

Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 1 av Kodeskolens kræsjkurs i programmering. Tema for første dag er variabler, input, løkker og betingelser. Dersom du står fast er det bare å spørre. God koding!

Variabler og regning

Oppgave 1 *Printing*

- a) Lag et program som skriver ut teksten «Hei, verden!», til skjermen.
- b) Lag et program der du først lagrer navnet ditt i en variabel, og så få programmet ditt til å skrive ut en hilsen direkte til deg.

Løsning oppgave 1 *Printing*

a)

```
1 print("Hei, verden!")
```

b)

```
1 navn = "Maria"
2 print(f"Hei, {navn}!")
```

Oppgave 2 *Kvadrattall*

Kvadrattall er heltall som er blitt kvadrert, altså ganget med seg selv, eller opphøyet i annen. I Python kan du regne ut kvadratet av et tall n enten ved å skrive $n*n$ eller $n**2$.

- a) Skriv et program som spør brukeren om et tall, og deretter skriver ut

kvadratet av tallet brukeren ga.

Pass på at når vi spør om et tall må vi gjøre om svaret til fra brukeren til en tallvariabel ved å skrive `int` på følgende måte: `int(input())`. Dette er fordi «int» står for «integer», som er engelsk for heltall.

- b) Bruk programmet ditt og prøve-feile metoden til å finne det minste tallet som har et kvadrat på over 1000.

Løsning oppgave 2 *Kvadrattall*

a)

```
1 base = float(input("Skriv inn et tall:"))
2 kvadrat = base*base
3 print(f"Kvadratet av {base} er {kvadrat}")
```

- b) Ved å bruke programmet og prøve-feile-metoden kan vi finne ut at det største tallet som har kvadrattall mindre enn 1000 er 31.

Oppgave 3 *Konvertering av temperatur*

I Norge oppgir vi temperaturer i målestokken *Celsius*, men i USA bruker de ofte målestokken *Fahrenheit*. Hvis du finner en kakeoppskrift fra USA kan det for eksempel stå at du skal bake kaken ved 350 grader. Da mener de altså 350°F. Vi vil nå lage et verktøy som kan konvertere denne temperaturen for oss, sånn at vi vet hva vi skal bake kaken ved i Celsius..

For å regne over fra Fahrenheit til Celsius bruker vi formelen:

$$C = \frac{5}{9}(F - 32).$$

Der F er antall grader i Fahrenheit, og C blir antall grader i celsius.

- a) Lag et program som spør brukeren om en temperatur i antall grader Fahrenheit, og skriver ut den tilsvarende temperaturen i antall grader Celsius.

Husk å gjøre om svaret til et tall, med enten `int(input())` eller `float(input())`.

- b) Bruk programmet ditt til å finne ut hvor mange grader Celsius 350°F svarer til. Virker det rimelig å skulle bake en kake ved denne temperaturen?

Programmet du har lagd tar en temperatur i Fahrenheit, og gjør om til Celsius. Men hva om vi ønsker å gå motsatt vei? Om vi ønsker å lage et nytt program som gjør motsatt, så må vi først ha en formel for F .

- c) Klarer du å ta uttrykket

$$C = \frac{5}{9}(F - 32).$$

og løse for F ?

- d) Lag et nytt program som spør brukeren om en temperatur i antall grader celsius, og så skriver ut den tilsvarende temperaturen.
- e) Bruk programmet ditt til å finne frysepunktet og kokepunktet til vann i Fahrenheit målestokken.

Løsning oppgave 3 *Konvertering av temperatur*

a)

```
1 fahrenheit = float(input('Fahrenheit: '))
2 celcius = (5/9)*(fahrenheit-32)
3
4 print(f'{fahrenheit} grader fahrenheit
   tilsvare {celcius:.0f} celcius')
```

b)

```
Fahrenheit: 350
350.0 grader fahrenheit tilsvare 177 celcius
```

177 ° C virker som en rimelig kakebaketemperatur

c)

$$F = \frac{9}{5}C + 32.$$

d)

```
1   celcius = float(input('Celcius: '))
2   fahrenheit = (9/5)*celcius + 32
3
4   print(f'{celcius} grader celcius tilsvare {
      fahrenheit:.0f} fahrenheit')
```

e)

```
Celcius: 0
0.0 grader celcius tilsvare 32 fahrenheit
```

```
Celcius: 100
100.0 grader celcius tilsvare 212 fahrenheit
```

Oppgave 4 *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(Hei , Verden!)
```

b)

```
1 Print("Hei, Verden!")
```

c)

```
1 person = input("Hva heter du?")  
2 print("Hei på deg, {navn}")
```

d)

```
1 pi = 3,14  
2 radius = 4  
3 areal = radius*pi**2
```

Løsning oppgave 4 *Finn fire feil!*

a) Vi må huske på fnuttene våre

```
1 print("Hei, Verden!")
```

b) Vi må bruke liten p i print:

```
1 print("Hei, Verden!")
```

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

```
1 navn = input("Hva heter du?")  
2 print("Hei på deg, {navn}")
```

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

```
1 pi = 3.14  
2 radius = 4  
3 areal = radius*pi**2
```

Løkker

Oppgave 5 *For løkker for tallrekker*

En **for** løkke har følgende syntaks:

```
1 for [løkkevariabel] in [mengde]
2     [kode som skal gjentas]
```

I denne oppgaven skal vi øve på **for**-løkker til å skrive ut noen tallrekker

- a) Skriv ut alle tall fra 0 til (men ikke med) 42
- b) Skriv ut alle partall fra 0 til og med 20
- c) Skriv ut hele 7-gangen fra 0 til og med 70

Løsning oppgave 5 *For løkker for tallrekker*

a)

```
1 for tall in range(43):
2     print(tall)
```

b)

```
1 for tall in range(11):
2     print(2*tall)
```

c)

```
1 for tall in range(11):
2     print(7*tall)
```

Oppgave 6 *Finne kvadrattall og kubikktall*

Lag et program som regner ut kvadrattallet og kubikktallet av tallene 1-5, og printer ut på en linje tallet og tilhørende kvadrat og kubikk. Bruk en **for**-løkke

og legg til mellomrom mellom tallene.

Løsning oppgave 6 *Finne kvadrattall og kubikktall*

```
1 for tall in range(1, 6):  
2     kvadrat = tall**2  
3     kubikk = tall**3  
4     print(f'{tall:4} {kvadrat:4} {kubikk:4}')
```

Oppgave 7 *Renter*

Bank 1 gir fast 3 prosent rente på sin sparekonto. Bank 2, derimot, gir 3,3 prosent rente de første fem årene før de skifter til 2,8 prosent rente. Du skal sette 10000, – i en bank i morgen. I denne oppgaven skal du bruke **for**-løkker til å simulere hva som skjer med pengene i de ulike bankene.

- a) Hvilken bank er best å bruke hvis du skal spare i ti år?
- b) Hvor lenge må du ha pengene i bank 1 for at det skal lønne seg fremfor bank 2?

Naboen din bestemmer seg for å heller sette inn 1000, – hver januar, istedenfor å sette inn en engangssum slik som du gjør.

- c) Hvilken bank er det best for naboen din å bruke hvis han skal spare i ti år?
- d) Hvor lenge må han ha pengene i bank1 for at det skal lønne seg fremfor bank 2?

Løsning oppgave 7 *Renter*

a)

```

1  rente_bank_1 = 3/100
2  rente_bank_2_første_fem_år = 3.3/100
3  rente_bank_2_siste_fem_år = 2.8/100
4
5  innskudd = 10_000
6
7  penger_i_bank_1 = innskudd
8  penger_i_bank_2 = innskudd
9
10 for år in range(10):
11     penger_i_bank_1 *= (1 + rente_bank_1)
12     if år < 5:
13         penger_i_bank_2 *= (1 + rente_bank_2_første_fem_år)
14     else:
15         penger_i_bank_2 *= (1 + rente_bank_2_siste_fem_år)
16
17 print(f"Etter 10 år står det {penger_i_bank_1:.2f}
      kr i bank 1, og {penger_i_bank_2:.2f} kr i
      bank 2.")

```

Etter 10 år står det 13439.16 kr i bank 1, og 13504.15 kr i bank 2.

- b) Denne oppgaven kan løses på to måter, enten ved å prøve og feile ved å endre hvor mange år simuleringen du lagde i forrige oppgave går igjen, eller ved å skrive et program som bruker en løkke for å teste hvor mange år det tar før det lønner seg med bank 1 og å bruke **break** når bank 1 lønner seg. Under er et eksempel på et slikt program

```

1  rente_bank_1 = 3/100
2  rente_bank_2_første_fem_år = 3.3/100
3  rente_bank_2_siste_fem_år = 2.8/100
4
5  innskudd = 10_000
6
7  penger_i_bank_1 = innskudd
8  penger_i_bank_2 = innskudd
9

```



```

10 for år in range(100):
11     penger_i_bank_1 *= (1 + rente_bank_1)
12     if år < 5:
13         penger_i_bank_2 *= (1 + rente_bank_2_før
14                             rste_fem_år)
15     else:
16         penger_i_bank_2 *= (1 + rente_bank_2
17                             _siste_fem_år)
18
19     if penger_i_bank_1 > penger_i_bank_2:
20         print(f"Etter {år+1} år lønner det seg å
21               bruke bank 1, for da står det {
22               penger_i_bank_1:.2f} kr i bank 1, og {
23               penger_i_bank_2:.2f} kr i bank 2.")
24         break
25 else:
26     print("Det tar mer enn 100 år før det lønner
27           seg å bruke bank 2")

```

Etter 13 år lønner det seg å bruke bank 1, for da står det 14685.34 kr i bank 1, og 14670.55 kr i bank 2.

c)

```

1  rente_bank_1 = 3/100
2  rente_bank_2_første_fem_år = 3.3/100
3  rente_bank_2_siste_fem_år = 2.8/100
4
5  innskudd = 0
6  årlig_innskudd = 1000
7
8  penger_i_bank_1 = innskudd
9  penger_i_bank_2 = innskudd
10
11 for år in range(10):
12     penger_i_bank_1 += årlig_innskudd
13     penger_i_bank_2 += årlig_innskudd
14     penger_i_bank_1 *= (1 + rente_bank_1)
15     if år < 5:

```

```

16         penger_i_bank_2 *= (1 + rente_bank_2_før
17                               rste_fem_år)
18     else:
19         penger_i_bank_2 *= (1 + rente_bank_2
20                               _siste_fem_år)
21
22 print(f"Etter 10 år står det {penger_i_bank_1:.2f}
23       kr i bank 1, og {penger_i_bank_2:.2f} kr i
24       bank 2.")

```

Etter 10 år står det 11807.80 kr i bank 1, og 11770.25 kr i bank 2.

d)

```

1  rente_bank_1 = 3/100
2  rente_bank_2_første_fem_år = 3.3/100
3  rente_bank_2_siste_fem_år = 2.8/100
4
5  innskudd = 1000
6  årlig_innskudd = 1000
7
8  penger_i_bank_1 = innskudd
9  penger_i_bank_2 = innskudd
10
11 for år in range(100):
12     penger_i_bank_1 *= (1 + rente_bank_1)
13     if år < 5:
14         penger_i_bank_2 *= (1 + rente_bank_2_før
15                               rste_fem_år)
16     else:
17         penger_i_bank_2 *= (1 + rente_bank_2
18                               _siste_fem_år)
19
20     penger_i_bank_1 += årlig_innskudd
21     penger_i_bank_2 += årlig_innskudd
22     if penger_i_bank_1 > penger_i_bank_2:
23         print(f"Etter {år+1} år lønner det seg å
24               bruke bank 1, for da står det {
25               penger_i_bank_1:.2f} kr i bank 1, og {

```

```

22         penger_i_bank_2:.2f} kr i bank 2.")
23         break
24     else:
25         print("Det tar mer enn 100 år før det lønner
        seg å bruke bank 2")

```

Etter 9 år lønner det seg å bruke bank 1, for da står det 11463.88 kr i bank 1, og 11449.66 kr i bank 2.

Betingelser

Oppgave 8 *Finn størst tall*

I denne oppgaven skal vi øve på **if**-tester ved å finne det største av to tall.

- Skriv et program som ber brukeren om å skrive inn to tall og lagre dem i to variabler, `tall1` og `tall2`.
- Utvid programmet ditt ved hjelp av **if** slik at dersom `tall2` er større enn `tall1`, skriver du ut en beskjed om at `tall1` er størst.
- Utvid programmet ditt videre ved hjelp av **elif**, slik at dersom `tall2` er større enn `tall1`, får brukeren beskjed om at `tall2` er størst.
- Fullfør til slutt koden ved å legge til en **else**-blokk, slik at dersom tallene er like, vil brukeren få beskjed om det.

Løsning oppgave 8 *Finn størst tall*

a)

```

1  tall1 = float(input('Si et tall'))
2  tall2 = float(input('Si et annet tall'))
3  if tall1 > tall2:
4      print(f'{tall1} er størst')
5  elif tall2 > tall1:

```

```

6     print(f'{tall2} er størst')
7 else:
8     print('Tallene er like!')

```

Oppgave 9 *ABC-formelen*

ABC-formelen for å løse annengradsformler er som følger:

La a , b og c være reelle tall, hvor $a \neq 0$. Da har likningen $ax^2 + bx + c = 0$ løsningene

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Lag et program som finner løsningene til en annengradslikning med $a = 1$, $b = 2.5$ og $c = 1$.
- Modifiser programmet ditt så det spør brukeren om verdier for a , b og c og skriver ut de tilhørende løsningene.
- Dersom $b^2 - 4ac < 0$ har den tilhørende annengradslikningen ingen reell løsning. Bruk en **if**-betingelse til å informere brukeren om at det ikke finnes noen reell løsning dersom $b^2 - 4ac < 0$
- Dersom $b^2 - 4ac = 0$ har likningen kun en løsning. Bruk en **elif** til å sjekke om dette er tilfelle og i så tilfelle informere brukeren om at det kun er en løsning

Løsning oppgave 9 *ABC-formelen*

a)

```

1  from math import sqrt
2
3  a = 1
4  b = 2.5
5  c = 1
6  løsning1 = (-b + sqrt(b**2 - 4*a*c))/(2*a)

```

```

7  løsning2 = (-b - sqrt(b**2 - 4*a*c))/(2*a)
8
9  print(f'Løsningene for likningen {a}x^2 + {b}x + {
    c} = er {løsning1} og {løsning2}')

```

b)

```

1  from math import sqrt
2
3  a = float(input('Hva er a? '))
4  b = float(input('Hva er b? '))
5  c = float(input('Hva er c? '))
6
7  løsning1 = (-b + sqrt(b**2 - 4*a*c))/(2*a)
8  løsning2 = (-b - sqrt(b**2 - 4*a*c))/(2*a)
9
10 print(f'Løsningene for likningen {a}x^2 + {b}x + {
    c} = er {løsning1} og {løsning2}')

```

c)

```

1  from math import sqrt
2
3  a = float(input('Hva er a? '))
4  b = float(input('Hva er b? '))
5  c = float(input('Hva er c? '))
6
7  rot_del = b**2 - 4*a*c
8
9  if rot_del < 0:
10     print(f' {a}x^2 + {b}x + {c} = 0 har dessverre
        ingen reelle løsninger')
11 else:
12     løsning1 = (-b + sqrt(rot_del))/(2*a)
13     løsning2 = (-b - sqrt(rot_del))/(2*a)
14     print(f'Løsningene for likningen {a}x^2 + {b}x
        + {c} = 0 er {løsning1} og {løsning2}')

```

d)

```

1  from math import sqrt
2
3  a = float(input('Hva er a? '))
4  b = float(input('Hva er b? '))
5  c = float(input('Hva er c? '))
6
7  rot_del = b**2 - 4*a*c
8
9  if rot_del < 0:
10     print(f' {a}x^2 + {b}x + {c} = 0 har dessverre
        ingen reelle løsninger')
11 elif rot_del == 0:
12     løsning = -b/(2*a)
13     print(f'Løsningen for likningen {a}x^2 + {b}x
        + {c} = 0 er {løsning}')
14 else:
15     løsning1 = (-b + sqrt(rot_del))/(2*a)
16     løsning2 = (-b - sqrt(rot_del))/(2*a)
17     print(f'Løsningene for likningen {a}x^2 + {b}x
        + {c} = 0 er {løsning1} og {løsning2}')
```