

Oppgaver

Her er et knippe repetisjonsoppgaver som vil være god forberedelse til Kodeskolens videregående kurs i programmering for lærere våren 2021.

Oppgavene er ment som oppfriskning på grunnleggende programmeringskonsepter som er relevante for realfag. God koding!

1 Variabler og regning

Oppgave 1 *Vekstfaktor*

Når du skal øke en verdi med p %, blir vekstfaktoren $1 + \frac{p}{100}$.

Hint: Det kan være lurt å lese igjennom alle deloppgavene før du begynner for å få en full oversikt over hva du skal gjøre

- a) Lag en variabel prosent som har verdien **31**
- b) Regn ut tilhørende vekstfaktor og lagre resultatet i en ny variabel, vekstfaktor
- c) Skriv ut vekstfaktoren til terminalvinduet med **print**. Hva blir vekstfaktoren for 31% vekst?
- d) Modifiser programmet ditt slik at det skriver ut vekstfaktoren for 67% istedet

Løsning oppgave 1 *Vekstfaktor*

a)

```
1 prosent = 31
```

b)

```
1 vekstfaktor = 1 + prosent/100
```

c)

```
1 print(f'Prosent: {prosent}%, vekstfaktor: {vekstfaktor}')
```

```
Prosent: 31%, vekstfaktor: 1.31
```

Vekstfaktoren for 31% er altså 1.31

d)

```
1 prosent = 67
2 vekstfaktor = 1 + prosent/100
3 print(f'Prosent: {prosent}%, vekstfaktor: {vekstfaktor}')
```

```
Prosent: 67%, vekstfaktor: 1.67
```

2 Input og betingelser

Oppgave 2 *Positive og negative tall*

I denne oppgaven skal vi bruke *betingelser* til å undersøke om et tall er positivt eller negativt.

- a) Bruk `input` og `float` til å be brukeren om et tall og lagre det i en variabel, `tall`
- b) Bruk `if` med en *betingelse* til å skrive ut beskjenen “tallet er positivt” dersom tall er større enn 0. Test programmet ditt med å gi 4 som input.
- c) Bruk `elif` og en betingelse til å skrive ut beskjenen “tallet er null” dersom tall er lik 0. Test programmet ditt ved å gi 0 som input.

- d) Bruk **else** til å skrive ut beskjeden “tallet er negativt” dersom tallet verken er større enn eller lik 0. Test programmet ved å gi programmet -1 som input.

Løsning oppgave 2 *Positive og negative tall*

a)

```
1 tall = float(input('Gi meg et tall: '))
```

b)

```
1 tall = float(input('Gi meg et tall: '))
2 if tall > 0:
3     print('Tallet er positivt')
```

```
Gi meg et tall: 4
Tallet er positivt
```

c)

```
1
2 if tall > 0:
3     print('Tallet er positivt')
4 elif tall==0:
5     print('Tallet er null')
```

```
Gi meg et tall: 0
Tallet er null
```

d)

```
1 tall = float(input('Gi meg et tall: '))
2
```

```

3  if tall > 0:
4      print('Tallet er positivt')
5  elif tall==0:
6      print('Tallet er null')
7  else:
8      print('Tallet er negativt')

```

```

Gi meg et tall:  -1
Tallet er negativt

```

3 Løkker

Oppgave 3 *For-løkker*

Bruk en **for**-løkke til å gjennomføre disse oppgavene.

- Print meldingen 'Hei, verden' 5 ganger.
- Print alle tallene fra 1 til 101.
- Modifiser programmet slik at det også printer ut alle kvadrattallene (x^2) mellom 1 og 10000 (100^2). Husk at du kan opphøye tallet tall i andre med kommandoen `tall**2`.

Løsning oppgave 3 *For-løkker*

a)

```

1  for verdi in range(5):
2      print("Hei, verden")

```

b)

```

1  for tall in range(1,101):
2      print(tall)

```

c)

```
1 for tall in range(1,101):  
2     print(tall, tall**2)
```

4 Funksjoner

Oppgave 4 *Funksjoner i Python*

I Python defineres en funksjon med følgende syntaks:

```
1 def funksjonsnavn(inputvariabel):  
2     # [instruksjonene funksjonen skal utføre]  
3     return #[det som skal returneres]
```

- a) Definer en *funksjon*, *doble* som tar inn *inputvariabelen*, *tall*, regner ut det dobbelte av *tall* og *returnerer* svaret av utregningen
- b) *Kall* på funksjonen med 3 som input
- c) Lagre resultatet av funksjonskallet i en variabel, *resultat*
- d) Skriv resultatet ut til terminalen med **print**

Løsning oppgave 4 *Funksjoner i Python*

a)

```
1 def doble(tall):  
2     return 2*tall
```

b)

```
1 doble(3)
```

c)

```
1 resultat = doble(3)
```

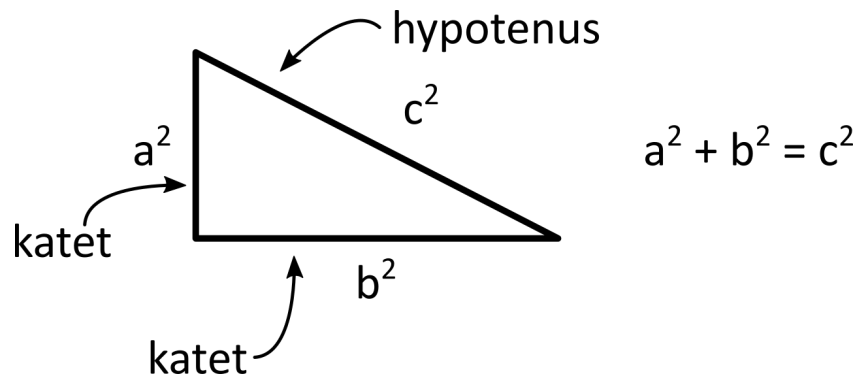
d)

```
1 print(resultat)
```

```
6
```

Oppgave 5 *Pytagoras og vektorlengder*

Vi kan finne hypotenusen til en rettvinklet trekant ved hjelp av pytagoras setning:



- a) Definer en funksjon `pytagoras(katet1, katet2)` som tar inn lengden til katenene i en rettvinklet trekant og returnere hypotenusen
- b) Definer en funksjon `vektorlengde(x1, x2)` som tar inn en vektor og returnerer lengden. Funksjonen skal kalle på pytagoras funksjonen.
- c) Bruk funksjonen din til å finne lengden til en vektor [6.2, 9.3]

Løsning oppgave 5 *Pytagoras og vektorlengder*

a)

```
1 def pythagoras(katet1, katet2):  
2     return sqrt(katet1**2 + katet2**2)
```

b)

```
1 def vektorlengde(x1, x2):  
2     return pythagoras(x1, x2)
```

c)

```
1 lengde = vektorlengde(6.2, 9.3)  
2 print(f'Vektorlengde: {lengde:.2f}')
```

```
Vektorlengde: 11.18
```

5 Plot

Oppgave 6 *Plotte annengradsfunksjon*

En funksjon er definert slik:

$$f(x) = x^2 + 3x - 10$$

I denne oppgaven skal vi bruke Python til å plotte funksjonen for $x = -5, \dots, 5$

- a) Definer en funksjon $f(x)$ som tar inn en x -verdi og returnerer $f(x)$ slik den er definert over.

- b) Bruk `arange` fra `numpy` (eller `pylab`) til å opprette en array, `x_verdier` som inneholder verdier fra -5 til 5 med steglengde 0.5.
- c) Bruk funksjonen `f` du definerte i a) til å regne ut tilhørende `y`-verdier og lagre dem i en variabel, `y_verdier`.
- d) Bruk `plot` og `show` fra `matplotlib.pyplot` (eller `pylab`) til å plote funksjonen med `x_verdier` på x-aksen og `y_verdier` på y-aksen.
- e) Bruk `xlim` til å endre plotteområde til mellom -5 og 5.

Løsning oppgave 6 *Plote annengradsfunksjon*

a)

```
1 def f(x):  
2     return x**2 + 3*x - 10
```

b)

```
1 from numpy import linspace  
2 x_verdier = linspace(-5, 5, 101)
```

c)

```
1 y_verdier = f(x_verdier)
```

d)

```
1 from matplotlib.pyplot import plot, show  
2 plot(x_verdier, y_verdier)  
3 show()
```

e)


```
1 from matplotlib.pyplot import plot, show, xlim
2 plot(x_verdier, y_verdier)
3 xlim(-5, 5)
4 show()
```