

# Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 1 av Kodeskolens videregående kurs i programmering for lærere høsten 2020.

Dersom du står fast er det bare å spørre. Oppgavene er ment som oppfriskning på grunnleggende programmeringskonsepter som er relevante for realfag. Oppgavene er organisert etter programmeringskonsept og den første oppgaven for hver seksjon er enklest (for den seksjonen). God koding!

## 1 Variabler, input og regning

### Oppgave 1 *Vekstfaktor*

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

- a) Lag en variabel prosent som har verdien 31
- b) Regn ut tilhørende vekstfaktor og lagre resultatet in 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
```

## Oppgave 2 *Finn fire feil!*

Her følger det fire programmer som har blitt skrevet feil. Finn feilen i hver programsnitt. Du kan godt kjøre programmet inn på din egen maskin, og kjøre det, da kan kanskje feilmeldingen hjelpe deg å skjønne hva som er galt.

Når du tror du skjønner hva som er galt, rett feilen på din egen maskin, og kjør programmet for å sjekke at det fungerer som det skal.

a)

```
1 Print('Python er gøy!')
```

b)

```
1 print(Hei, Python!)
```

c)

```
1 side = int(input('Hva er siden på kvadratet?'))
2 print(f'Arealet er {sidelengde**2}')
```

d)

```
1 radius = 4
2 pi = 3,14
3 volum = (4/3)*radius*pi**3
```

## Løsning oppgave 2 *Finn fire feil!*

a) Vi må bruke liten p i print:

```
1 print('Python er gøy!')
```

b) Vi må huske på fnuttene våre

```
1 print('Hei, Python!')
```

c) Vi må passe på at vi bruker riktig variabel:

```
1 sidelengde = int(input('Hva er siden på kvadratet?
'))
2 print(f'Arealet er {sidelengde**2}')
```

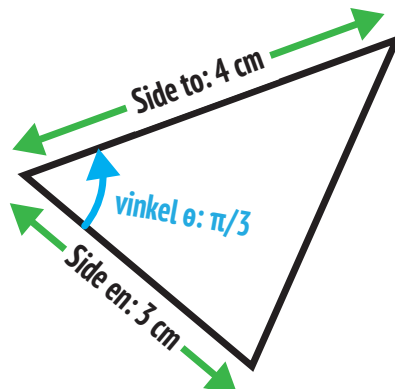
d) Vi må bruke punktum som desimaltegn, ikke komma:

```
1 radius = 4
2 pi = 3.14
```

```
3 volum = (4/3)*radius*pi**3
```

### Oppgave 3 *Arealsetningen*

Vi kjenner til to sider og vinkelen mellom dem i en trekant:



*Arealsetningen* sier at arealet er gitt ved:

$$\frac{1}{2} \times \text{side1} \times \text{side2} \times \sin \theta$$

- Vi ser at vi kommer til å trenge sin-funksjonen. Importer sin og pi fra math biblioteket.
- Lag variablene side1, side2 og vinkel og gi dem verdiene 3, 4 og  $\frac{\pi}{3}$  (**OBS:** vinkelen er gitt i radianer)
- Regn ut arealet til trekanten og skriv resultatet ut til terminalen
- Modifiser programmet ditt til å først spørre brukeren om verdiene til side1, side2 og vinkel med float og input og så skrive ut arealet til terminalen.

### Løsning oppgave 3 *Arealsetningen*

a)

```
1 from math import sin, pi
```

b)

```
1 side1 = 3
2 side2 = 4
3 vinkel = pi/3
```

c)

```
1 areal = (1/2)*side1*side2*sin(vinkel)
2 print(f'Arealet er {areal}')
```

d)

```
1 from math import sin, pi
2 side1 = float(input('Hva er lengden til side en?'))
3 side2 = float(input('Hva er lengden til side to?'))
4 vinkel = float(input('Hva er vinkelen mellom de?'))
5 areal = (1/2)*side1*side2*sin(vinkel)
6 print(f'Arealet er {areal}')
```

## 2 Betingelser

#### Oppgave 4 *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 beskjedene “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 beskjedene “tallet er null” dersom `tall` er lik 0. Test programmet ditt ved å gi 0 som input.
- d) Bruk `else` til å skrive ut beskjedene “tallet er negativt” dersom tallet verken er større enn eller lik 0. Test programmet med -1 som input.

#### Løsning oppgave 4 *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

```

### Oppgave 5 *Beskrive tilstanden til kjemiske stoff*

En venn av deg har lagd et program som forteller deg om vann fryser eller ikke basert på temperatur. Men det finnes jo flere stoffer enn vann, og du har lyst til å lage et program som kan fortelle deg om de er fast form, væske eller gass.

Programmet til vennen din ser du her:

```

1  kokepunkt = 100 # grader Celcius
2  frysepunkt = 0  # grader Celcius
3  stoff = 'vann'
4
5  temperatur = float(input(f'Hvilken temperatur har {
6      stoff}?'))

```

```

7 if temperatur < frysepunkt:
8     print('Det er et fast stoff')
9 elif temperatur < kokepunkt:
10    print('Det er en væske')
11 else:
12    print('Det er en gass')

```

- a) Utvid programmet til at du bruker `input` funksjonen til å spørre brukeren om hvilket stoff den vil vite fasen til.
- b) Bruk en `if` og en betingelse til å sjekke om brukeren skriver vann. Hvis brukeren skrev vann skal kokepunkt variabelen settes til 100 og frysepunkt variabelen settes til 0. Hvis brukeren skrev noe annet enn 'vann' skal 'Jeg vet ikke om dette stoffet' skrives ut til brukeren. NB: Hvis brukeren skriver inn noe annet enn vann så vil denne beskjedens printes ut, og så vil vi få en feilmelding!
- c) Nå må vi passe på at programmet avsluttes uten å krasje dersom brukeren skriver inn noe annet enn vann. For å gjøre det trenger vi `exit` funksjonen fra `sys` biblioteket. Start derfor programmet ditt med å hente denne funksjonen ved å skrive `from sys import exit`.
- d) Modifiser `else`-blokken til programmet ditt, slik at du kaller på funksjonen `exit()` etter at 'Jeg vet ikke om dette stoffet' skrives ut til brukeren. Test programmet ditt, hva skjer om du skriver inn et stoff det ikke kjenner til (f.eks. saltvann)
- e) Legg til en mulighet for å skrive *håndsprit*. Håndsprit fryser på ca  $-89$  grader og koker på ca  $83$  grader.
- f) Utvid programmet til å støtte kvikksølv og luft. Kvikksølv fryser på ca  $-39$  grader og koker på ca  $357$  grader, luft fryser på ca  $-210$  grader og koker på ca  $-196$  grader.

### Løsning oppgave 5 *Beskrive tilstanden til kjemiske stoff*

Under ser vi hele programmet som ble lagd i oppgaven over, og hver nye linje har en kommentar som sier hvilken deloppgave den ble skrevet i.

```

1 from sys import exit # c)

```



```

2
3 stoff = input("Hvilket stoff har vi?") # a)
4
5 if stoff == 'vann': # b)
6     kokepunkt = 100 # b)
7     frysepunkt = 0 # b)
8 elif stoff == 'håndsprit': # e)
9     frysepunkt = -89 # e)
10    kokepunkt = 83 # e)
11 elif stoff == 'kvikksølv': # f)
12     frysepunkt = -39 # f)
13     kokepunkt = 357 # f)
14 elif stoff == 'luft': # f)
15     frysepunkt = -210 # f)
16     kokepunkt = -196 # f)
17 else: # b)
18     print('Jeg kjenner ikke til dette stoffet.') # b)
19     exit() # d)
20
21
22 temperatur = float(input(f'Hvilken temperatur har {
    stoff}'))
23
24 if temperatur < frysepunkt:
25     print('Det er et fast stoff')
26 elif temperatur < kokepunkt:
27     print('Det er en væske')
28 else:
29     print('Det er en gass')

```

### 3 Løkker

#### Oppgave 6 *For-løkker*

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

- a) Print meldingen 'Hei, verden' 5 ganger.

- b) Print alle tallene fra 1 til 101.
- c) 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 6 *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)
```

### Oppgave 7 *Kjøpe telefon på kreditt*

Hallgeir har veldig lyst på en ny telefon som koster 2999 kroner. Problemet er bare at han har brukt opp sparepengene sine på andre ting. For å få kjøpt telefonen skaffer Hallgeir et kredittkort som har 30% rente. Etter å ha kjøpt telefonen går det tre år før Hallgeir betaler tilbake kreditten. I denne oppgaven skal vi undersøke hvor mye Hallgeir blir nødt til å betale da.

- a) Opprett variablene `renter`, `originalpris` og `antall_år` og gi med verdiene 30, 2999 og 3
- b) Regn ut vekstfaktoren til renta og lagre den i en variabel `rentevekstfaktor`

- c) Opprett en variabel, `lån`. Denne variabelen skal holde orden på hvor stort lånet til Hallgeir er. Til å begynne med er lånet like stort som prisen på telefonen. Sett altså lån-variabelen til å ha verdien `2999`
- d) Bruk en `for`-løkke til å simulere hvordan lånet vokser for hvert år. Hint: for hvert år skal lånet ganges med vekstfaktoren du regnet ut i oppgave b)
- e) Oppdater programmet ditt til å skrive ut størrelsen på lånet for hvert år. Hvor mye skylder Hallgeir etter tre år?
- f) Hvor mye ekstra kostet telefonen i forhold til originalprisen?
- g) Gå inn på <https://kredittkort.com/> og se hvilket kort som kommer øverst og noter deg renta. Endre renta i programmet ditt til å matche denne renta. Gå inn på <https://www.prisjakt.no/category.php?k=103> og se hvilken telefon som er mest populær. Endre originalprisen i programmet ditt til å matche prisen til den telefonen. Kjør programmet nå. Hva ville denne telefonen og dette kredittkortet kostet Hallgeir?

### Løsning oppgave 7 *Kjøpe telefon på kreditt*

a)

```
1 renter = 30
2 originalpris = 2999
3 antall_år = 3
```

b)

```
1 rentevækstfaktor = 1 + renter/100
```

c)

```
1 lån = originalpris
```

d)

```

1  for år in range(antall_år):
2      lån *= rentevekstfaktor
3
4  print(f"Hallgeir skylder {lån:.2f} kroner etter {
      antall_år} år")

```

```
Hallgeir skylder 6588.80 kroner etter 3 år
```

- e) Det kostet 3589.80 kroner mer enn originalprisen å kjøpe telefonen på kreditt.
- f) Det kredittkortet med lavest rente har en rente på 23,1%
- g) Den vanligste telefonen er en iPhone 11 64GB og koster 7990 kroner. Om vi bruker disse tallene i koden får vi at telefonen koster 14904,62 kroner, det er nesten dobbelt så mye som originalprisen!

## 4 Funksjoner

### Oppgave 8 *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 8 *Funksjoner i Python*

a)

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

b)

```
1 doble(3)
```

c)

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

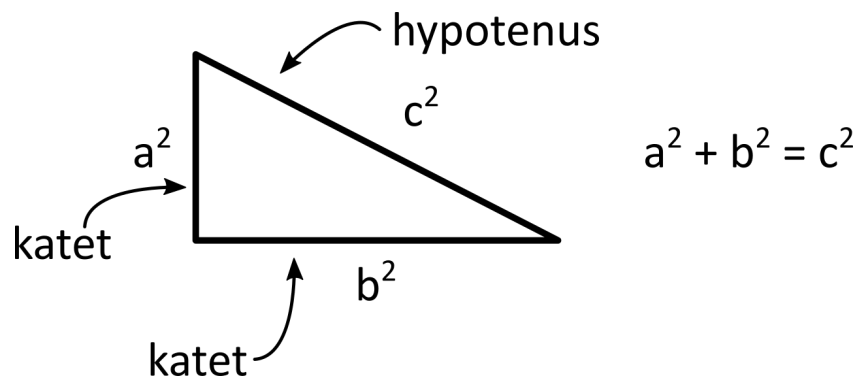
d)

```
1 print(resultat)
```

```
6
```

### Oppgave 9 *Pytagoras og vektorlengder*

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



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

### Løsning oppgave 9 *Pythagoras og vektorlengder*

a)

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

b)

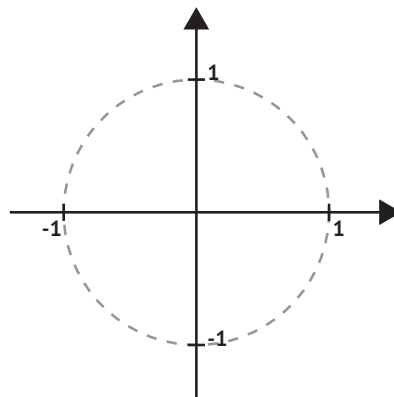
```
1 def vektorlengde(x1, x2):
2     return pythagoras(x1, x2)
```

c)

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

### Oppgave 10 *Sjekke om punkt er i en sirkel*

I denne oppgaven skal vi undersøke om et punkt er innenfor en enhetssirkel. For å gjøre det må vi sjekke om avstanden mellom punktet og origo er mindre en 1. Vi ser av figuren under at denne avstanden er lik hypotenusen til en rettvinklet trekant med katet lik  $x$ -koordinaten og  $y$ -koordinaten.



- a) Lag en funksjon `pythagoras(katet1, katet2)` som tar inn lengden til to kateter i en rettvinklet trekant og returnerer hypotenusen
- b) Lag en funksjon `innenfor_enhetssirkel(x,y)` som tar inn  $x$  og  $y$  koordinatene til et punkt og returnerer `True` dersom punktet er innenfor enhetssirkelen og `False` dersom det er utenfor. Funksjonen skal kalle på `pythagoras` funksjonen
- c) Bruk funksjonen din til å sjekke om følgende punkter er innenfor sirkelen
  - $(0,0)$
  - $(1.5, 9.8)$
  - $(-0.5, 0.99)$

## Løsning oppgave 10 *Sjekke om punkt er i en sirkel*

a)

```
1 from math import sqrt
2
3 def pythagoras(katet1, katet2):
4     return sqrt(katet1**2 + katet2**2)
```

b)

```
1 def innenfor_enhetssirkel(x, y):
2     return pythagoras(x, y) < 1
```

c)

•

```
1 svar = innenfor_enhetssirkel(0, 0)
2 print(f'Er punktet innenfor enhetssirkelen?: {
    svar}')
```

Er punktet innenfor enhetssirkelen?: True

•

```
1 svar = innenfor_enhetssirkel(1.5, 9.8)
2 print(f'Er punktet innenfor enhetssirkelen?: {
    svar}')
```

Er punktet innenfor enhetssirkelen?: False

•

```
1 svar = innenfor_enhetssirkel(-0.5, 0.99)
2 print(f'Er punktet innenfor enhetssirkelen?: {
    svar}')
```

Er punktet innenfor enhetssirkelen?: False



## 5 Plot

### Oppgave 11 *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` 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` til å plotte 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 11 *Plotte 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)

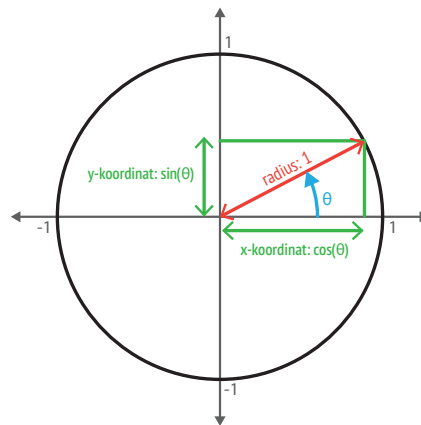
```
1 from matplotlib.pyplot import plot, show
2 plot(x_verdier, y_verdier)
3 show()
```

e)

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

### Oppgave 12 *Sirkelparameterisering*

I denne oppgaven skal vi bruke Python til å tegne en sirkel med radius lik 1 som er sentrert i origo. En slik sirkel er gitt ved parameteriseringen  $x(\theta) = \cos(\theta)$  og  $y(\theta) = \sin(\theta)$ . Hvis du lurer på hvorfor sirkelen har denne parameteriseringen er lurt å legge merke til at det vi ønsker å tegne rett og slett er en enhetssirkel. Og hvis man har vinkelen  $\theta$  så er  $x$ - og  $y$ -koordinaten til punktet med den vinkelen gitt ved  $x = \cos(\theta)$  og  $y = \sin(\theta)$



- a) Bruk `linspace` til å opprette en *array*, `theta`, som skal inneholde 100 tall fra 0 til  $2\pi$
- b) Opprett en variabel `x` som inneholder x-koordinatene til punktene på sirkelen (x koordinatene er gitt ved  $\cos(\theta)$ )
- c) Opprett en variabel `y` som inneholder de korresponderende y-koordinatene til punktene på sirkelen (altså  $\sin(\theta)$ ).
- d) Tegn sirkelen med `plot`-kommandoen (Hint: husk `show`).
- e) Se på sirkelen, er det noe som ikke stemmer?
- f) For å få sirkelen til å være sirkulær, ikke elliptisk, så må en centimeter på x-aksen tilsvare en centimeter på y-aksen. For å få til dette kan vi skrive `axis("equal")`. Det tvinger aksene til å ha samme skalering. Legg til denne kommandoen i programmet ditt. Kjør koden og se på plottet, har vi en sirkel nå?

### Løsning oppgave 12 *Sirkelparameterisering*

a)

```
1 from pylab import *
2 theta = linspace(0, 2*pi)
```

**b)**

```
1 x = cos(theta)
```

**c)**

```
1 y = sin(theta)
```

**d)**

```
1 from matplotlib.pyplot import *  
2 plot(x, y)  
3 show()
```

**e)**

```
1 plot(x, y)  
2 axis("equal")  
3 show()
```

Resultatplott:

