

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-behandlinger*. 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.

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.

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`.

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

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

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

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 dobbeltsjekke 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'`.

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'`.

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)

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 terminalen.
- 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!
```

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"]
```