

# Kræsjkurs i programmering med Python: dag 1

2. mars 2023



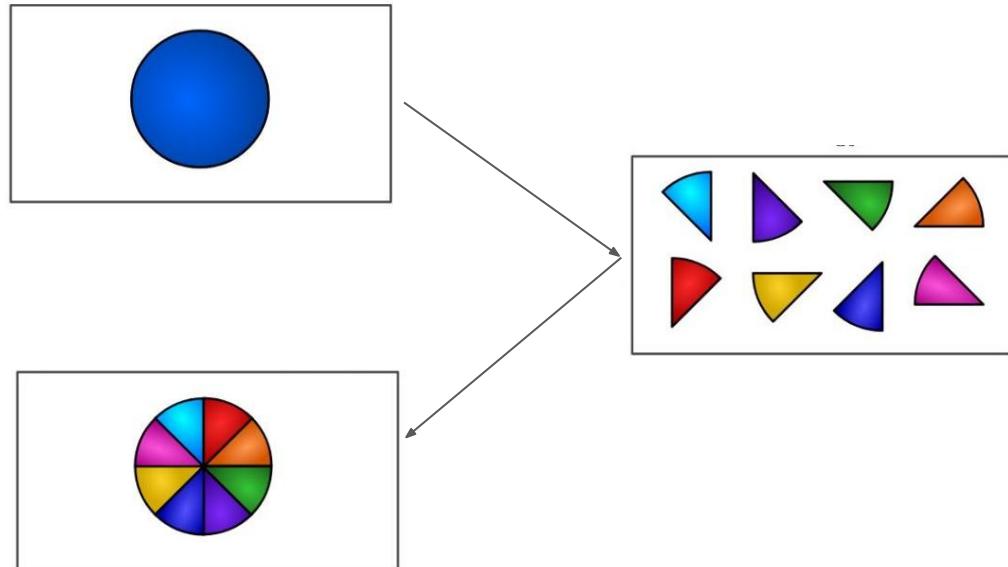
# Programmering handler om lage dataprogram



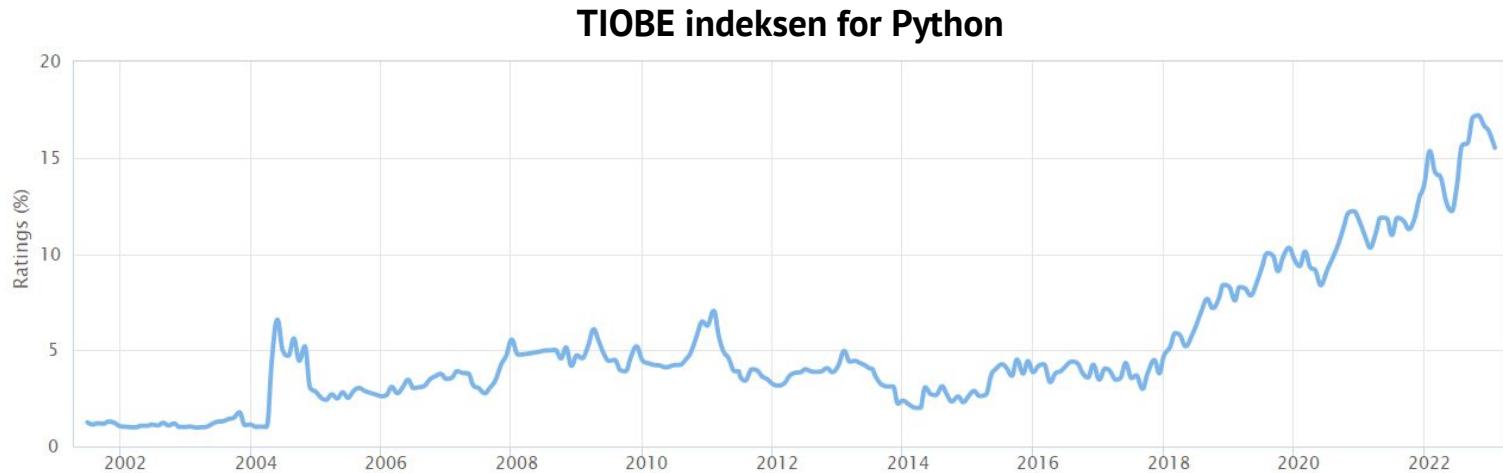
```
1 # -*- coding: utf-8 -*-
2
3 rente = 2.8
4 vekstfaktor = 1 + rente/100
5
6 innskudd = 1000
7 pengemengde = innskudd
8 år = 0
9
10 while pengemengde < 2*innskudd:
11     print(f"Etter {år:2} år: {pengemengde:.2f} kr")
12     år += 1
13     pengemengde *= vekstfaktor
14
15 print(f"Det tok {år} år å doble pengene")
16 |
```

# Programmering handler om å bryte opp kompliserte problemer i små biter, og løse dem steg for steg

Dette kalles gjerne for algoritmisk tankegang



# Dette kurset skal fokusere på programmering i Python



Det finnes mange forskjellige *programmeringsspråk*. Vi velger Python fordi det er lett å lære, anvendelig og plattformuavhengig



# Dette kurset går igjennom grunnleggende programmeringskonsepter i programmeringsspråket Python

## Dag 1

- Algoritmisk tenking
- Introduksjon til Python
- Variabler, regning, print

## Dag 2

- Betingelser
- Tekstbehandling
- Lister og løkker

## Dag 3

- Mer om lister og løkker
- Filbehandling

(Med forbehold om endringer underveis)

# **Vi er tre kursholdere med ansvar for en undervisningsdag hver**

**Dag 1**

---



**Marie Roald**

**Dag 2**



**Lena Myklebust**

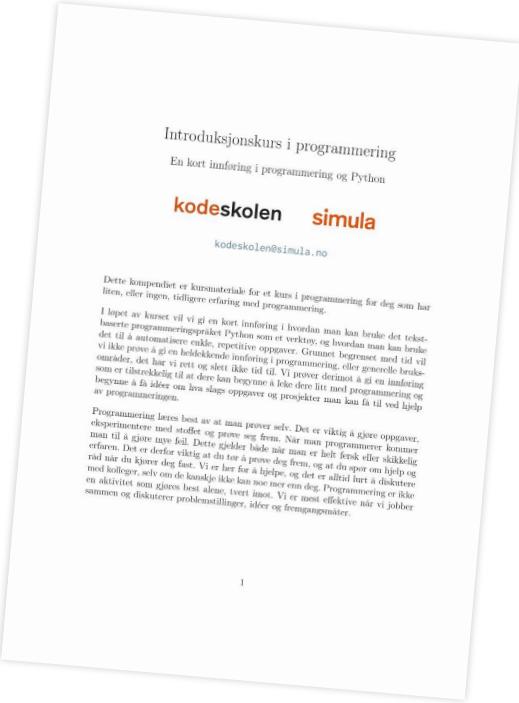
**Dag 3**

---

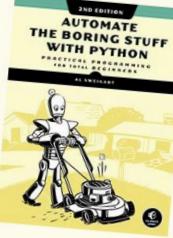


**Anders Høst**

# [github.com/kodeskolen/simula-kodekurs-v23](https://github.com/kodeskolen/simula-kodekurs-v23)



**AUTOMATE THE BORING STUFF WITH PYTHON**  
By Al Sweigart. Over 500,000 copies sold. [Free to read under a CC license.](#)



"The best part of programming is the triumph of seeing the machine do something useful. Automate the Boring Stuff with Python frames all of programming as these small triumphs; it makes the boring fun."  
- Hilary Mason, Data Scientist and Founder of [Fast Forward Labs](#)

"I'm having a lot of fun breaking things and then putting them back together, and just remembering the joy of turning a set of instructions into something useful and fun, like I did when I was a kid."  
- Wil Wheaton, [WilWheaton.net](#)

**PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS**  
If you've ever spent hours renaming files or updating hundreds of spreadsheet cells, you know how tedious tasks like these can be. But what if you could have your computer do them for you?

[kodeskolen / simula-kodekurs-v23](#) Public

Code Issues Pull requests Actions Projects Security Insights

main · 1 branch · 0 tags

MarieRaold Oppdater installasjonsguide

README.md Oppdater oversikt

Thonny Installasjonsguide.pdf Oppdater installasjonsguide

kodeskolen\_jukseapp.pdf Legg til materiale

Kompendium.pdf Legg til materiale

ressursguide.indd Legg til materiale

README.indd

Go to file Code

## Programmering i Python

### Hva er denne siden?

På denne siden deler vi kursmateriale for vårt krasjkurs i Python. Materialer legger vi ut etterhvert som kurset går, kom derfor gjerne tilbake ved en senere anledning for å finne nyere materialer!

### Kursets innhold

Kurset vil passe for deg som har lite eller ingen programmeringserfaring fra før. I kurset vil du få erfaring med noen av de mest grunnleggende programmeringskonsepene, og ved slutt av kurset skal du kunne lage dine egne små dataprogram.

### Kurforberedelser

På kurset skal vi kode sammen i et program som heter Thonny. Vi ber dere derfor installere dette programmet på forhånd og har laget en liten installasjonsguide for Windows og Mac som du finner her.

### Materiale

#### Kompendium og jukseapp

Kompendiet for kurset finner du her. Dette kompendiet er laget for kurset STKD6700 Introduction to Programming, og dekker dermed det vi går igjennom i dette kurset og litt mer. Kompendiet er en fin ressurs både for å slå opp temaer vi går igjennom i dette kurset og for å lære mer programmering på egenhånd.

# Hva er en algoritme?

**En algoritme er en nøyaktig serie med instrukser for å utføre en oppgave eller løse et problem**

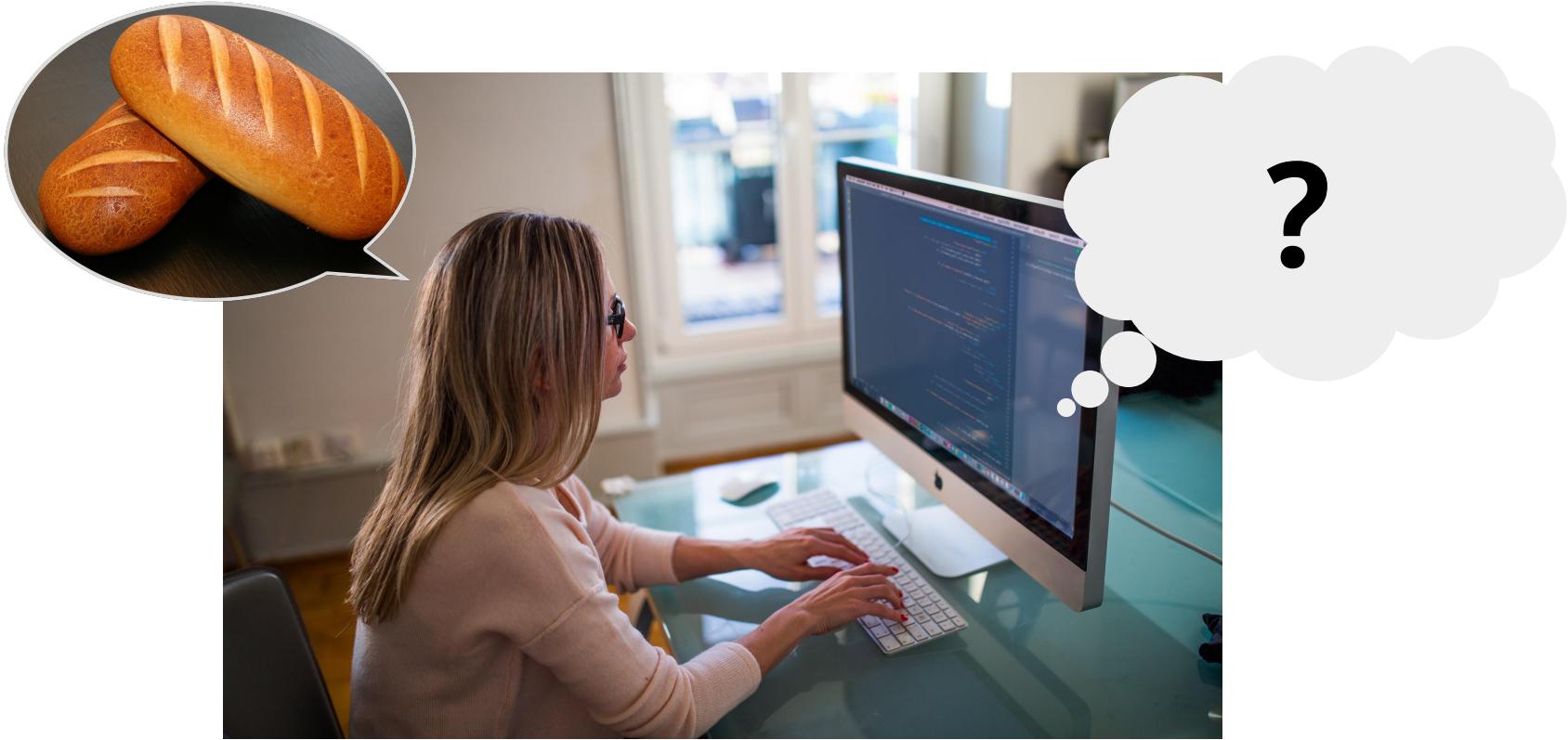
# En algoritme er en nøyaktig serie med instrukser for å løse et problem



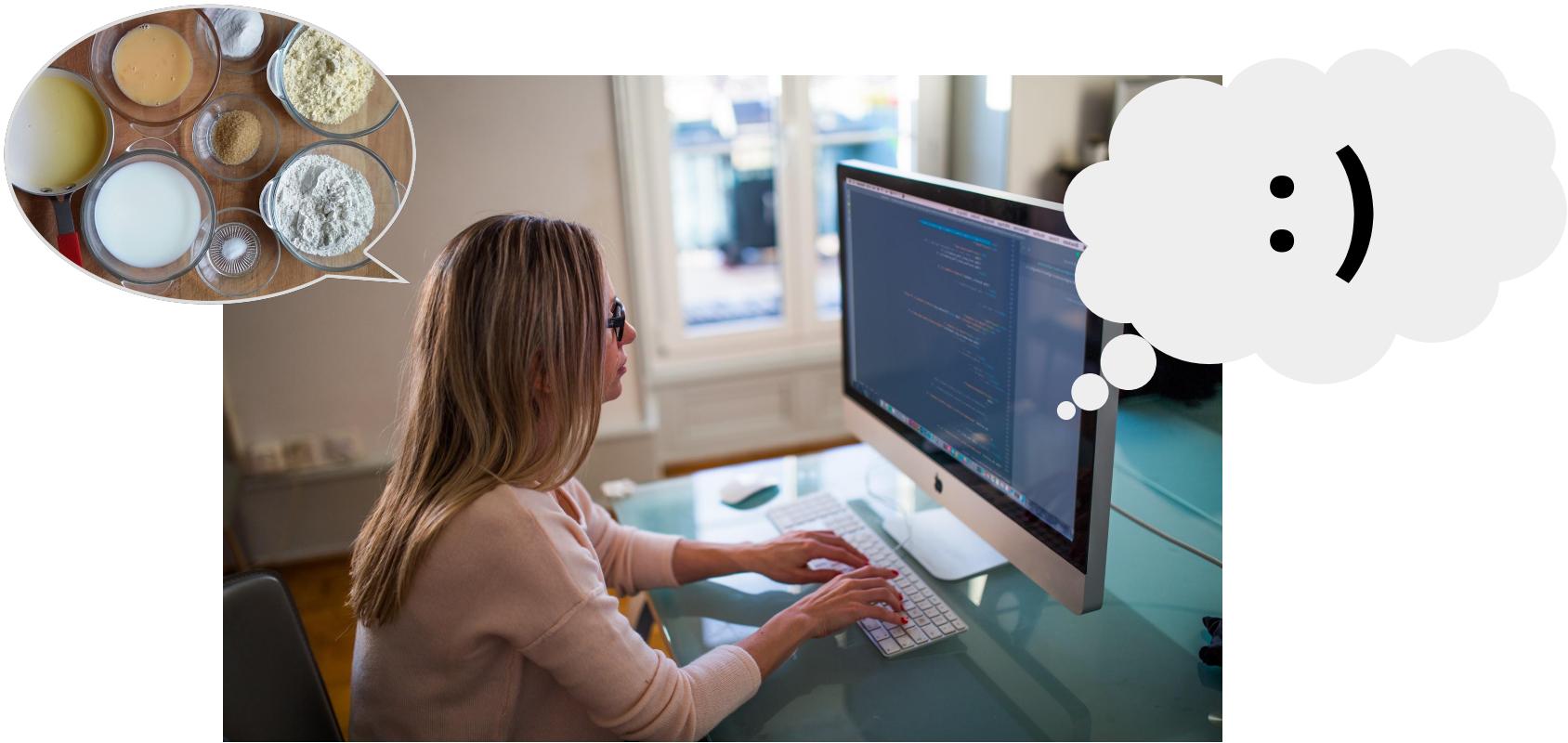
# En algoritme er en nøyaktig serie med instrukser for å løse et problem



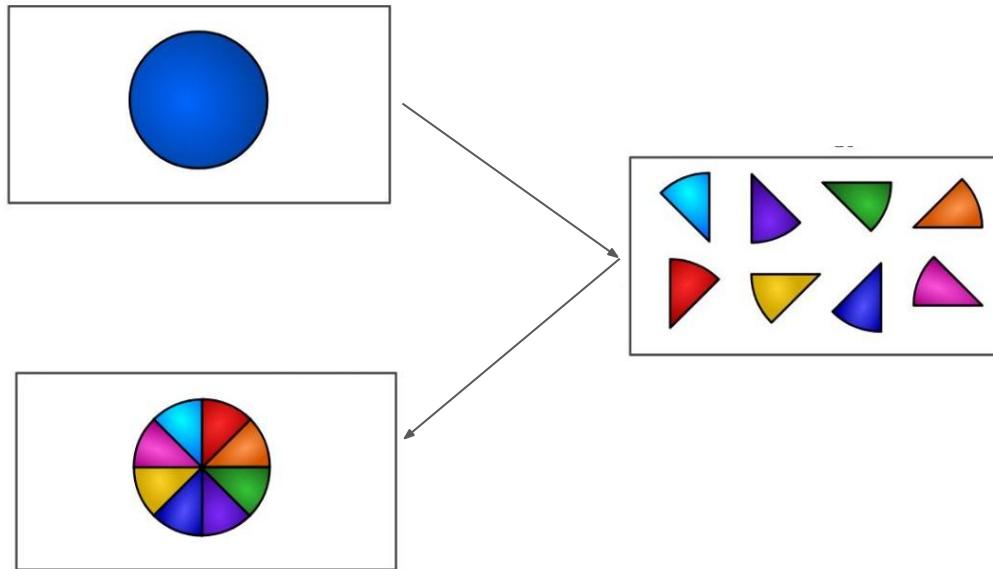
# Datamaskinen er dum, så den trenger nøyaktige instrukser



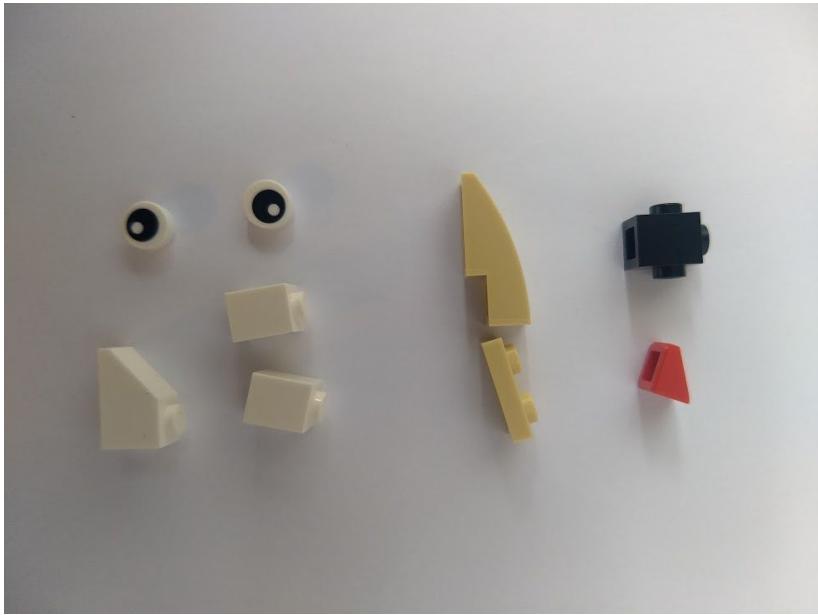
# Datamaskinen er dum, så den trenger nøyaktige instrukser



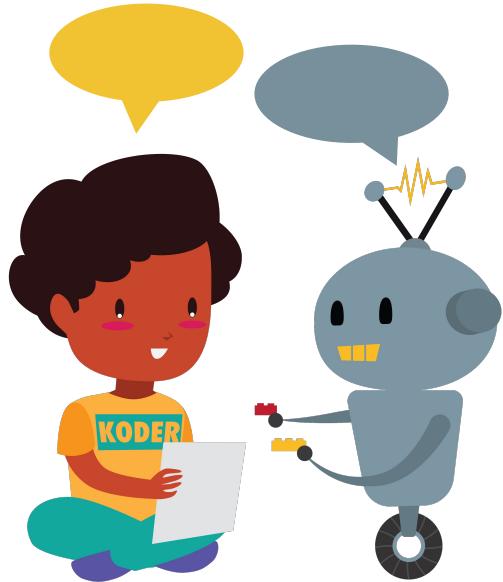
**Det at vi må være nøyaktige gjør at vi må bruke algoritmisk tenking**



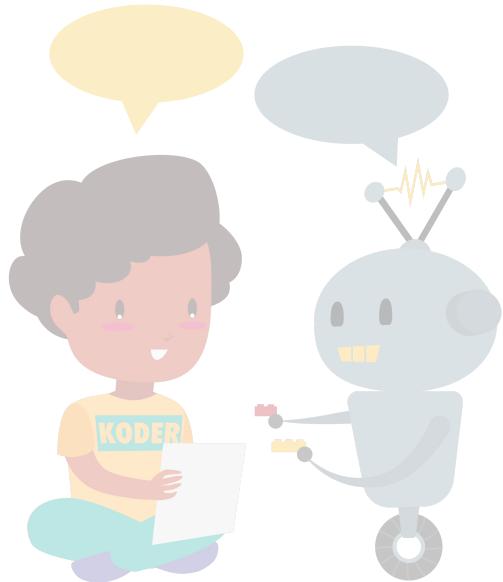
# Oppgave: Programmer en partner til å bygge lego



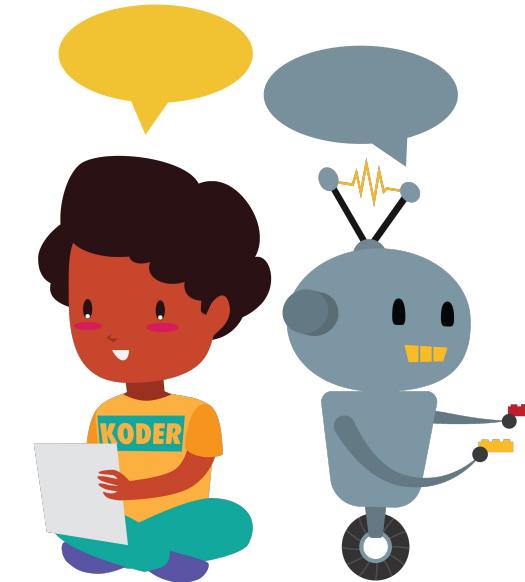
**Den første runden skal gjøres i samarbeid, her skal dere gjenskape figuren på bildet ved hjelp av brikkene dere får utdelt**



For andre runde skal dere sitte rygg mot rygg, en er koder og en er datamaskin. Koderen har oppskriften og datamaskinen har brikkene. Koderen skal instruere datamaskinen til å bygge figuren uten å selv se fasiten.

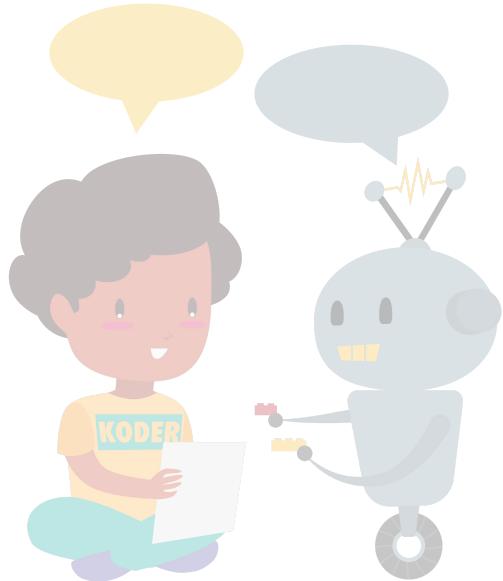


1

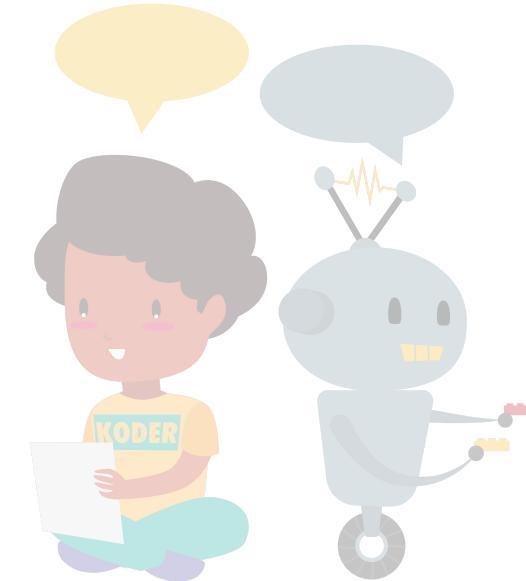


2

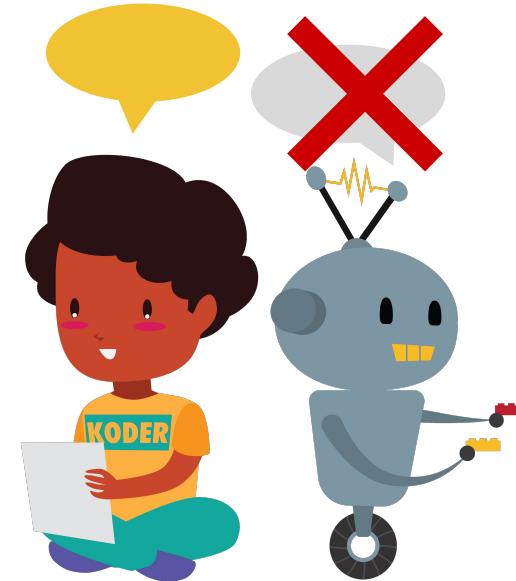
Runde tre skal gjøres som i forrige oppgave, men nå får ikke datamaskinen lov til å prate underveis. Da er det viktig å være nøyne!



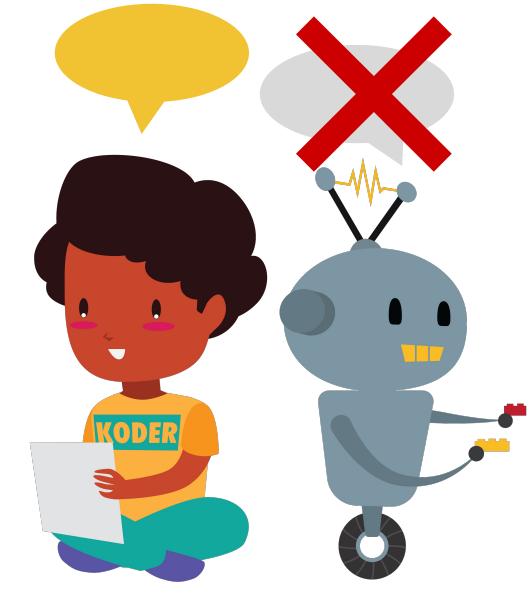
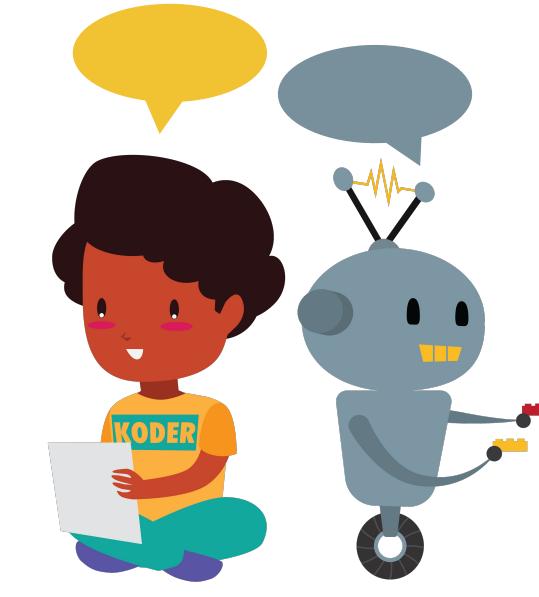
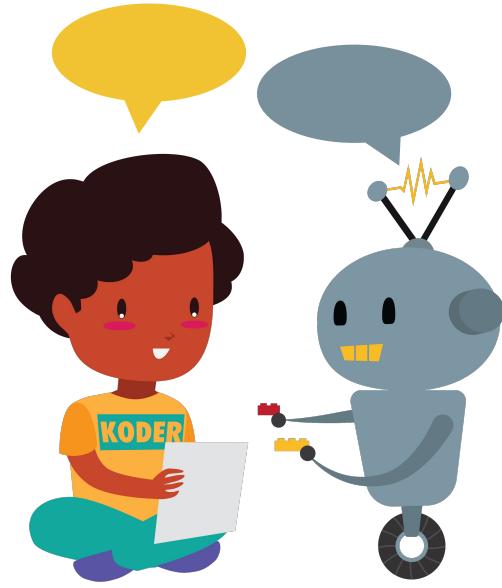
1



2



3



1

**Begge** kan se på fasit

**Begge** kan snakke

2

**Kun koder** kan se på fasit

**Begge** kan snakke

3

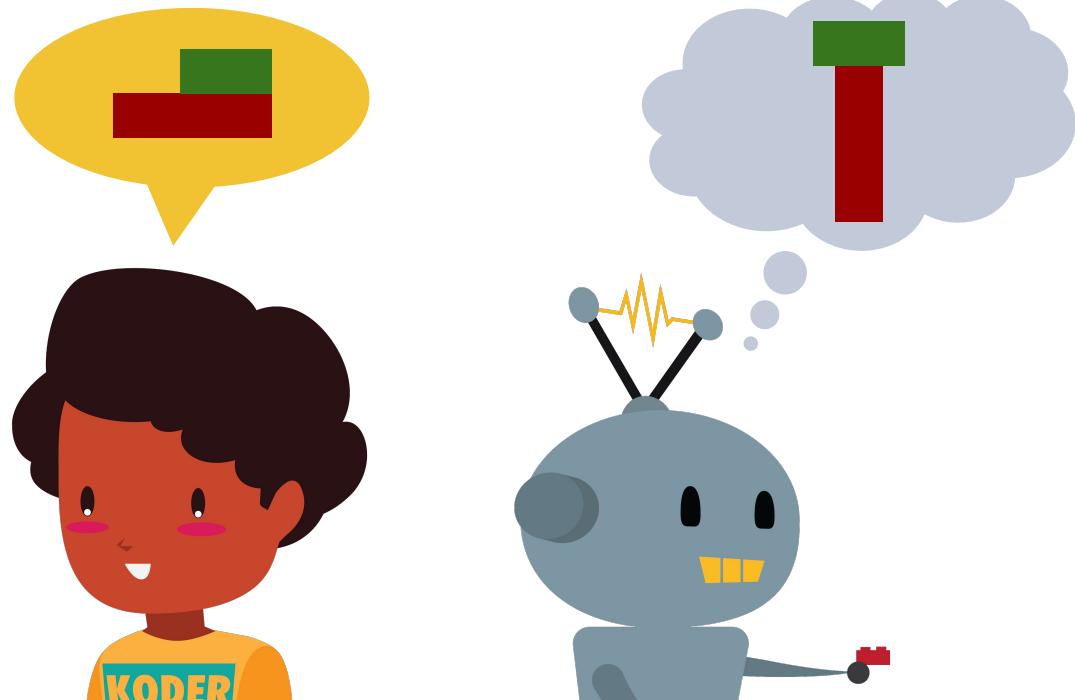
**Kun koder** kan se på fasit

**Kun koder** kan snakke

# Refleksjoner: Hva er vanskelig? Og hvordan kan det løses?



**Det er viktig å være helt nøyaktig, datamaskinen vet bare det du forteller den**

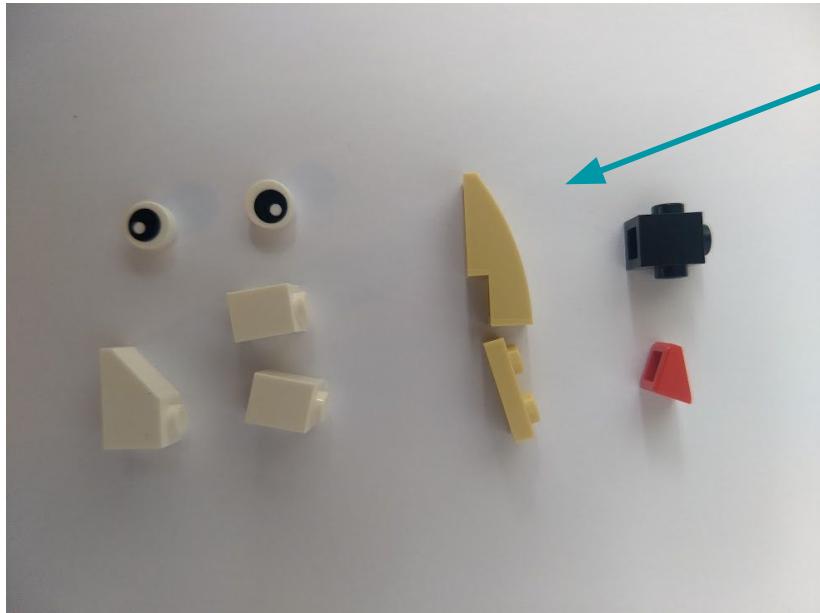


**Det er lurt å teste deler av programmet ditt underveis for å være sikker på at man er på rett spor**



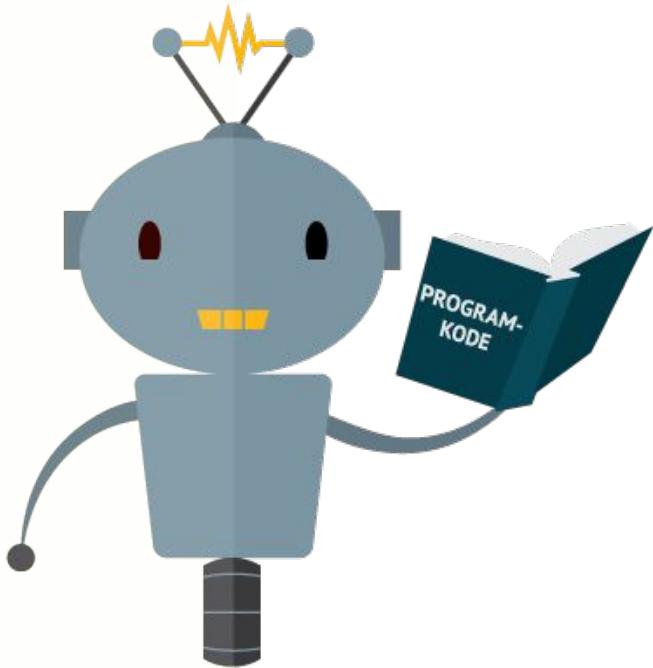
Photo: [Aroha Knits](#)

**For at en datamaskin skal skjønne hva du mener, må vi definere bestemte navn med bestemte verdier**



**“Nebbebrikke”**

**Et program er en algoritme skrevet på en måte slik at en datamaskin kan utføre den**



```
1 r = 3 #cm  
2 A = 3.14*r**2  
3 print(f'Arealet av sirkel med radius {r} cm er {A} cm^2')
```

