

## Bonusoppgaver

I denne oppgavesamlingen finner du et knippe med bonusoppgaver for dag 2 dersom du vil ha litt mer å bryne deg på. God koding!

### Funksjoner

#### Oppgave 1 *Beste pizzapris*

Hallgeir har lyst til å kjøpe pizza og sjekker prisen på nettet. Der står det at pizzaen koster 200 kroner for stor pizza med diameter 40 cm og 110 for liten pizza med diameter 27 cm. Hallgeir har lyst til å velge den pizzaen som er billigst per kvadratcentimeter med pizza og han vet at arealet til en sirkel er gitt ved:

$$\text{Areal} = \pi \times \text{radius}^2$$

For enkelhetsskyld så sier vi at pi er 3.14 istedenfor å importere den fra noe bibliotek.

- a) Opprett en variabel, `diameter` som skal ha verdien 40
- b) For å bruke formelen for arealet av en sirkel trenger vi radiusen. Vi vet at diameteren til en sirkel er det dobbelte av radiusen. Opprett ennå en variabel, `radius`, som skal være halvparten av diameteren.
- c) Regn ut arealet av sirkelen, lagre resultatet i en variabel, `areal` og skriv ut hvor mange kvadratcm med pizza den store pizzaen inneholder.
- d) Regn ut hvor mye pizzaen koster i kroner per kvadratcentimeter med pizza. Skriv svaret ut til skjermen.
- e) Opprett et funksjon `kvadratcentimeter_pris(diameter, pris)` som tar inn diameteren til en pizza og radiusen og returnerer prisen per kvadratcentimeter.
- f) Bruk funksjonen du lagde i oppgaven over til å regne ut kvadratcentimeterprisen til den lille pizzaen.

- g) Hvilken pizza skal Hallgeir velge dersom han vil ha pizzaen som er billigst per kvadratcentimeter?

### Løsning oppgave 1 *Beste pizzapris*

a)

```
1 diameter = 40
```

b)

```
1 radius = diameter/2
```

c)

```
1 areal = 3.14 * radius**2
2 print(areal)
```

d)

```
1 pris = 200
2 kr_per_kvadratcentimeter = pris/areal
3 print(kr_per_kvadratcentimeter)
```

e)

```
1 def kvadratcentimeter_pris(diameter, pris):
2     radius = diameter/2
3     areal = 3.14 * radius**2
4     kr_per_kvadratcentimeter = pris/areal
5     return kr_per_kvadratcentimeter
```

f)

```
1 print(kvadratcentimeter_pris(27, 110))
```

g)

```
1 #Pizzaen med diameter 40cm og pris 200 kr gir
   billigst kvadratcentimeter.
```

## Skilpaddeprogrammering

## Oppgave 2 *Plotte med Turtle*

For å lage plot trenger vi funksjon som skalerer et tall fra en måleverdi til et koordinatsystem.

- a) Lag en funksjon `omskaler(tall, a1, b1, a2, b2)` Denne funksjonen skal transformere et tall på denne måten:

$$\text{omskaler}(x, a_1, b_1, a_2, b_2) = a_2 + x \frac{b_2 - a_2}{b_1 - a_1} + a_1 \frac{b_2 - a_2}{b_1 - a_1} \quad (1)$$

Denne funksjonen tar inn et tall mellom `a1` og `b1` og gir ut et tall mellom `a2` og `b2`. Dersom `tall` er lik `a1` vil omskaleringen være lik `a2`. Tilsvarende, dersom `tall` er lik `b1`, vil omskaleringen være lik `b2`. Hvis `tall` ligger midt mellom `a1` og `a2`, så vil resultatet av omskaleringen ligge midt mellom `a1` og `b1`.

- b) Nå skal vi teste at omskaleringsfunksjonen vår virker. Sjekk at

$$\text{omskaler}(0, 0, 1, 10, 20) = 10 \quad (2)$$

$$\text{omskaler}(1, 0, 1, 10, 20) = 20 \quad (3)$$

$$\text{omskaler}(0.5, 0, 1, 10, 20) = 15 \quad (4)$$

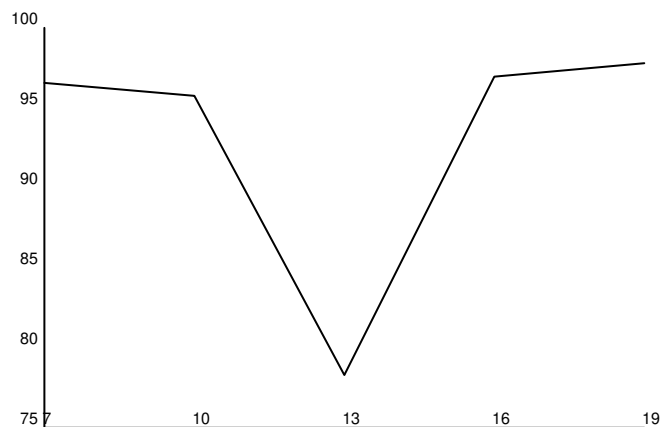
$$\text{omskaler}(2, 0, 10, 100, 200) = 120 \quad (5)$$

Tabellen under viser målinger av internetthastigheten i en leilighet i Grünerløkka på forskjellige tidspunkt i løpet av en dag

Klokkeslett [time]	Nedlastingshastighet [Mbps]
7	96.56
10	95.75
13	78.30
15	96.95
19	97.79

- c) Lag en liste `hastigheter` som inneholder måleverdiene for internetthastighet og en liste `timer` som inneholder timene hastighetene ble målt

- d) Opprett en skilpadde med `turtle.Turtle()`
- e) Bruk en løkke til å gå igjennom hastighetsmålingene og timene. Inne i løkka skal du omskalere hver hastighetsmåling til en y-verdi mellom 0 og 200 og hver timeverdi til en x-verdi mellom -300 og 300.
- f) Bruk `goto` inne i løkka til å gå til en x og y verdi med skilpadda di og tegne en strek (hint: Det kan være lurt å bruke `penup` og `pendown` for å flytte skilpadda til første målepunktet uten å tegne en strek).
- g) **Bonusoppgave:** Prøv å gjenskape figuren under ved å tegne opp koordinataksene og bruke `write` funksjonen til å tegne opp tallverdier på aksene.



## Løsning oppgave 2 *Plotte med Turtle*

a)

```
1 def omskaler(tall, a1, b1, a2, b2):
2     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2 - a2)/(b1 - a1)
```

b)

```

1 def omskaler(tall, a1, b1, a2, b2):
2     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
      - a2)/(b1 - a1)
3
4 print(omskaler(0, 0, 1, 10, 20))
5 print(omskaler(1, 0, 1, 10, 20))
6 print(omskaler(0.5, 0, 1, 10, 20))
7 print(omskaler(2, 0, 10, 100, 200))

```

```

10.0
20.0
15.0
120.0

```

c)

```

1 def omskaler(tall, a1, b1, a2, b2):
2     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
      - a2)/(b1 - a1)
3
4 måletidspunkt = [7, 10, 13, 16, 19]
5 internetthastighet = [96.56, 95.75, 78.3, 96.95, 9
      7.79]

```

d)

```

1 import turtle
2
3 def omskaler(tall, a1, b1, a2, b2):
4     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
      - a2)/(b1 - a1)
5
6 måletidspunkt = [7, 10, 13, 16, 19]
7 internetthastighet = [96.56, 95.75, 78.3, 96.95, 9
      7.79]
8
9
10 antall_målinger = len(måletidspunkt)
11 penn = turtle.Turtle()
12

```

```

13 turtle.done()
14 turtle.bye() # Nødvendig hvis vi bruker Spyder

```

e)

```

1 import turtle
2
3 def omskaler(tall, a1, b1, a2, b2):
4     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
5         - a2)/(b1 - a1)
6
7 måletidspunkt = [7, 10, 13, 16, 19]
8 internetthastighet = [96.56, 95.75, 78.3, 96.95, 9
9     7.79]
10
11 antall_målinger = len(måletidspunkt)
12 penn = turtle.Turtle()
13
14 for målingsnummer in range(antall_målinger):
15     x = omskaler(måletidspunkt[målingsnummer], 7,
16         19, -150, 150)
17     y = omskaler(internetthastighet[målingsnummer
18 ], 75, 100, 0, 200)
19
20 turtle.done()
21 turtle.bye() # Nødvendig hvis vi bruker Spyder

```

f)

```

1 import turtle
2
3 def omskaler(tall, a1, b1, a2, b2):
4     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
5         - a2)/(b1 - a1)
6
7 måletidspunkt = [7, 10, 13, 16, 19]
8 internetthastighet = [96.56, 95.75, 78.3, 96.95, 9
9     7.79]
10
11

```

```

9
10 antall_målinger = len(måletidspunkt)
11 penn = turtle.Turtle()
12 penn.penup()
13 for målingsnummer in range(antall_målinger):
14     x = omskaler(måletidspunkt[målingsnummer], 7,
15                 19, -150, 150)
16     y = omskaler(internetthastighet[målingsnummer
17                 ], 75, 100, 0, 200)
18     penn.goto(x, y)
19     penn.pendown()
20
21 turtle.done()
22 turtle.bye() # Nødvendig hvis vi bruker Spyder

```

g)

```

1 import turtle
2
3 def omskaler(tall, a1, b1, a2, b2):
4     return tall*(b2 - a2)/(b1 - a1) + a2 - a1*(b2
5         - a2)/(b1 - a1)
6
7 måletidspunkt = [7, 10, 13, 16, 19]
8 internetthastighet = [96.56, 95.75, 78.3, 96.95, 9
9     7.79]
10
11 antall_målinger = len(måletidspunkt)
12 penn = turtle.Turtle()
13 penn.penup()
14 for målingsnummer in range(antall_målinger):
15     x = omskaler(måletidspunkt[målingsnummer], 7,
16                 19, -150, 150)
17     y = omskaler(internetthastighet[målingsnummer
18                 ], 75, 100, 0, 200)
19     penn.goto(x, y)
20     penn.pendown()
21
22

```

```

20 penn.penup()
21 penn.goto(-150, 0)
22 penn.pendown()
23 for målingsnummer in range(antall_målinger):
24     klokkeslett = måletidspunkt[målingsnummer]
25     x = omskaler(klokkeslett, 7, 19, -150, 150)
26     penn.goto(x, 0)
27     penn.write(klokkeslett)
28
29
30 penn.penup()
31 penn.goto(-150, 0)
32 penn.pendown()
33 for målingsnummer in range(antall_målinger):
34     hastighet = 75 + målingsnummer * 25/antall_målinger
35     y = omskaler(hastighet, 75, 100, 0, 200)
36     penn.goto(-150, y)
37     penn.write(hastighet, align="right")
38 penn.hideturtle()
39 turtle.done()
40 turtle.bye() # Nødvendig hvis vi bruker Spyder

```