



# Algoritmisk tenking og programmering for alle

Sammen om oppdraget 2022

Svein & Harald



# Algoritmer

Nøyaktig beskrivelse av framgangsmåten for å utføre en oppgave.

## Analogt

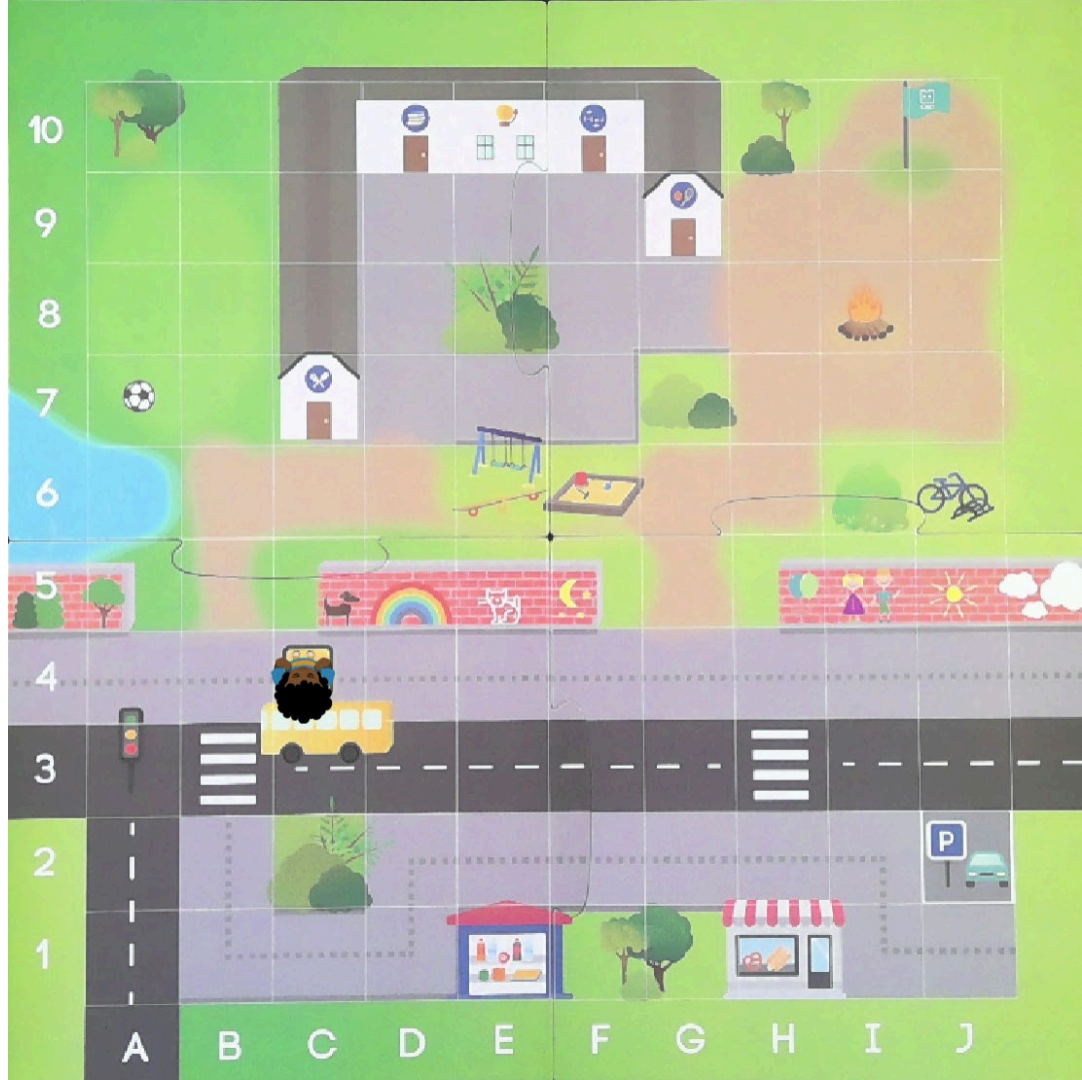
- Vi kan uttrykke algoritmer ved hjelp av forskjellige ikke-digitale representasjoner.
- Også ikke-fysiske representasjoner.

## Digitalt

- Utgangspunkt i det analoge.
- Sette sammen én eller flere algoritmer til et dataprogram

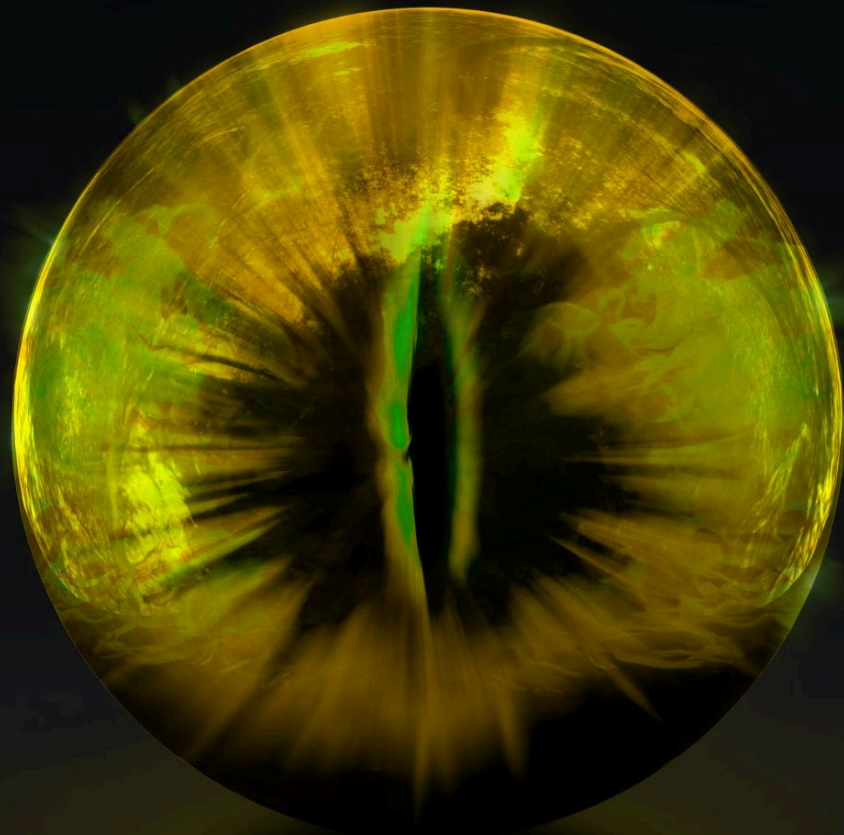


- Lag en beskrivelse som eleven kan bruke for å komme til inngangsdøren helt til høyre på den øverste bygningen.
- To mulige ruter.



# Forutse

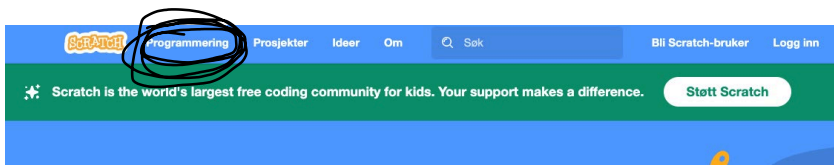
- Se på den utdelte kodesnutten.
- Hva kommer til å skje når du trykker på mellomrom?
- [Kontroller](#)



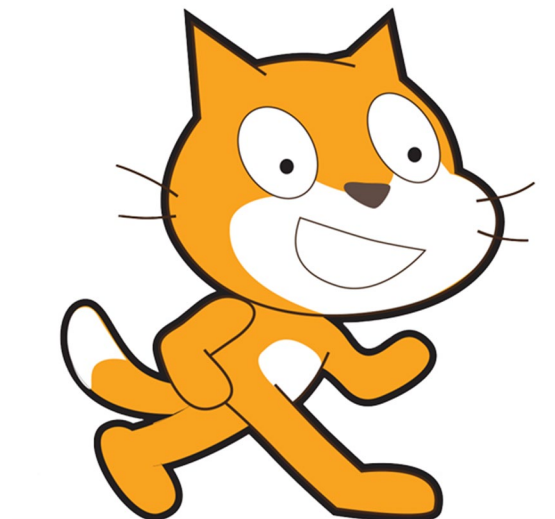
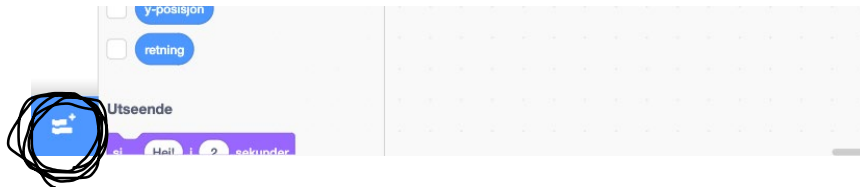


# Introduksjon til Scratch

- [www.scratch.mit.edu](http://www.scratch.mit.edu)



- Legg til penn-tillegget



# Lag programmet i Scratch

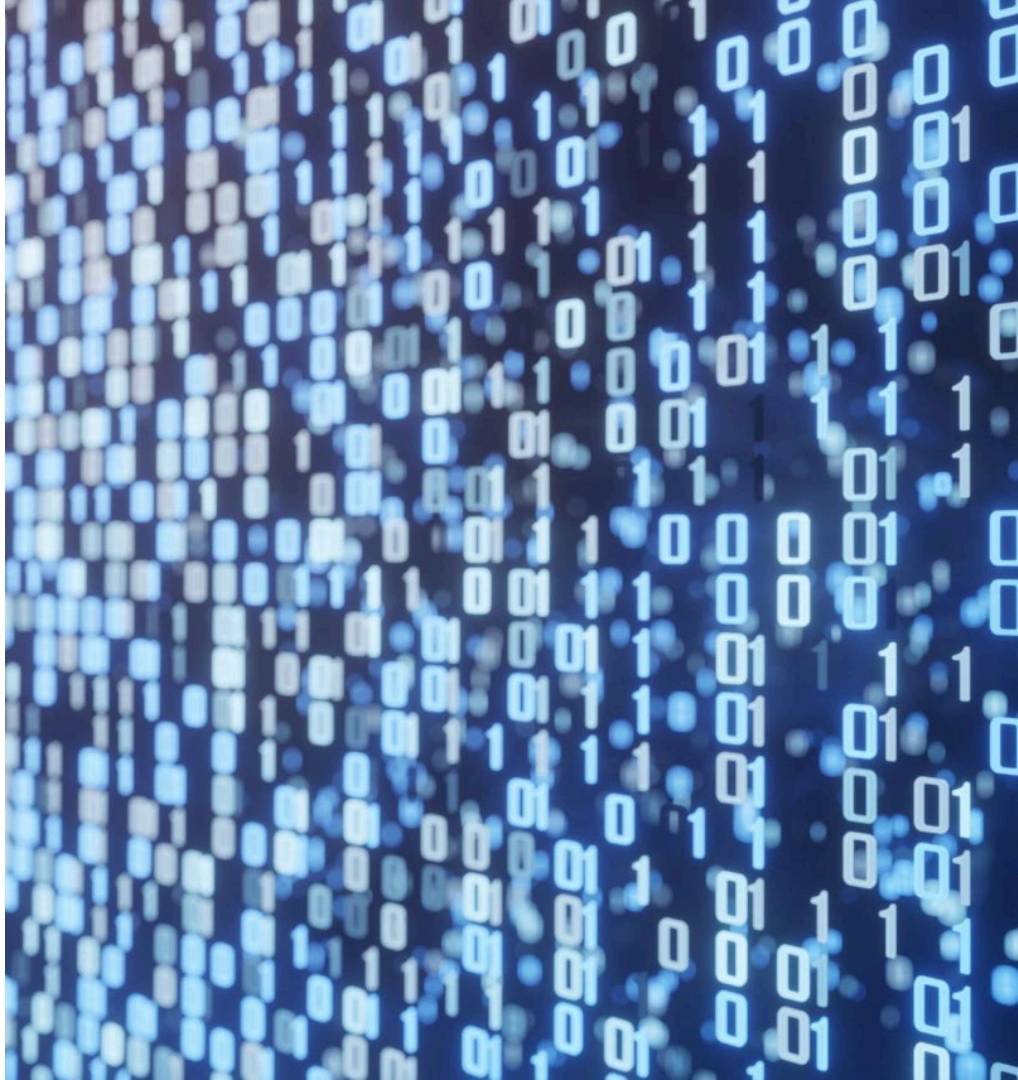
- Er det muligheter for optimalisering?
- Modifiser programmet slik at det tegner en regulær:
  - Trekant
  - Femkant
  - Sekskant
  - ...



# Programmering + Matematikk

Diskuter i grupper:

- Hvilke programmeringsbegrep har dere jobbet med?
- Hvor er matematikken?



# PRIMM-metode

## **Predict**

- Tolke og forutsi hva et program vil gjøre når det blir kjørt i gang. Elevene bør kjenne de fleste kommandoene.

## **Run**

- Kjør et ferdig program digitalt. Hvordan fungerer koden, og fungerer den som forutsett? Hvorfor/hvorfor ikke?

## **Investigate**

- Kodeforståelse. Kodelinjer, blokker, relasjon og makro.
- Spore, forklare, kommentere, feilsøk og retting.

## **Modify**

- Ta i bruk hele eller deler av ferdig program.
- Endre dette slik at det utfører annen lignende oppgave.

## **Make**

- Lag et program for å løse et spesifikt problem.
- Bruk kjente strukturer og begrep.
- Elevene må selv lage algoritmen som løser problemet.







**MATEMATIKKSENTERET**