

Oppgave 1: Printing

1a) Lag et program som skriver ut teksten “Hello, World!” til skjermen.

1b) Lag et program hvor du lagrer navnet ditt som en variabel, og deretter skriver ut lengden på navnet ditt med kommandoen `len(name)` .

1c) Lag et program som spør brukeren om navnet dems, og deretter skriv en beskjed tilbake som bruker navnet de har gitt. *Hint:* Du kan spørre brukeren et spørsmål med `input` .

Oppgave 2: Lister

Ta en titt på følgende program:

```
x = [1]
x.append(2)
x.append(3)

print(x)

x[0] = 4

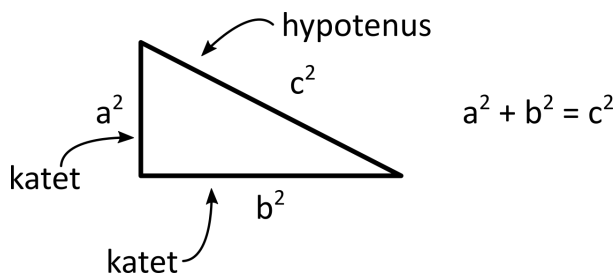
print(x)
print(len(x))
print(type(x))
print(type(x[0]))
```

Gå igjennom programmet for hånd og prøv å forutsi hva programmet skriver ut. Test så ved å kjøre koden.

Oppgave 3: Kvadrattall

Skriv en funksjon som tar et tall som argument inn, og returnerer kvadrattallet. **Hint:** Et kvadrattall er et tall ganget med seg selv, for eksempel så er $3 \times 3 = 9$, så 9 er kvadrattallet til 3.

Oppgave 4: Pytagoras



Oppgave a)

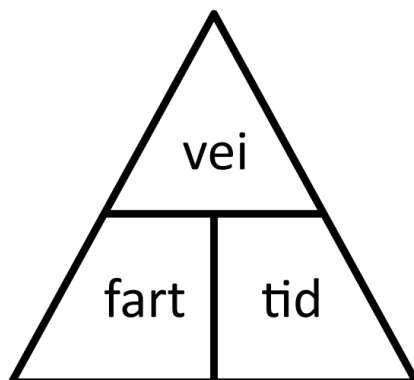
Skriv en funksjon som bruker Pytagoras til å regne ut lengden hypotenusen av en rettvinklet trekant, gitt lengden på de to katetene som inputen. Kall funksjonen din for `hypotenus(a, b)` .

Oppgave b)

Skriv en funksjon som bruker Pytagoras til å regne ut lengden på den ene kateten, gitt lengden på hypotenusen og lengden på den andre kateteten som input. Kall funksjonen din `katet(b, c)`.

Oppgave 5: vei-fart-tid kalkulator

En viktig formel for å beskrive bevegelse er vei-fart-tid formelen. Egentlig er det tre formler i et, for vi kan stokke om på den avhengig av hva vi ønsker å regne ut. Av og til tegnes vei-fart-tid formelen som en pyramide:



Oppgave a) Skriv opp uttrykket for å regne ut *vei*, gitt *fart* og *tid*. Lag en funksjon som bruker formelen, kall denne `vei(fart, tid)`. Vi vil at *fart* skal oppgis i kilometer i timen, *vei* i antall kilometer og *tid* i antall minutter, da må du huske å regne om tiden fra antall timer til antall minutter.

Oppgave b) Skriv opp uttrykket for å regne ut *fart*, gitt *vei* og *tid*. Lag en funksjon som bruker formelen, kall denne `fart(vei, tid)`.

Oppgave c) Skriv opp uttrykket for å regne ut *tid*, gitt *vei* og *fart*. Lag en funksjon som bruker formelen, kall denne `tid(vei, fart)`.

Oppgave d) Du skal nå lage en vei-fart-tid-kalkulator. Når dette programmet kjører skal det først spørre brukeren om de ønsker å regne ut *vei*, *fart*, eller *tid*. Avhengig av svaret skal så programmet spørre om de to størrelsene, og så regne ut og skrive ut svaret.

Oppgave 6: Summere tallrekker

La oss si vi ønsker å summere tallene fra 1 til og med 100. Dette blir litt slitsomt å gjøre for hånd. Så la oss bruke programmering.

Oppgave a) Bruk en løkke til å finne summen av tallene 1 til 10. Sjekk at du får riktig svar ved å finne den samme summen for hånd.

Oppgave b) Endre på programmet ditt så du finner summen av tallene fra 1 til og med 100.

Den kjente matematikeren Gauss ble gitt denne oppgaven av sin lærer på 1700-tallet. Læreren tenkte nok at dette skulle holde Gauss opptatt en god stund med å legge sammen tall etter tall. Gauss hadde ikke tilgang på en datamaskin, så han kunne ikke automatisere jobben, men ha la merke til et mønster i tallene. Gauss la merke til at om vi starter på begge endene av rekka får vi et mønster

$$1 + 100 = 101,$$

$$2 + 99 = 101,$$

$$3 + 98 = 101,$$

...

$$50 + 51 = 101.$$

Fra dette mønsteret klarte Gauss å finne en formel for summen av tallene fra 1 til n , og uttrykket hans var

$$s = \frac{n(n+1)}{2}.$$

Oppgave c) Forklar hvorfor formelen beskriver mønsteret Gauss ser. *Hint:* Hva er $(n+1)$ hvis n er 100? Og hvorfor tar vi $n/2$ og ganger med dette tallet?

Oppgave d) Lag en funksjon som bruker formelen til Gauss, kall funksjonen din `gaussum(n)`.

Oppgave d) Bruk `gaussum` funksjonen du lagde og sammenlign med svaret du fikk i oppgave (b). Gir de to samme svar?

Oppgave e) Finn summen av tallene fra 1 til og med 10 millioner. Bruk først løkkeløsningen din, og så `gaussum` -funksjonen din. Hvilken måte er raskest? Hvorfor tror det er sånn?

Oppgave 7: Løkker for hånd

For hver løkke, gå igjennom for hånd og forutsi hva som skrives ut. Etterpå kan du kjøre løkkene og se hva som

Oppgave a)

```
total = 0
for i in range(-5, 5):
    total += i
print(total)
```

Oppgave b)

```
total = 1
for _ in range(5):
    total *= total
print(total)
```

Oppgave c)

```
total = 1
while total < 10:
    total += 2
print(total)
```

Oppgave d)

```
total = 100
antall_doblinger = 0

while total < 500:
    total *= 2
    antall_doblinger += 1

print(antall_doblinger)
```

For hver løkke, kjør løkka og se om du hadde rett.
