel månednummer. Be brukeren om hva denne skal være, og husk å gjøre den om til heltall.
- d) Skriv ut hvilken måned i måneder som har indeksen månednummer.
- e) Mennesker teller jo fra 1, mens Python teller fra 0. Vi vil skrive ut månedene slik mennesker tenker, altså januar er måned 1, ikke måned 0. For å få til dette må du trekke 1 fra månednummer. Gjør dette, skriv ut måneden som har indeksen månednummer `-1`.
- f) Pynt litt på koden din slik at utskriften ser fin ut.

Løsning oppgave 2 *Månednummer med lister*

```
1 måneder = ['januar', 'februar', 'mars', 'april', 'mai']
```

```

    , 'juni', 'juli', 'august', 'september', 'oktober',
    'november', 'desember']
2  antall_måneder = len(måneder)
3  print(f'Det er {antall_måneder} måneder, og de er {må
    neder}')
4
5  månednummer = input('Hvilket månednummer vil du vite
    navnet på? ')
6
7  måned = måneder[månednummer - 1]
8  print(f'Måned nummer {månednummer} er {måned}')

```

Oppgave 3 *Kattenavn*

I denne oppgaven skal vi bli vant med `in` operatoren for lister. Karl har tre katter: Koseguri, Attakai og Teebo. Vi skal lage et program som spør brukeren om navnet på en katt og sjekker om det er en av Karl sine katter med `in`.

- Lag en liste, `karls_katter` som inneholder navnet på alle kattene.
- Bruk `input` til å be brukeren om et kattenavn og lagre det i en variabel, `kattenavn`.
- Bruk `in` operatoren og en betingelse til å fortelle brukeren om kattenavnet er med i `karls_katter` lista eller ikke.

Løsning oppgave 3 *Kattenavn*

a)

```
1  karls_katter = ["Koseguri", "Attakai", "Teebo"]
```

b)

```
1  kattenavn = input("Hva heter katten")
```

c)

```
1 if kattenavn in karls_katter:
2     print(f"{kattenavn} er en av Karls katter")
3 else:
4     print(f"{kattenavn} er ikke en av Karls katter
      :(")
```

For-løkker

Oppgave 4 *For løkker*

Syntaksen for å løkke over en liste med en **for**-løkke i Python er slik:

```
1 for [løkkevariabel] in [liste]:
2     [det som skal gjentas, f.eks print(løkkevariabel)]
```

- a) Opprett en liste, navneliste, som skal inneholde fem navn. (Du kan velge selv hvilke navn)
- b) Bruk en **for**-løkke til å gå igjennom lista med navn og skrive ut en hilsen til hvert navn

Løsning oppgave 4 *For løkker*

a)

```
1 navneliste = ["Olivia", "Henrik", "Emma", "Eline",
               "Liam"]
```

b)

```
1 for navn in navneliste:
2     print(f"Heisann, {navn}!")
```

Oppgave 5 *Sum av arealer*

Moussa har seks kvadratiske plater med forskjellig sidelengder. Sidelengdene er lagret i en liste:

```
1 sidelengder = [1, 3, 4, 4, 6, 2.3]
```

For å finne summen av arealene til platene, tenker Moussa at han kan bruke en **for**-løkke

- a) Opprett en liste, `sidelengder`, slik som over
- b) Bruk en **for**-løkke til å løkke over sidelengdene for hvert kvadrat og regne ut arealet ved hjelp av sidelengden. Lagre arealet i en variabel, `areal`
- c) Lag en variabel, `arealsum`, som er `0` før løkka begynner og som øker med størrelsen på hvert areal du regner ut (**Hint:** `arealsum += areal`)
- d) Skriv ut totalarealet til terminalen

Løsning oppgave 5 *Sum av arealer*

a)

```
1 sidelengder = [1, 3, 4, 4, 6, 2.3]
```

b)

```
1 for sidelengde in sidelengder:
2     areal = sidelengde**2
3     print(areal)
```

c)

```
1 arealsum = 0
2 for sidelengde in sidelengder:
3     areal = sidelengde**2
```

```
4     print(areal)
5     arealsum += areal
```

d)

```
1     arealsum = 0
2     for sidelengde in sidelengder:
3         areal = sidelengde**2
4         print(areal)
5         arealsum += areal
6     print(arealsum)
```

Tekst-behandling

Oppgave 6 *Lipogram*

I 1939 tok forfatter Ernest Vincent Wright på seg utfordringen å skrive en hel roman uten å bruke bokstaven “e”. Boken heter *Gadsby* og er over 50 000 ord lang! Tekster som unngår en spesiell bokstav kalles et *lipogram* og det å unngå bokstaven “e” er spesielt vanskelig siden det er den mest brukte bokstaven i engelsk. La oss bruke Python til å teste en bit av *Gadsby* og dobbeltsjekk at Wright ikke har sneket inn noen “e”-er.

a) Lagre første avsnitt av *Gadsby* i en variabel ved å kopiere koden under:

```
1     gadsby_avsnitt = """If Youth, throughout all
        history, had had a champion to stand up for it;
        to show a doubting world that a child can
        think; and, possibly, do it practically; you
        wouldn't constantly run across folks today who
        claim that "a child don't know anything." A
        child's brain starts functioning at birth; and
        has, amongst its many infant convolutions,
        thousands of dormant atoms, into which God has
        put a mystic possibility for noticing an adult's
        act, and figuring out its purport"""
```

b) Bruk `in` operatoren til å sjekke om bokstaven 'e' finnes i avsnittet.

Skriv ut resultatet til terminalen

- c) **BONUS:** Den andre mest vanlige bokstaven i engelsk er bokstaven "a". Bruk en **for**-løkke til å løkke over hver bokstav i teksten og sjekk om det er en 'a'. Opprett en variabel som teller antall 'a'-er i teksten ved å øke med 1 hver gang en bokstav er 'a'.

Løsning oppgave 6 *Lipogram*

a)

```
1 gadsby_avsnitt = """If Youth, throughout all
    history, had had a champion to stand up for it;
    to show a doubting world that a child can
    think; and, possibly, do it practically; you
    wouldn't constantly run across folks today who
    claim that "a child don't know anything." A
    child's brain starts functioning at birth; and
    has, amongst its many infant convolutions,
    thousands of dormant atoms, into which God has
    put a mystic possibility for noticing an adult's
    act, and figuring out its purport"""
```

b)

```
1 if "e" in gadsby_avsnitt:
2     print("Jeg fant en e!")
3 else:
4     print("Ingen e!")
```

c)

```
1 antall_a = 0
2 for bokstav in gadsby_avsnitt:
3     if bokstav == "a":
4         antall_a += 1
5 print(antall_a)
```

Oppgave 7 *Sjekke filtyper*

I denne oppgaven skal du bruke **in**-operatoren for å identifisere .txt filer i en liste av forskjellige filnavn. La oss si at du har en liste med filnavn:

```
1 filnavnliste = ['logg020320.txt', 'resultater1_45.csv',  
                'programvare.exe', 'logg030320.txt', 'Screenshot-  
                2020-09-15.png', 'logg040320.txt']
```

- a) Opprett en liste ved å kopiere kodelinjen over
- b) Bruk en **for**-løkke sammen med **in**-operatoren og en betingelse til å skrive ut kun de filnavnene som inneholder teksten **'.txt'**.

Løsning oppgave 7 *Sjekke filtyper*

a)

```
1 filnavnliste = ['logg020320.txt', 'resultater1_45.  
                csv', 'programvare.exe', 'logg030320.txt', '  
                Screenshot-2020-09-15.png', 'logg040320.txt']
```

b)

```
1 for filnavn in filnavnliste:  
2     if ".txt" in filnavn:  
3         print(filnavn)
```

Oppgave 8 *Måneder med r i navnet*

Mila har hørt at hun bør få i seg ekstra D-vitamin i måneder som inneholder "r". Så hun har lyst til å bruke Python til å løkke gjennom alle månednavnene og skrive ut kun de som inneholder bokstaven "r".

- a) Opprett en liste, månednavn som inneholder navnet på alle 12 måneder

- b) Bruk en **for**-løkke til å løkke igjennom disse månedene og skriv ut hvert månednavn til terminalen
- c) Modifiser programmet til å bruke en betingelse for å kun skrive ut de månedene som inneholder bokstaven r. (**Hint**: du kan bruke **in**-operatoren for å se om en tekststreng er inneholdt i en annen tekststreng)

Løsning oppgave 8 *Måneder med r i navnet*

a)

```
1 månednavn = ["Januar", "Februar", "Mars", "April",  
               "Mai", "Juni", "Juli", "August", "September",  
               "Oktober", "November", "Desember"]
```

b)

```
1 for måned in månednavn:  
2     print(måned)
```

c)

```
1 for måned in månednavn:  
2     if "r" in måned  
3         print(måned)
```

Slå opp i dokumentasjonen

Oppgave 9 *Help!*

Sacha er missfornøyd med at **print**-funksjonen lager en ny linje for hvert funksjonskall. Hun har lyst til å se i dokumentasjonen om det er mulig å bruke **print** uten dette. I denne oppgaven skal du bruke **help** funksjonen i Python til å lese dokumentasjonen til **print**-funksjonen

- a) Bruk **help(print)** til å skrive dokumentasjonen til **print** ut til termi-

nalén.

- b) Hvilke 4 “Optional keyword arguments” står listet i dokumentasjonen til `print`?
- c) Hva står som beskrivelsen til end argumentet. Hva tror du det betyr?
- d) Hva vil output for følgende program bli? (Skriv ned svaret først. Kjør programmet for å sjekke.)

```
1 print("sofa")
2 print("pute")
```

- e) Hva med følgende program ? (Skriv ned svaret først. Kjør programmet for å sjekke)

```
1 print("sofa", end=" ")
2 print("pute", end=" ")
```

- f) Modifiser kun end-argumentene i programmet over for å få følgende output:

```
sofa-pute!
```

Løsning oppgave 9 *Help!*

a)

```
1 help(print)
```

b)

```
Optional keyword arguments:
file:  a file-like object (stream); defaults
       to the current sys.stdout.
sep:   string inserted between values, default
       a space.
end:   string appended after the last value,
       default a newline.
```

```
flush: whether to forcibly flush the stream.
```

c)

```
end:    string appended after the last value,  
        default a newline.
```

Dette betyr at end-argumentet bestemmer hva som skal være på slutten av beskjedene som skrives ut til brukeren. Som standard er dette et linjeskift. Husk: `print`-funksjonen lager et nytt linjeskift hver gang den blir brukt. Men vi kan også sette den til å være noe annet.

d)

```
sofa  
pute
```

e)

```
sofapute
```

f)

```
1 print("sofa", end="-")  
2 print("pute", end="!")
```

Oppgave 10 *Slå opp i dokumentasjonen*

I denne oppgaven skal du slå opp i dokumentasjonen for å finne ut av hva `list.count`-funksjonen gjør.

- a) Bruk en nettleser eller `help`-funksjonen (eller noe annet) til å finne dokumentasjonen for `list` i Python.
- b) Hva er beskrivelsen for `count`-funksjonen i dokumentasjonen?
- c) Basert på denne beskrivelsen, bruk `count` til å finne antall ganger teksten «blå» finnes i lista under

```
1 farger = ["rosa", "blå", "blå", "lilla", "sort", "  
           rød", "lilla", "blå"]
```

Løsning oppgave 10 *Slå opp i dokumentasjonen*

a) Her er det mange mulige løsninger. For eksempel kan du skrive `help(list)` inn i et Python script og kjøre det. Eller du kan skrive det inn i et Python terminalvindu. Det er også en mulighet å gå inn på <https://docs.python.org/3/tutorial/datastructures.html>.

b) I beskrivelsen til `count` står det:

```
list.count(x)
    Return the number of times x appears in the
    list.
```

c)

```
1 farger = ["rosa", "blå", "blå", "lilla", "sort", "
   rød", "lilla", "blå"]
2 antall = farger.count("blå")
3 print(f"Antall ganger 'blå' er i lista: {antall}")
```

```
Antall ganger 'blå' er i lista: 3
```