Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 2 av Kodeskolens kræsjkurs i programmering for lærere.

Tema for andre dag er *funksjoner*, *plot* og *sammensatte eksempler*. 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 vanskelig. 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!

Oppgave 1 *Absoluttverdifunksjon*

Absoluttverdien til et tall, a, er gitt ved:

$$\mid a \mid = \begin{cases} a, & \text{hvis } a \geq 0 \\ -a, & \text{hvis } < 0 \end{cases}$$

a) Lag en funksjon som tar inn ett tall, a og returnerer absoluttverdien til tallet.

Løsning oppgave 1 *Absoluttverdifunksjon*

a)

```python
def absoluttverdi(tall):
    if tall > 0:
        return tall
    else:
        return -tall
```

Oppgave 2 *Midtpunktfunksjon*

Midtpunktet mellom to tall, a og b er gitt ved:

$$\frac{a+b}{2}$$

a) Lag en funksjon som tar inn to tall, `a` og `b` og returnerer midtpunktet mellom dem

b) Bruk funksjonen til å finne midtpunktet mellom 34 og 86

Løsning oppgave 2 *Midtpunktfunksjon*

a)

```python
def midtpunkt(a, b):
    return (a + b)/2
```

b)

```python
a = 34
b = 86

m = midtpunkt(a, b)
print(f'Midpunktet mellom {a} og {b} er {m}')
```

Oppgave 3 *Kanonkule*

Året er 1384 i en fiktiv borg i sør-England. Franskmennene har omringet borgen og kanonene er rullet frem. Dette har åpenbart gjort Lorden som eier borgen nærvøs, og han spør deg, hans fremste lærdmann om kanonkulene vil nå borgen med den avstanden de har.

Heldigvis har spionene til kongen klart å stjele en av kanonene, og etter mye om og men, har du funnet ut at kulene forlater kanonen med en fart på ca 50 meter i sekundet. Kanonene er låst i en 30 graders vinkel, så posisjonen til en kanonkule kan beskrives med disse funksjonene

$$x(t) = 25\frac{m}{s}t$$
$$y(t) = 45\frac{m}{s}t - 5\frac{m}{s^2}t^2,$$

hvor $x(t)$ er hvor langt kanonkula har bevegd seg bortover (i meter) t sekund etter at den ble skutt og $y(t)$ er høyden til kanonkula (i meter) t sekund etter at den ble skutt.

a) Definer en funksjon `kanonkule_x(t)` som returnerer hvor langt kanonkula har bevegd seg bortover etter `t` sekund.

b) Definer en funksjon `kanonkule_y(t)` som returnerer høyden til kanonkula etter `t` sekund.

c) Definer en funksjon `kanonkule_posisjon(t)` som returnerer både hvor langt kula har bevegd seg bortover og høyden til den etter `t` sekund.

d) Kanonene er plassert 200 meter unna borgen (i x retning), og er i samme høyde som borgen. Bruk funksjonen vi akkurat lagde og prøv ut forskjellige verdier for `t`. Treffer kanonkula borgen?

e) (Bonusoppgave) Lag et program som bruker `input` funksjonen (som vi lærte om forrige kursdag) til å be om et antall sekunder og printer ut posisjonen til kanonkula så lang tid etter at den ble skutt.

f) (Bonusoppgave) Bruk en løkke til å printe ut posisjonen til kanonkula de ti første sekundene kanonkula er i lufta.

Løsning oppgave 3 *Kanonkule*

a)

```
def kanonkule_x(t):
    return 25*t
```

b)

```
def kanonkule_y(t):
    return 45*t - 5*t**2
```

c)

```
def kanonkule_posisjon(t):
    x = kanonkule_x(t)
    y = kanonkule_y(t)
    return x,y
```

d)

```
tid = 10
x, y = kanonkule_posisjon(tid)
print(f"lengde: {x}m  høyde:{y}m")
```

e)

```
tid = float(input("Skriv inn antall sekunder etter
    avfyrt skudd:"))
x, y = kanonkule_posisjon(tid)
print(f"lengde: {x}m  høyde:{y}m")
```

f)

```
print("Posisjon de første 10 sekundene:")
for tid in range(0,11):
    x,y = kanonkule_posisjon(tid)
    print(f"tid:{tid:4.1f}s lengde:{x:3.0f}m  hø
       yde:{y:3.0f}m")
```

```
Posisjon de første 10 sekundene:
tid: 0.0s lengde:  0m  høyde:  0m
tid: 1.0s lengde: 25m  høyde: 40m
tid: 2.0s lengde: 50m  høyde: 70m
tid: 3.0s lengde: 75m  høyde: 90m
tid: 4.0s lengde:100m  høyde:100m
tid: 5.0s lengde:125m  høyde:100m
tid: 6.0s lengde:150m  høyde: 90m
tid: 7.0s lengde:175m  høyde: 70m
```

```
tid: 8.0s lengde:200m  høyde: 40m
tid: 9.0s lengde:225m  høyde:  0m
tid:10.0s lengde:250m  høyde:-50m
```

Her ser vi at når kula er framme ved borgen (200m) er den 40 meter over bakken. Den treffer bakken ved 225m, altså 25m bortenforborgmuren, altså treffer den borgen. (med mindre det er en veldig liten borg)

Oppgave 4 *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:

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

a) Opprett en variabel, `pizza_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, `pizza_radius`, som skal være halvparten av diameteren.

c) Regn ut arealet av sirkelen, lagre resultatet i en variabel, `pizza_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 `pizza_kvadratcentimeter_pris(pizza_diameter, pizza_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 4 *Beste pizzapris*

a)
```
pizza_diameter = 40
```

b)
```
pizza_radius = pizza_diameter/2
```

c)
```
pizza_areal = 3.14 * pizza_radius**2
print(pizza_areal)
```

d)
```
pizza_pris = 200
kr_per_kvadratcentimeter = pizza_pris/pizza_areal
print(kr_per_kvadratcentimeter)
```

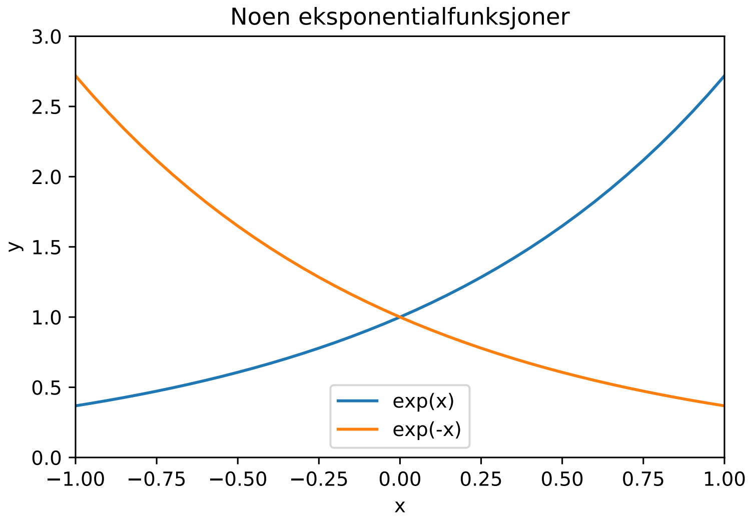
e)
```
def pizza_kvadratcentimeter_pris(pizza_diameter,
    pizza_pris):
  pizza_radius = pizza_diameter/2
  pizza_areal = 3.14 * pizza_radius**2
  kr_per_kvadratcentimeter = pizza_pris/
     pizza_areal
  return kr_per_kvadratcentimeter
```

f)
```
print(pizza_kvadratcentimeter_pris(27, 110))
```

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

Oppgave 5 *Plotting*

I denne oppgaven skal vi ende opp med et plot som ser slik ut



a) Opprett en array med tallrekken som starter på 0, slutter på 1 og har et mellomrom på 0.05 mellom hvert element. Lagre den arrayen i en variabel du kaller `x`.

b) Opprett en array som inneholder e^x for alle verdier i `x` variabelen din (hint: husk `exp` funksjonen i `pylab`) og lagre dette arrayet i en array med navnet `y1`.

c) Lag et plot som viser x på førsteaksen og e^x på andreaksen for x-verdier mellom 0 og 1.

d) Pynt på plottet ved å sette grenser for x-aksen og y-aksen med `xlim` og `ylim` funksjonene. Sjekk at aksene dine har endret seg siden oppgave **b)**.

e) Gi aksene dine merkelapper (f.eks x og y) med `xlabel` og `ylabel` funksjonene.

f) Gi plottet ditt en tittel (f.eks $y = exp(x)$) med `title` funksjonen.

g) Opprett en ny variabel, `y2` og som inneholder e^{-x} for alle verdier i `x` arrayet ditt.

h) Plot $y1$ og $y2$ i samme plot.

i) (Bonusoppgave) Gi kurvene dine merkelapper som forteller hva de representerer og vis frem disse merkelappene med `legend` funksjonen.

Løsning oppgave 5 *Plotting*

a)

```
from pylab import arange

x = arange(0,1,0.05)
```

b)

```
from pylab import arange, exp

x = arange(0,1,0.05)
y1 = exp(x)
```

c)

```
from pylab import arange, exp, plot, show

x = arange(-1,1.05,0.05)
y1 = exp(x)
plot(x,y1)
show()
```

d)

```
from pylab import arange, exp, plot, show, xlim, ylim

x = arange(-1,1.05,0.05)
y1 = exp(x)
```

```
plot(x,y1)
xlim(-1,1)
ylim(0,3)
show()
```

e)

```
from pylab import arange, exp, plot, show, xlim,
   ylim, xlabel, ylabel

x = arange(-1,1.05,0.05)
y1 = exp(x)
plot(x,y1)
xlim(-1,1)
ylim(0,3)
xlabel('x')
ylabel('y')
show()
```

f)

```
from pylab import arange, exp, plot, show, xlim,
   ylim, xlabel, ylabel, title

x = arange(-1,1.05,0.05)
y1 = exp(x)
plot(x,y1)
xlim(-1,1)
ylim(0,3)
xlabel('x')
ylabel('y')
title('$y=e^x$')
show()
```

g)

```
from pylab import arange, exp, plot, show, xlim,
   ylim, xlabel, ylabel, title

x = arange(-1,1.05,0.05)
```

```
y1 = exp(x)
y2 = exp(-x)
plot(x,y1)
xlim(-1,1)
ylim(0,3)
xlabel('x')
ylabel('y')
title('$y=e^x$')
```

h)

```
from pylab import arange, exp, plot, show, xlim,
    ylim, xlabel, ylabel, title

x = arange(-1,1.05,0.05)
y1 = exp(x)
y2 = exp(-x)
plot(x,y1)
plot(x,y2)
xlim(-1,1)
ylim(0,3)
xlabel('x')
ylabel('y')
title('$y=e^x$ og $y=e^{-x}$')
```

i)

```
from pylab import arange, exp, plot, show, xlim,
    ylim, xlabel, ylabel, title, legend

x = arange(-1,1.05,0.05)
y1 = exp(x)
y2 = exp(-x)
plot(x,y1, label='$y=e^x$')
plot(x,y2, label='$y=e^{-x}$')
xlim(-1,1)
ylim(0,3)
xlabel('x')
ylabel('y')
title('$y=e^x$ og $y=e^{-x}$')
```

```
legend()
show()
```

Oppgave 6 *Grafisk løsning av likning*

I denne oppgaven skal vi løse en likning *grafiks*, dvs. ved å lage og lese av en graf.

a) Definer en funksjon som returnerer $f(x) = sin(x^2)$

b) Definer en funksjon som returnerer $g(x) = x^2 + x/5 - exp(-x)$

c) Opprett arrays for å lagre x-verdier mellom + og 1.7, samt verdier man får ved å kalle på f og g med disse x-verdiene.

d) Tegn grafen til $f(x)$ for x-verdier mellom 0 og 1.7

e) Tegn grafen til $g(x)$ for x-verdier mellom 0 og 1.7 i samme plot som $f(x)$.

f) For hvilken x er $f(x)$ og $g(x)$ like? (ca)

g) Marker punktet hvor $f(x)$ og $g(x)$ er like med en rød sirkel

Løsning oppgave 6 *Grafisk løsning av likning*

a)

```
from pylab import *

def f(x):
    return sin(x**2)
```

b)

```
def g(x):
```

```python
    return x**2 + x/5 - exp(-x)
```

c)

```python
x = arange(0, 1.7 + 0.05, 0.05)
```

d)

```python
y1 = f(x)
plot(x, y1)
show()
```

e)

```python
y2 = g(x)
plot(x, y1)
plot(x, y2)
show()
```

f) Vi leser av grafen at funksjonene er (omtrent) like for 1

g)

```python
plot(x, y1)
plot(x, y2)
plot(1, g(1), 'ro')
show()
```

Oppgave 7 *Fortsettelse kanonkule*

Fortsettelse fra kanonkuleoppgaven. Vi har akkurat fortalt lorden i borgen vår at kanonene kan treffe og han tror ikke på oss. Derfor bestemmer vi oss for å lage en tegning hvor vi viser kulas bane. Om du ikke fikk til forrige kanonkuleoppgave kan du bruke denne funksjonen

```python
def kanonkule_posisjon(t):
    x = 25*t              # kanonkule_x(t)
    y = 45*t - 5*t**2     # kanonkule_y(t)

    return x, y
```

a) Lag en array, `t` som inneholder en tallrekke som starter på 0, slutter på 9 og har en steglenge på 0.1 (Tips: hvis du vil at tallrekka skal slutte på 9 er det lurt å sette sluttposisjonen til `slutt+steglengde`, som i dette tilfellet er 9.1).

b) Bruk `kanonkule_posisjon(t)` til å regne ut kanonkulas `x`-koordinater og `y`-koordinater for hvert tidspunkt i `t` array-et vårt.

c) Lag et plot som viser `x`-posisjon på førsteaksen on `y`-posisjon på andreaksen.

d) Ved å løse likningen $x(t) = 200$ ser vi at kanonkula treffer borgen etter åtte sekund. Marker punktet $(x(8), y(8))$ med et rødt punkt.

Løsning oppgave 7 *Fortsettelse kanonkule*

a)

```
from pylab import *
t = arange(0,9.1,0.1)
```

b)

```
x_pos, y_pos = kanonkule_posisjon(t) #får to
   arrays med x og y posisjoner
```

c)

```
plot(x_pos,y_pos) #plott banen
```

d)

```
treff_x, treff_y = kanonkule_posisjon(8)
#finn treffpunktet ved 8 sek
plot(treff_x,treff_y,'ro')
#plott punktet, 'ro' for å få et rødt punkt
show()
```