

Samling 4:

Problemløsning med og uten datamaskin:
algoritmisk tenkning og programmering

Desentralisert kompetanseutvikling

Borkedalen skole

13.09.23

Universitetet i Agder

Program for dagen

12.40-15.30, pauser underveis når det passer

- Hvordan har det gått med medieproduksjonen?
 - eksempler på bruk i klassene
- Introduksjon til dagens tema
- Nivådeling og videre innføring i temaet
- Hva kan du finne i nettressursen?
- Arbeid med nettressurs og oppgaver
- Avsluttende undersøkelse

bit.ly/borkedalnett

Ressursside med hjelpevideoer og oppgaver for samling 4. Her finner du også pdf for dagens samling.

Samling 4 DEKOMP

- Algoritmisk tenkning som praktisk problemløsningsmetode
- Koding vs. programmering
- Eksempler på algoritmisk tenkning og programmering
- Visuell koding med blokker i Code.org, Scratch og MakeCode (Micro:bit og Arcade).



Samling 4

Problemløsning med og uten datamaskin – algoritmisk tenkning og programmering

Hjelpevideoer og oppgaver

Temaet som blir gjennomgått i Samling 4:

Klikk deg inn nedenfor for å få tilgang til PDF-presentasjon for Samling 4 som gæes gjennom for fysiske samlinger.

[PDF-presentasjon for Samlingen 4](#)

Generelle tips til bruk av ressursiden:

- Ønsker du å skrive ut noen av oppgavene kan du på PC-en bruke tastene **Ctrl + P** for å printe ut oppgaveteksten.
- For å få oppgavetekst i nettleseren større eller mindre, bruk tastene **Ctrl+** eller **Ctrl-**
- For å få video vist større/i fullskjerm klikker du på fullskjerm-knappen ✓ i

videoavspilleren:



Regjeringen vil:

- Gi skolen et digitalt løft gjennom (...) vekt på utvikling av teknologiforståelse, digital dømmekraft og skaperkraft.
- Sikre at elever får kunnskap om og forståelse for teknologi, algoritmisk tenkning og koding i flere fag, allerede tidlig i skoleløpet.
- Gi innovasjon og skaperkraft en god plass i fremtidens skole.

nrk.no

Facebook Foto, Video ...aakon Sundbø Canvas LinkedIn Learning Photography Videography Marketing/Business Audio tech Layout/Design UI/UX/Usability 3D Wordpress Social Gamification EduTech Schooling Mobile iOS Web-tech

Lærere trenger hjelp til å knekke koden på koding – NRK Innlandet – Lokale nyheter, TV og radio fagfornyelsen 2020 - Google-søk

NRK TV NRK RADIO NRK P3 NRK UR


NRK Nyheter Sport Kultur Humor Distrikt Mer

Haakon Søk

Innlandet Tips oss Langlesing Radio TV

Skal lære elevene koding, men forstår det ikke selv

Lærere over hele landet ber om hjelp til å til å lære barn om programmering. – Mange har ikke kunnskapen som trengs, sier fagekspert.



Haakon Nesse Moreau
Journalist

Publisert 30. des. 2021 kl. 13:49
Oppdatert 31. des. 2021 kl. 08:57

ENGASJERTE: Fremtidens næringsliv vil kreve enorm digital kompetanse. Det er en av grunnene til å engasjere barn i programmering. Bildet er fra Learnlink sitt kurs der barn lærer å kode samtidig som de gamer.
FOTO: LEARNLINK

I 2020 ble det innført ny læreplan i norske skoler. Plutselig skulle

<https://www.nrk.no/innlandet/laerere-trenger-hje-lp-til-a-knekke-koden-pa-koding-1.15781343>

Ny teknologi og roboter setter arbeidsplasser i fare

På grunn av teknologisk utvikling står 40 prosent av norske jobber i fare for automatisering, i følge OECD.

Sist oppdatert
11. mar. 2020



Iris Christine Schjerve
Journalist

Les mer om:
Valg av yrke

Liker 4 Del Tweet

<https://studenttorget.no/index.php?show=6512&expand=3796,3931,6512&artikkelid=20171>

SUPER:BIT

Sender ut 60.000 micro:bits til norske skoler: – Vi har stor tro på denne unike muligheten



Med super:bit skal norske skolelever lære seg å programmere og få bedre teknologiforståelse. (Foto: Marius Jørgensen)

MARTIN BRAATHEN RØISE DIGITALISERING OG OFFENTLIG IT 14. MAI 2019 - 20:00

NRK, Vitensenteret og Lær Kidsa Koding går sammen om et felles prosjekt for å lære barneskoleelever programmering og gi dem teknologiforståelse. I løpet av de neste tre årene skal klassesett med den lille datamaskinen micro:bit sendes ut til 2 400 norske barneskoler.

<https://www.digi.no/artikler/na-skal-60-000-norske-skoleelever-laere-seg-a-kode-med-denne-dingsen-vi-har-stor-tro-pa-denne-unike-muligheten/4652>



I august blir programmering en obligatorisk del av matematikkfaget i grunnskolen. Forskere sier at det nå er viktig at lærerne får den nødvendige kompetansen for å kunne gi god undervisning i programmering. (Foto: Bård Halvorsen, HiØ)

– Programmering vil bli en utfordring for lærere

Til høsten blir programmering en del av matematikkfaget i grunnskolen. Nå advarer forskere mot å tro at økt fokus på programmering alene er nok til å lykkes.

Ann-Kristin Johansen
RÅDGIVER

Høgskolen i Østfold

Lørdag 11. juli 2020 - 04:30

<https://forskning.no/barn-og-ungdom-hogskolen-i-ostfold-matematikk/programmering-vil-bli-en-utfordring-for-laerere/1711838>

Tips & Tricks & Utdannelse

Norge trenger utviklere: Her er fire morsomme måter å lære seg koding

Skrevet av Eirik Solheim 29. januar 2020

Sliter med å finne nok utviklere. Det kan tvinge «årets startup» ut av Norge

– Vi har søkt etter utviklere til mange ledige stillinger

– Mer attraktive enn gull

46 åringer fra Stabakk mener

Spør et mangfold av jobbmuligheter

Digitaliseringen blir en del av

– Hvis programmering og

Tøff konkurranse om utviklere

i20
sound trigger

o14
bright led

Kåret til mest attraktive yrke

De får seg lett jobb, får god lønn og kan jobbe hjemmefra. Det er hovedårsakene til at

Vi søker etter dyktige utviklere/ programmerere

Disse utviklerne tjener best i Norge

UTVIKLERKOMPETANSE
Alle vil ha utviklere med disse ferdighetene

– Vi hadde hatt jobb til dem på dagen.

<https://nrkbeta.no/2020/01/29/norge-trenger-utviklere-her-er-fire-morsomme-mater-a-laere-seg-koding/>

Kultur | Skole og utdanning

Nesten ingen av lærerne kan faget de nå skal lære bort



Hver firkant på nettbrettet blir til en kommando i robotens oppførsel.

<https://www.aftenposten.no/kultur/i/OpLb33/nesten-ingen-av-laererne-kan-faget-de-naa-skal-laere-bort>



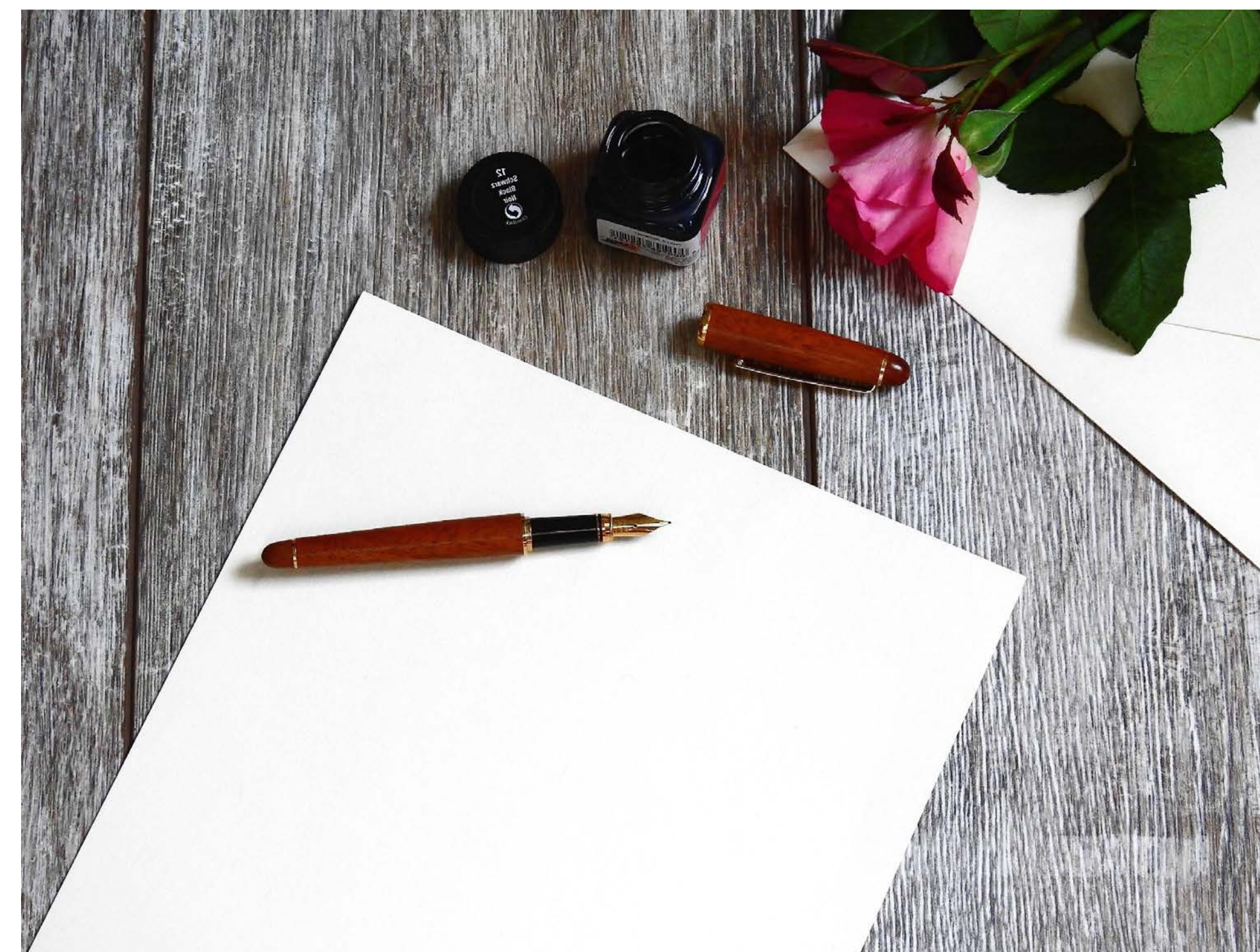
Exact Instructions Challenge - THIS is why my kids hate me. | Josh Darnit:
https://www.youtube.com/watch?v=cDA3_5982h8



Presise verbale instruksjoner

Nå skal vi sette algoritmisk tenkning «ut i livet».

I denne aktiviteten skal du være datamaskinen. En datamaskin må ha presise instruksjoner. Rekkefølgen for instruksjonene er naturligvis avgjørende. Hva tror du skjer når instruksjonene ikke er presise nok?



Lytt og tegn

Samlingens tema og aktiviteter vil være nivådelt



Nivå 1

For deg som har lite kjennskap til temaet fra før og som ønsker en grunnleggende innføring om algoritmisk tenkning og problemløsning med og uten datamaskin



Nivå 2

For deg som kjenner litt til temaet fra før.
Her står du mere fritt til å velge verktøy og oppgaver.



Nivå 3

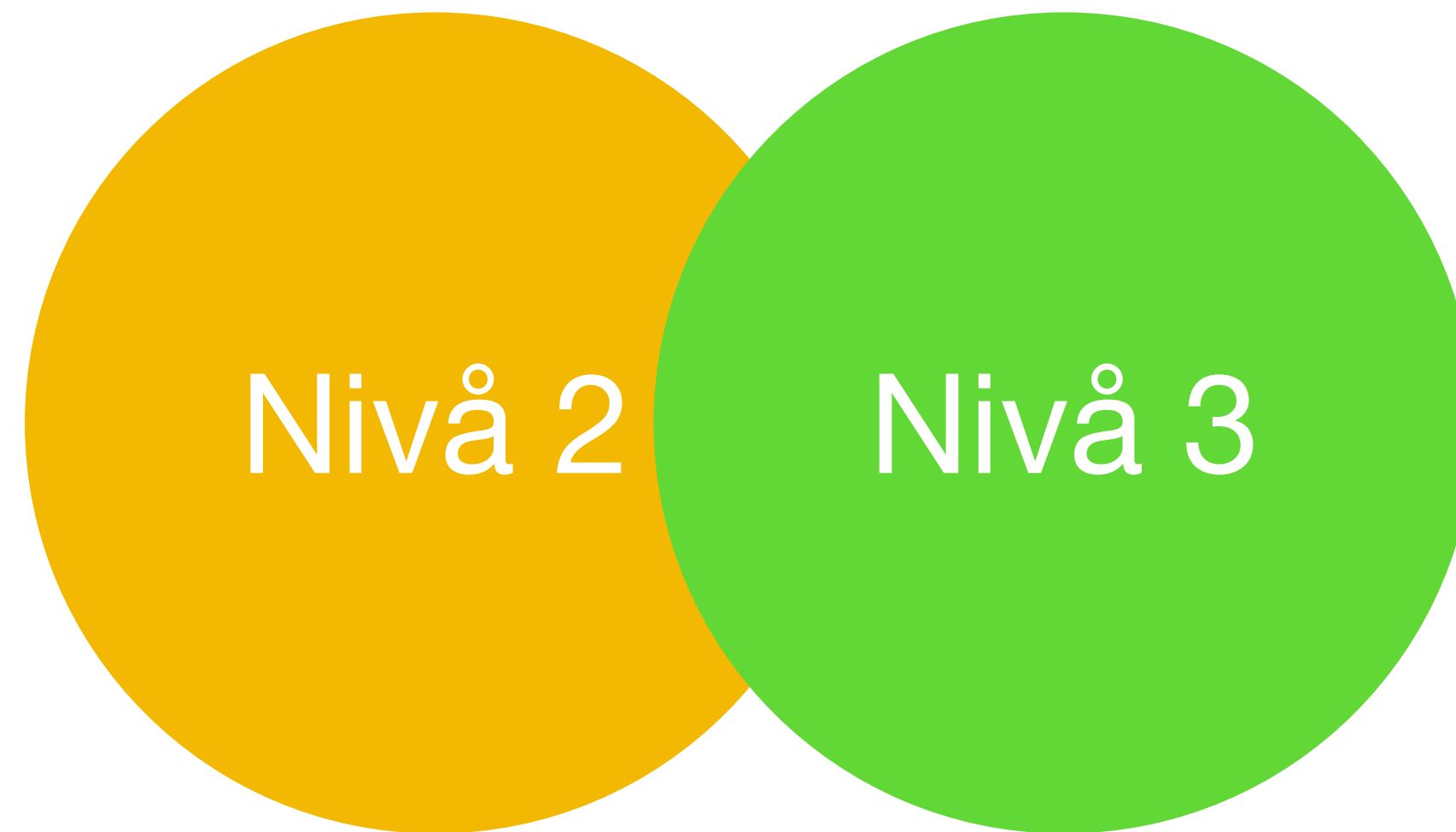
For deg som har mye erfaring med temaet.
Her står du fritt til å jobbe selvstendig med alle ressursene

Vi deler oss i to rom

**Rom 1:
Gry**



**Rom 2:
Haakon**





Nivå 1

Eksempler fra planverket hvor digitalisering blir konkret satt opp som kompetansemål

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kompetansemål etter 7. trinn

- bruke teknologi og digitale verktøy til å skape, øve inn og bearbeide musikk

Kompetansemål etter 4. trinn

- eksperimentere med rytmer, melodier og andre grunnelementer, sette sammen mønstre til komposisjoner, også ved bruk av digitale verktøy, og beskrive arbeidsprosesser og resultater

Kompetansemål etter 2. trinn

- leke med musikkens grunnelementer gjennom lyd og stemme, lage mønstre og sette sammen mønstrene til enkle improvisasjoner og komposisjoner, også med digitale verktøy

Kunst og håndverk

Kompetansemål etter 10. trinn

- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Kompetansemål etter 7. trinn

- bruke digitale verktøy til å planlegge og presentere prosesser og produkter
- analysere visuelle virkemidler i ulike medier og belyse et aktuelt tema gjennom foto eller infografikk
- bruke programmering til å skape interaktivitet og visuelle uttrykk

Kompetansemål etter 4. trinn

- bruke enkle komposisjonsprinsipper i fotografi og digitale verktøy

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener
- utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger

Kompetansemål etter 7. trinn

- designe og lage et produkt basert på brukerbehov
- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Kompetansemål etter 4. trinn

- utforske teknologiske systemer som er satt sammen av ulike deler, og beskrive hvordan delene fungerer og virker sammen
- designe og lage et produkt basert på en kravspesifikasjon

Kompetansemål etter 2. trinn

- presentere funnene sine og beskrive hvordan eleven har kommet fram til dem
- presentere egne ideer til teknologiske oppfinnelser

Norsk

Kompetansemål etter 10. trinn

- utforske og vurdere hvordan digitale medier påvirker og endrer språk og kommunikasjon
- bruke kilder på en kritisk måte, markere sitater og vise til kilder på en etterrettelig måte i egne tekster

Kompetansemål etter 7. trinn

- orientere seg i faglige kilder på bibliotek og digitalt, vurdere hvor pålitelige kildene er, og vise til kilder i egne tekster
- reflektere etisk over hvordan eleven framstiller seg selv og andre i digitale medier
- presentere faglige emner muntlig med og uten digitale ressurser
- utforske og beskrive samspillet mellom skrift, bilder og andre uttrykksformer og lage egne sammensatte tekster

Kompetansemål etter 4. trinn

- holde muntlige presentasjoner med og uten digitale ressurser

Kompetansemål etter 2. trinn

- lese med sammenheng og forståelse på papir og digitalt og bruke enkle strategier for leseforståelse
- lage tekster som kombinerer skrift med bilder

Samfunnsfag

Kompetansemål etter 10. trinn

- bruke samfunnsfaglege metodar og digitale ressursar i eigne undersøkingar, presentere funn ved bruk av digitale verktøy og drøfte kor gyldige og relevante funna er
- vurdere på kva måtar ulike kjelder gir informasjon om eit samfunnsfagleg tema, og reflektere over korleis algoritmar, einsretta kjelder eller mangel på kjelder kan prege forståinga vår
- utforske korleis teknologi har vore og framleis er ein endringsfaktor, og drøfte innverknaden teknologien har hatt og har på enkeltmenneske, samfunn og natur
- utforske og reflektere over eigne digitale spor og høvet til å få sletta spora og å verne om retten ein sjølv og andre har til privatliv, personvern og opphavsrett
- utforske ulike plattformer for digital samhandling og reflektere over korleis digital deltaking og samhandling påverkar forma på og innhaldet i samfunnsdebatten

Kompetansemål etter 7. trinn

- gjennomføre ei samfunnsfagleg undersøking og presentere resultatata ved hjelp av eigna digitale verktøy
- presentere ei aktuell nyheits sak og reflektere over forskjellar mellom fakta, meiningar og kommersiell budskap i mediebiletet
- utforske korleis menneske i fortida livnærte seg, og samtale om korleis sentrale endringar i livsgrunnlag og teknologi har påverka og påverkar demografi, levekår og busetjingsmønster
- reflektere over korleis ein sjølv og andre deltek i digital samhandling, og drøfte kva det vil seie å bruke dømmekraft sett i lys av reglar, normer og grenser

Kompetansemål etter 4. trinn

- samtale om reglar og normer for personvern, deling og beskyttelse av informasjon og om kva det vil seie å bruke dømmekraft i digital samhandling
- utforske og presentere samfunnsfaglege spørsmål, søkje etter informasjon i ulike kjelder og vurdere kor nyttig informasjonen er til å belyse spørsmåla

Kompetansemål etter 2. trinn

- samtale om moglegheiter og utfordringar ved digital samhandling

Matte

Kompetansemål etter 10. trinn

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultatata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering
- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Kompetansemål etter 8. trinn

- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett
- lage og vurdere budsjett og rekneskap ved å bruke rekneark med cellereferansar og formlar

Kompetansemål etter 6. trinn

Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Kompetansemål etter 4. trinn

- lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer

Kompetansemål etter 3. trinn

- lage og følge reglar og trinnvise instruksjonar i leik og spel knytte til koordinatsystemet

Kompetansemål etter 2. trinn

- lage og følge reglar og trinnvise instruksjonar i leik og spel

Eksempler fra planverket hvor programmering blir konkret satt opp som kompetansemål

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kunst og håndverk

Kompetansemål etter 7. trinn

- bruke programmering til å skape interaktivitet og visuelle uttrykk

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener

Kompetansemål etter 7. trinn

- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Matte

Kompetansemål etter 10. trinn

- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering

Kompetansemål etter 8. trinn

- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett

Kompetansemål etter 6. trinn

- bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster

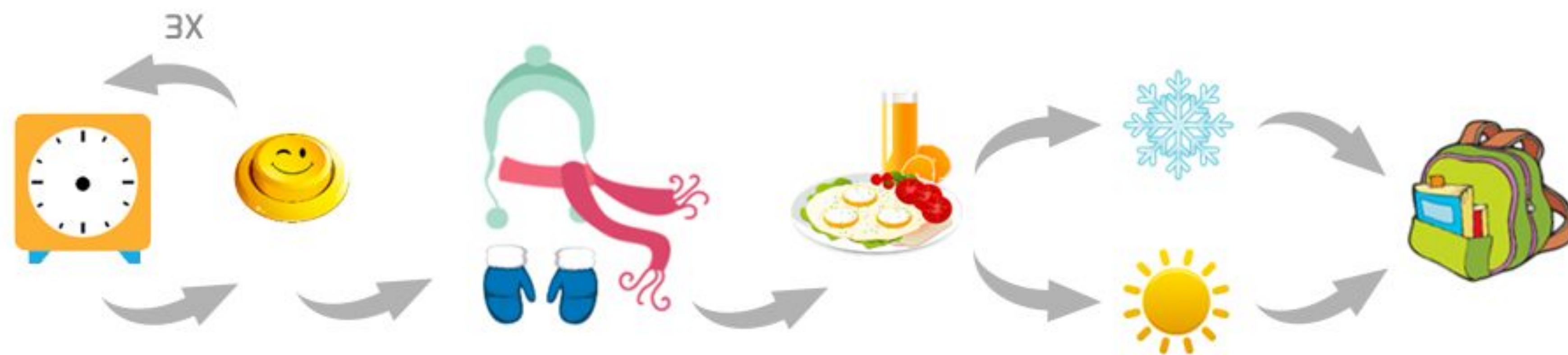
Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Algoritmisk tenkning

En algoritme kan forklares som en meget detaljert trinn-for-trinn oppskrift som må følges i nøyaktig riktig rekkefølge.

Et eksempel på en algoritme



Den algoritmiske tenkeren

Nøkkelbegrep

1 Logikk
Analysere og forutse

2 Algoritmer
Regler og steg-for-steg

3 Dekomposisjon
Bryte ned i mindre deler

4 Mønstre
Finne og bruke likheter

5 Abstraksjon
Fjerne unødvendige detaljer

6 Evaluering
Gjøre vurderinger

Arbeidsmåter

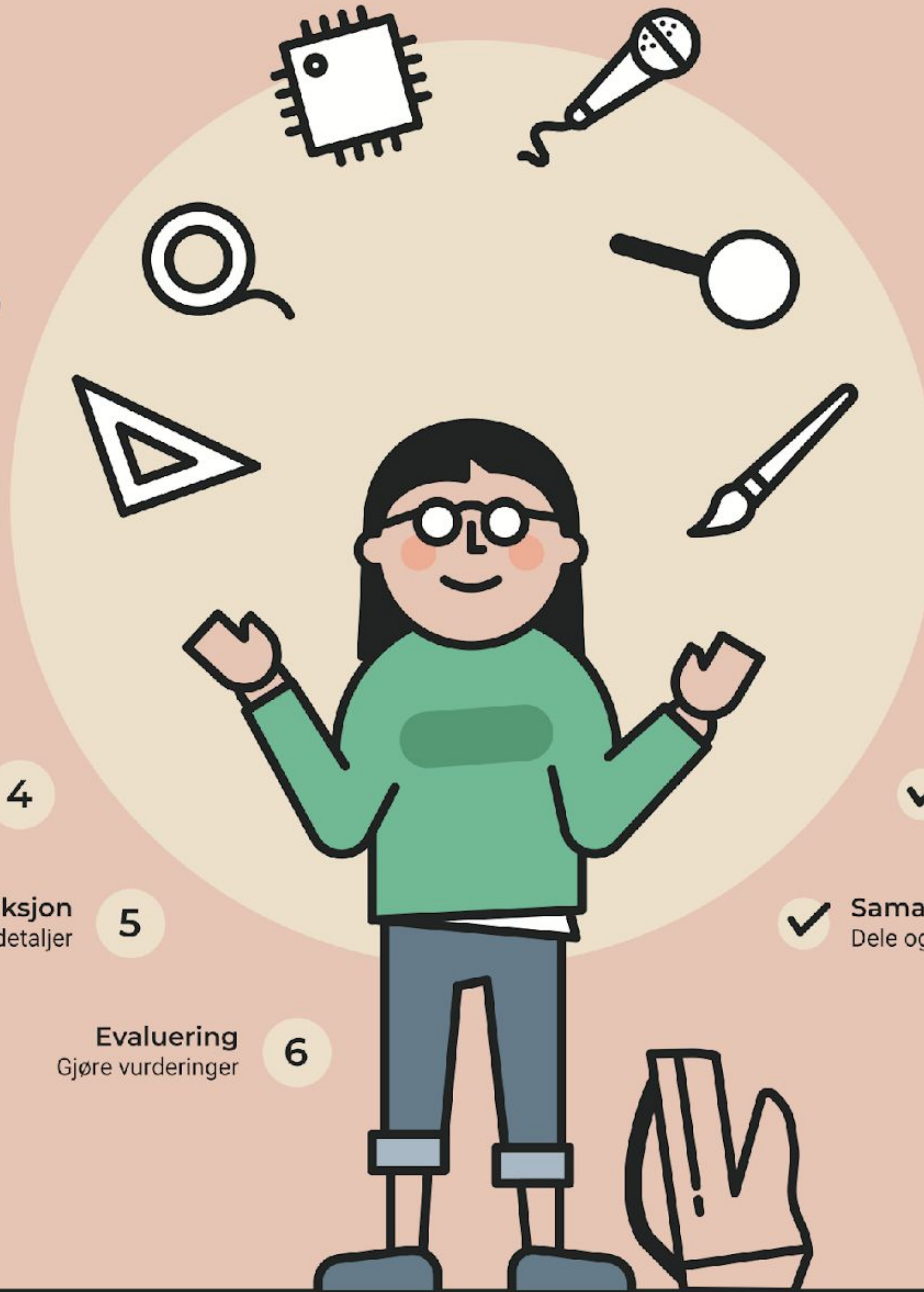
✓ **Fikle**
Utforske og eksperimentere

✓ **Skape**
Designe og lage

✓ **Feilsøke**
Oppdage og rette feil

✓ **Holde ut**
Fortsette og prøve igjen

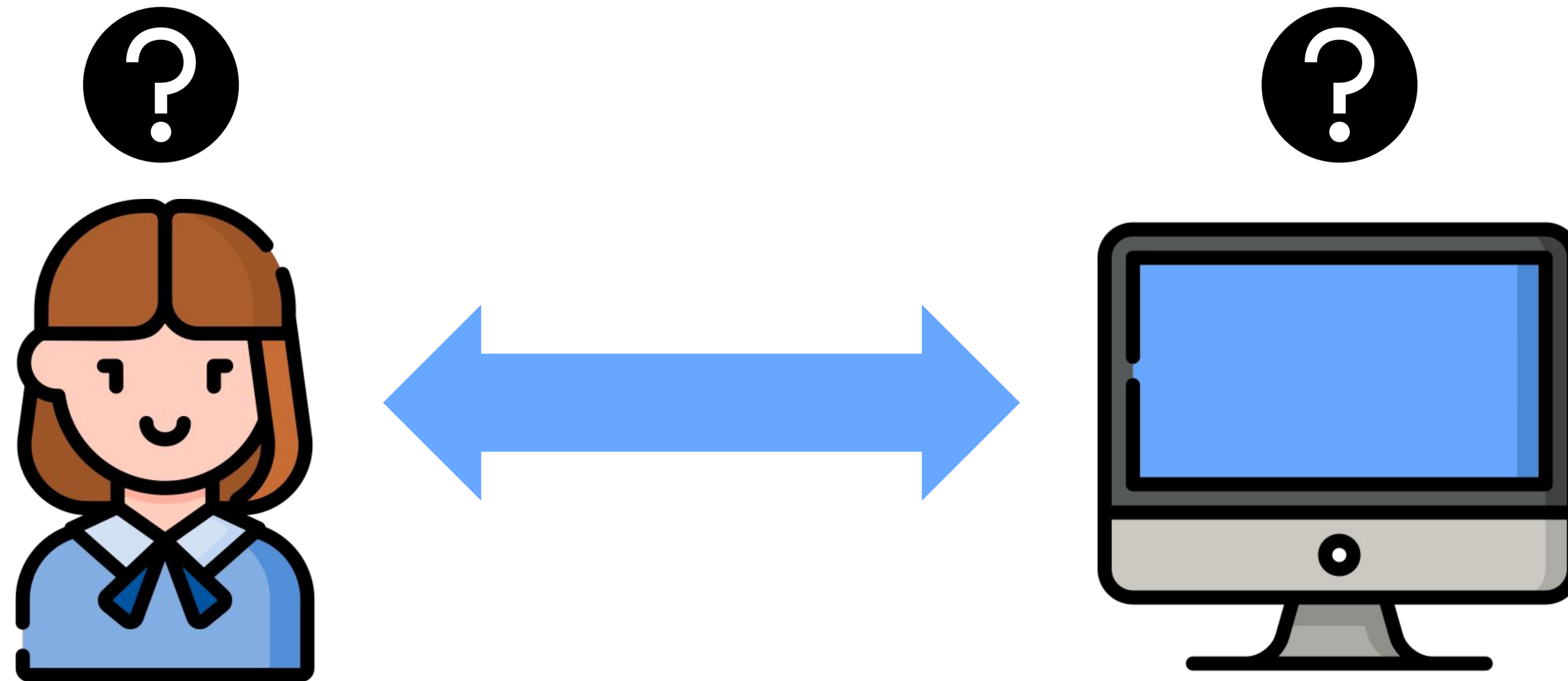
✓ **Samarbeide**
Dele og jobbe sammen



Algoritmisk tenkning

- Problemløsningsmetode
- Systematisk
- Vurdering: teknologi vs. menneske
- *Algoritmisk tenkning er den norske oversettelsen av det engelske ordet computational thinking.*

Hva er programmering?



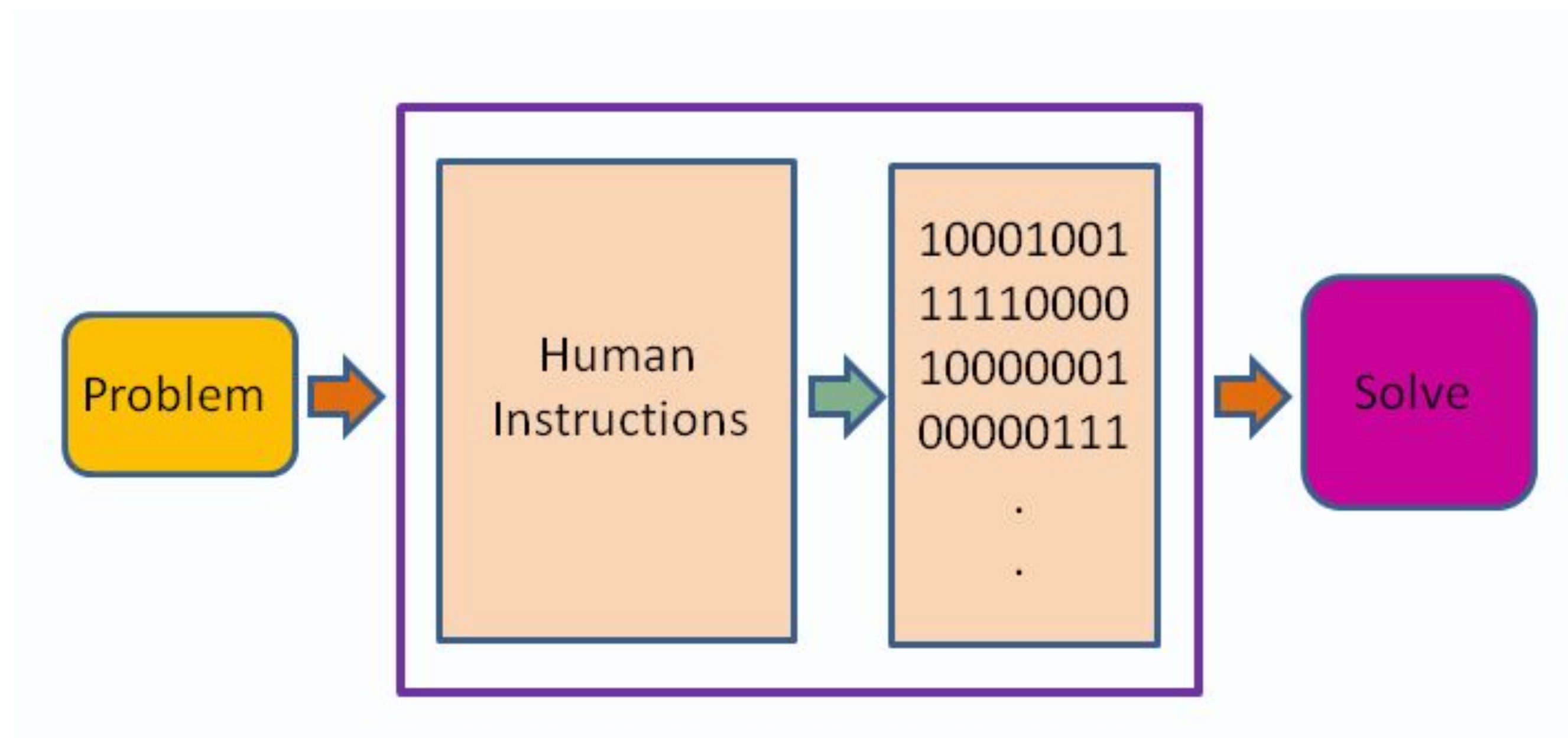


Med programmering gjelder det å kunne formulere et problem og finne løsningen ved hjelp av et dataprogram.

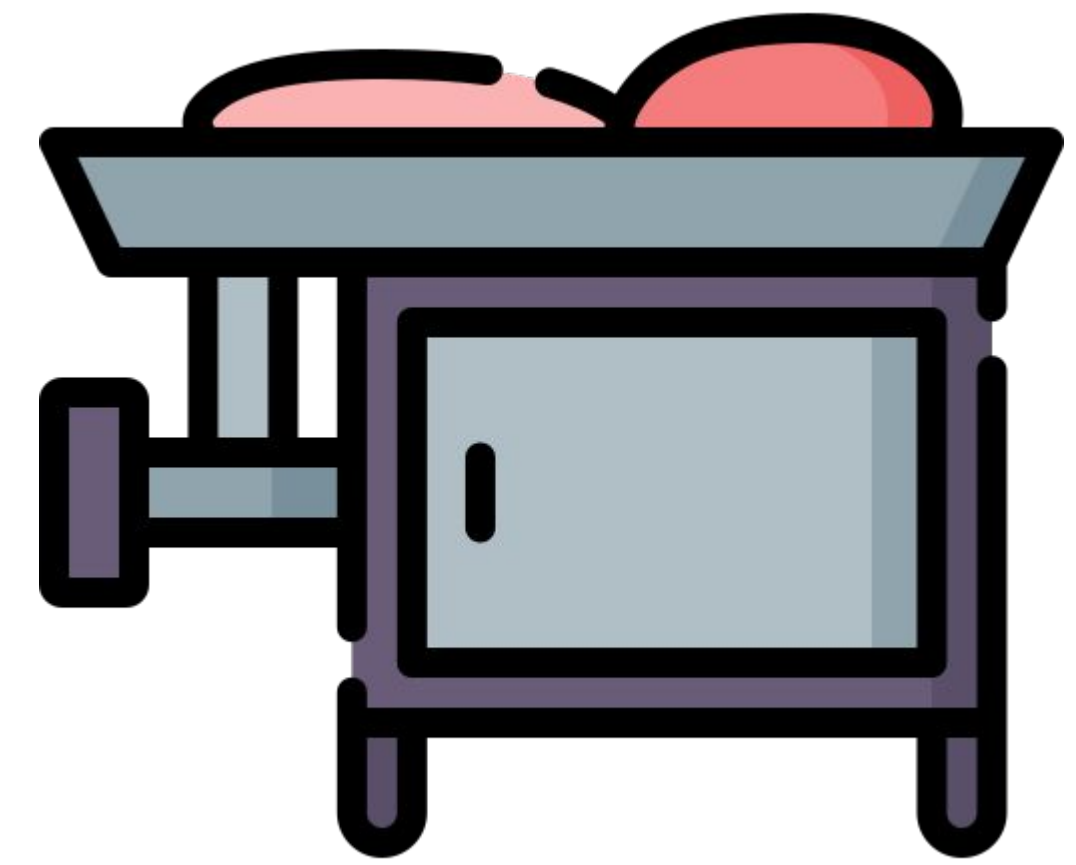
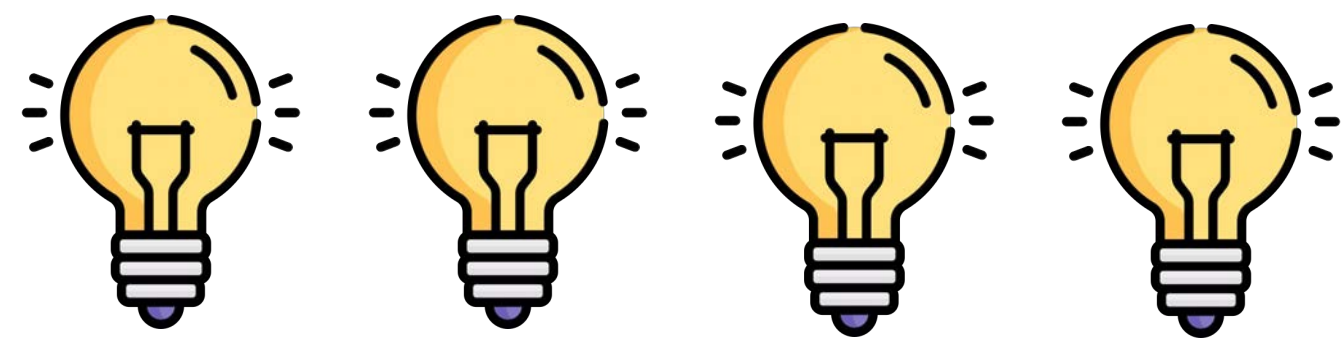
Hentet fra: <https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Selve prosessen med å skrive programmet kalles i IT-sjargong ofte for *koding*.

Hentet fra: <https://www.hiof.no/lusp/slik/forskning/grupper/begynneropplering/aktuelt/aktuelt-2019/lererne-er-positive-til-programmering---men-mangle.html>



Er koding og programmering det samme?



0 0 1 0 1 1

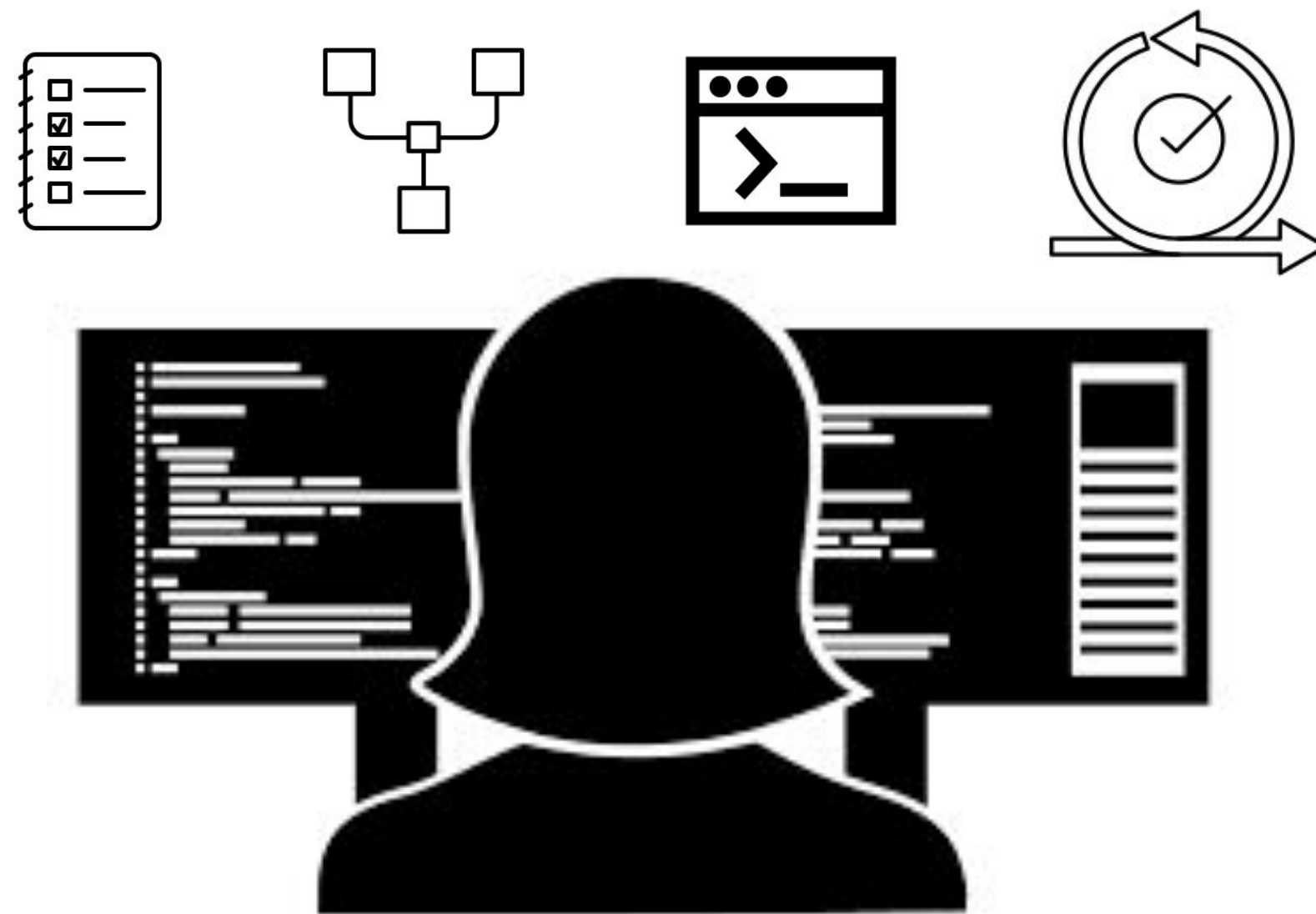
Å lære å kode handler om å lære seg
det språket som skal til for å
kommunisere med datamaskinen



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Programmering er derimot noe langt mer. Det involverer blant annet også tankeprosessene som går forut for kodingen.



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Kodespråk

Tekstbaserte

```
1 /*
2  * This line basically imports the "stdio" header file, part of
3  * the standard library. It provides input and output functionality
4  * to the program.
5  */
6 #include <stdio.h>
7
8 /*
9  * Function (method) declaration. This outputs "Hello, world\n" to
10 * standard output when invoked.
11 */
12 void sayHello(void) {
13     // printf() in C outputs the specified text (with optional
14     // formatting options) when invoked.
15     printf("Hello, world!\n");
16 }
17
18 /*
19 * This is a "main function". The compiled program will run the code
20 * defined here.
21 */
22 int main(void)
23 {
24     // Invoke the sayHello() function.
25     sayHello();
26     return 0;
27 }
```

Visuelle



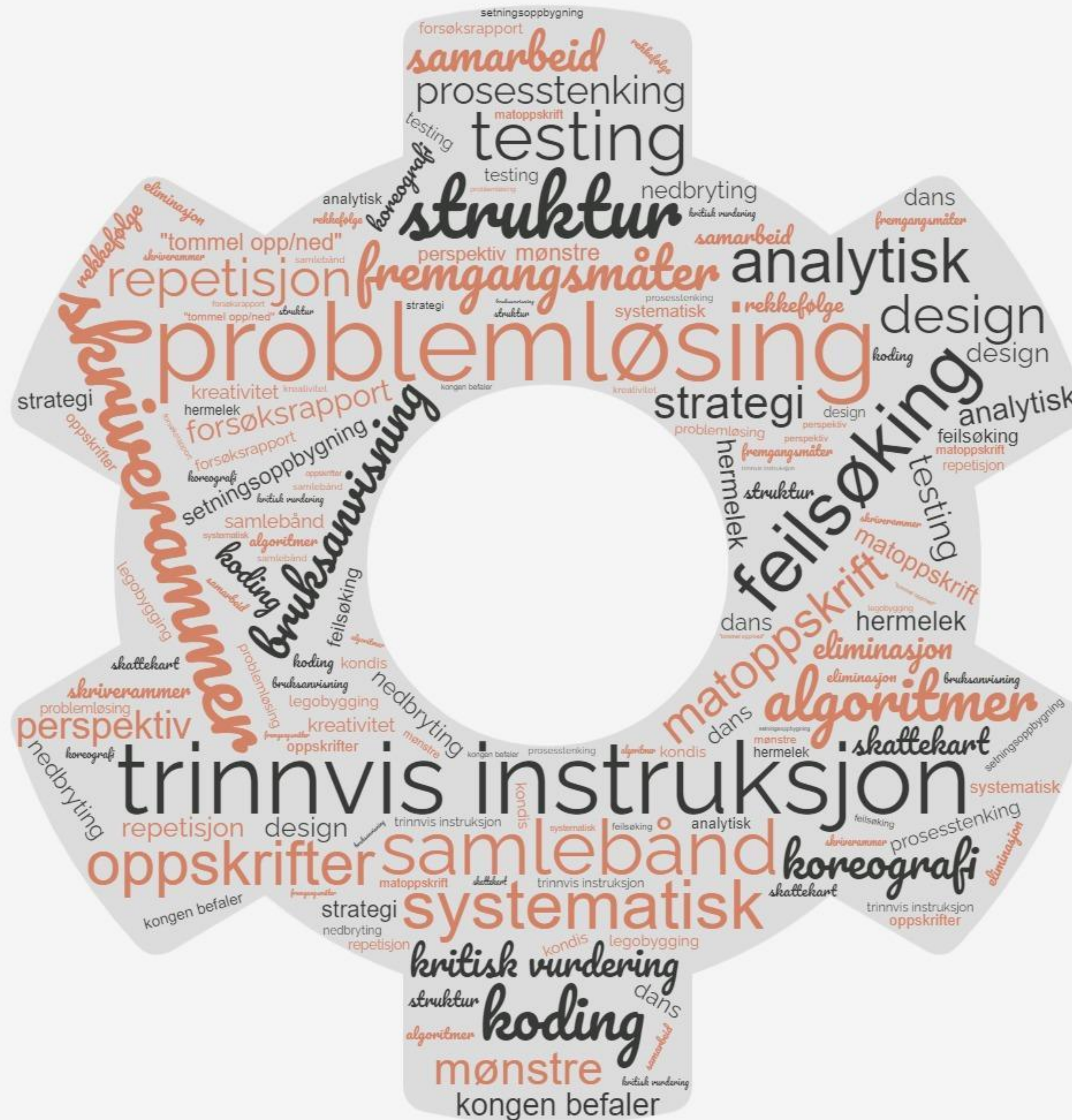
Klarer du å ta grisen?

Åpne nettadressen -> bit.ly/tagrisen



Eksempler på problemløsning med
og uten datamaskin: algoritmisk
tenking og programmering i
fagfornyelsen

Nivå 1



Robot: algoritmisk tenkning på barnetrinnet

Her skal barna lære å:

- omgjøre virkelige handlinger til instruksjoner
- få praktiske erfaringer i å kode med symboler
- få erfaringer med behov for presisjon innen koding



Analog programmering i kroppsøving

KODEDANS, 1.-4. TRINN

Oppgaven passer til:

Fag: Kroppsøving

Trinn: 1.-4. trinn men kan tilpasses eldre

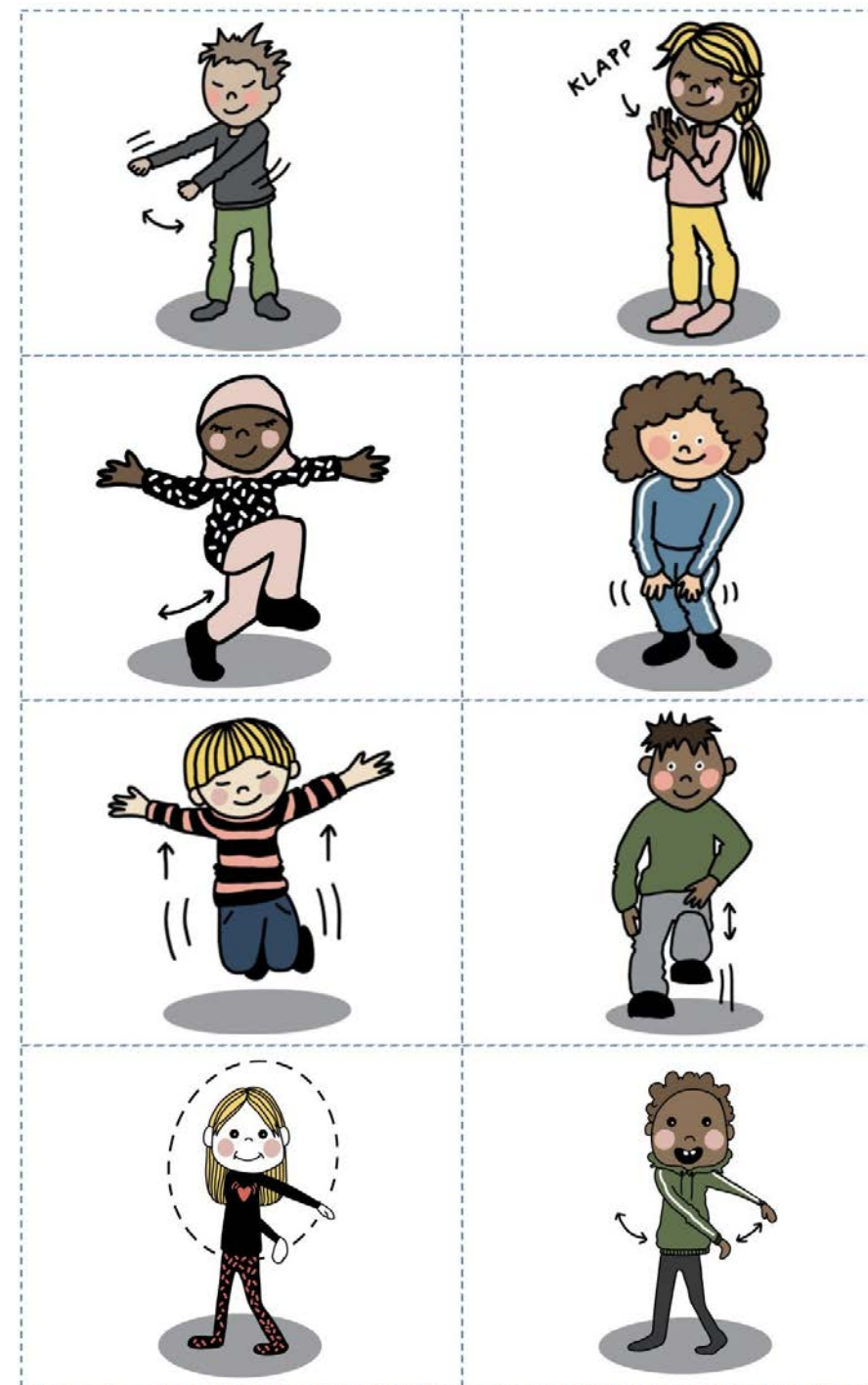
Tema: Dans og koreografi

Mål for kodingen:

- kunne forstå retning, gjentakelser, rekkefølger og algoritmer.
- kunne lage en oppskrift med repetisjoner

Kompetansemål for kroppsøving:

- Utforske egen kroppslig bevegelse i lek og andre aktiviteter, alene og sammen med andre
- utforske og gjennomføre leker, idrettsaktiviteter, danser og andre bevegelsesaktiviteter
- øve på sammensatte bevegelser, alene og sammen med andre

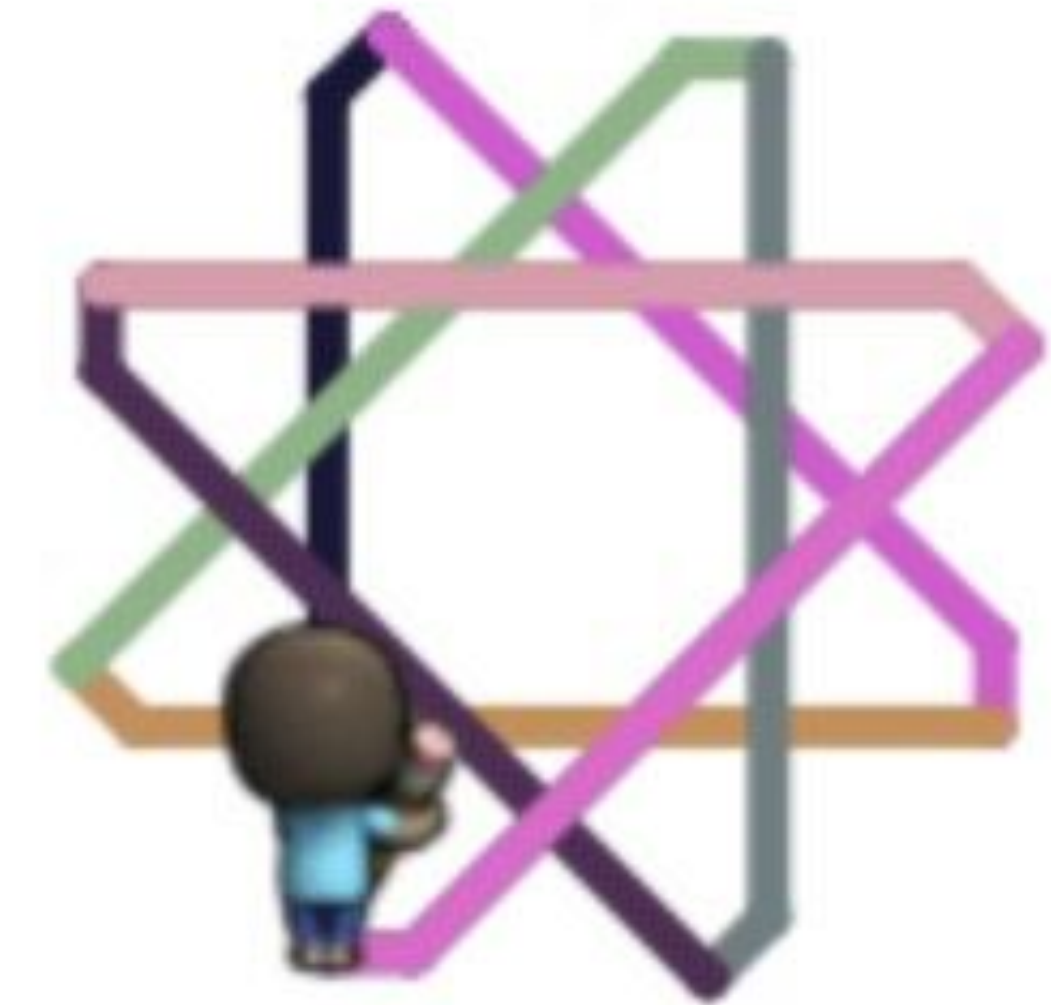


LÆRERVEILEDNING pdf

Eksempel på programmering i matematikk

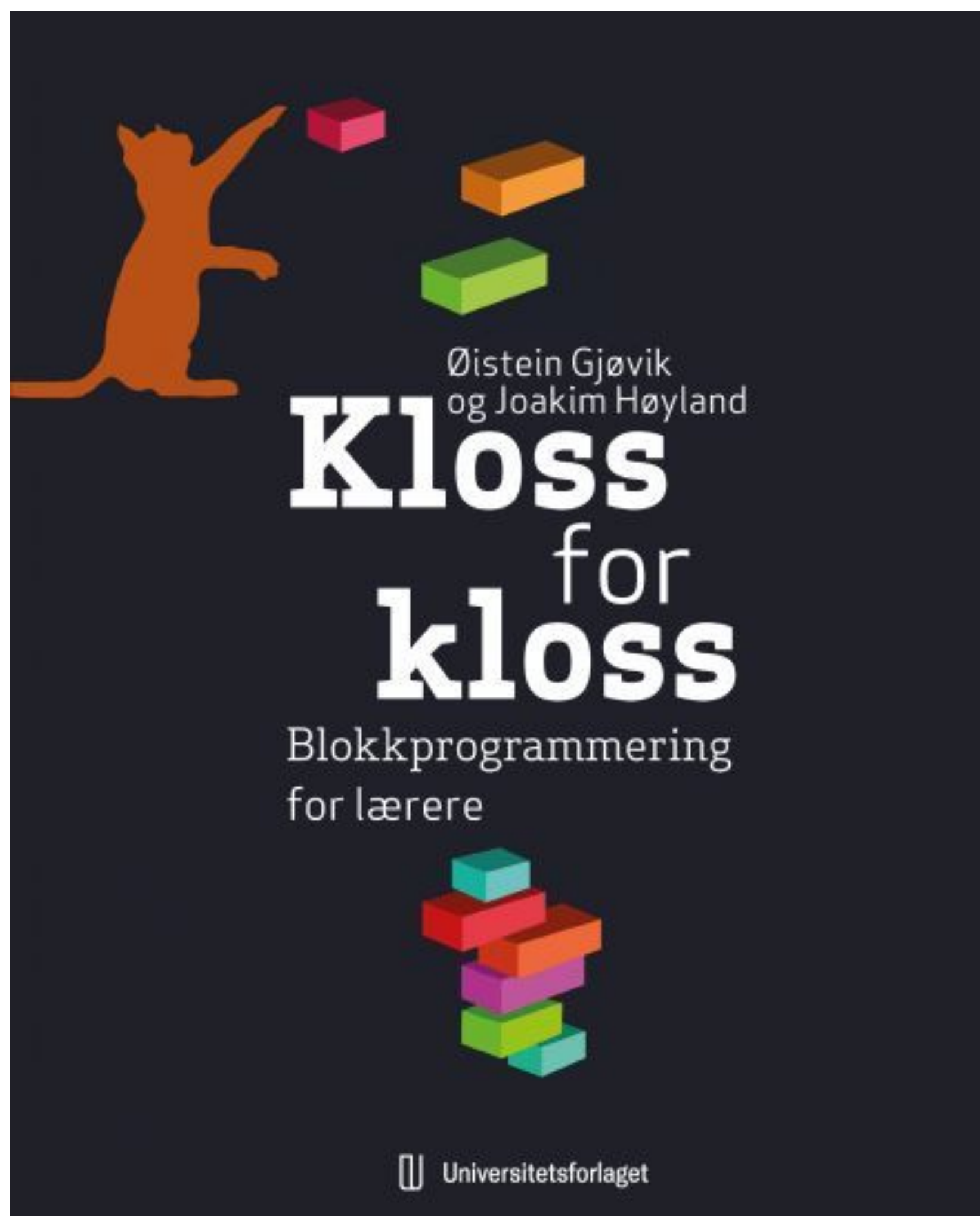
- lage og følge regler og trinnvise instruksjonar i leik og spel (knytt til koordinatsystemet)
2. trinn (3. trinn)
- lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse
5. trinn
- utforske korleis algoritmar kan skapast, testast og forbe­trast ved hjelp av programmering
8. trinn
- bruke programmering til å utforske matematiske eigenskapar og samanhengar
9. trinn

Artist: tegn en figur



<https://studio.code.org/s/artist/stage/1/puzzle/10>

Ressurser



Linda Liukas: Hei Ruby-bøkene



[Bit.ly/borkedalnett](https://bit.ly/borkedalnett)



Ressursside for Samling 4



Eksempler på aktiviteter

Her finner du både analoge, frakoblede programmeringsoppgaver og digitale programmeringsoppgaver.

I nettressursen for samling 4 finner du eksempler på analoge, frakoblede programmeringsoppgaver og digitale programmeringsoppgaver.

Hvordan gjennomføre oppgave ved bruk av nøkkelelementer i algoritmisk tenkning

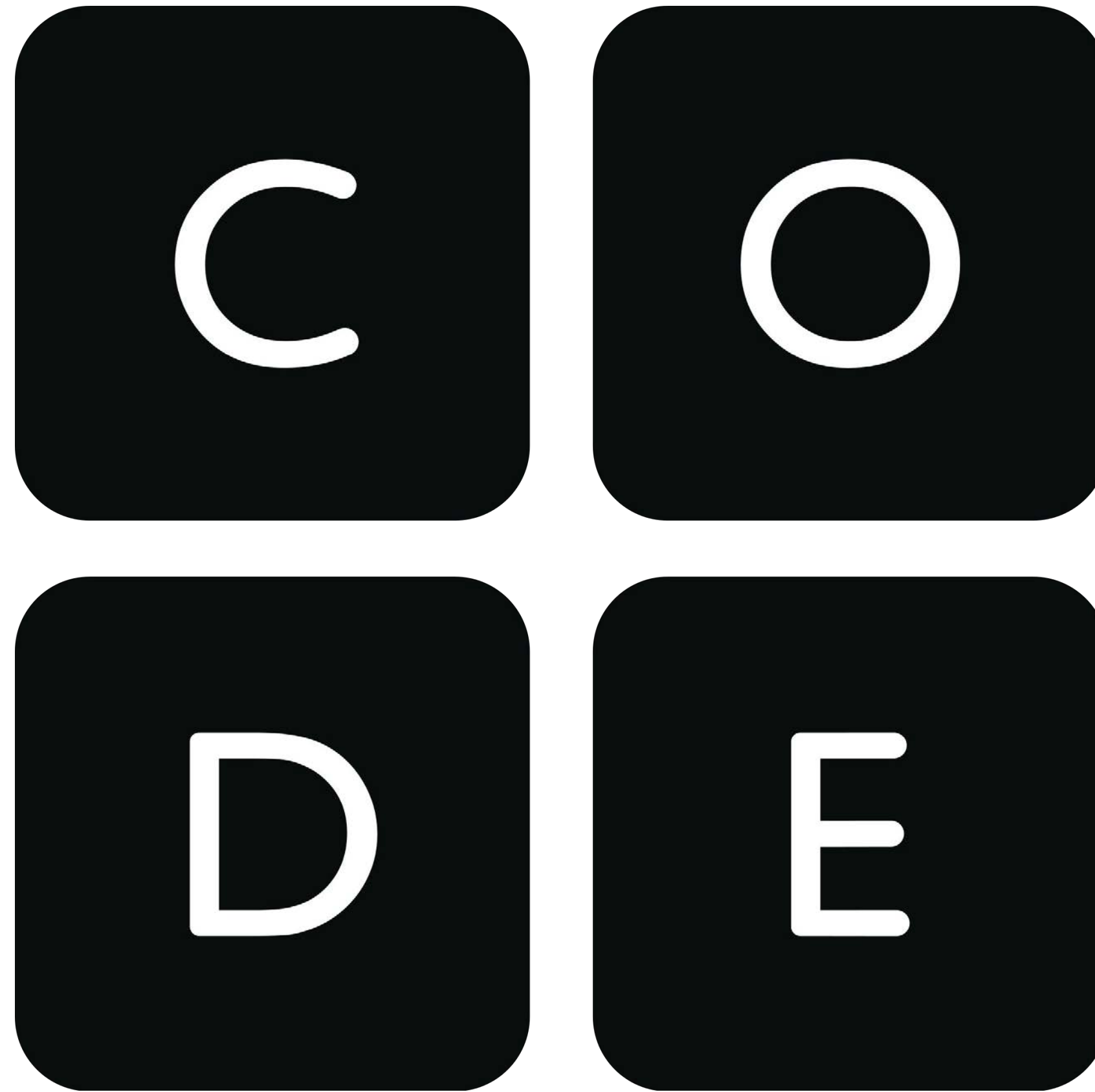
Bruk av algoritmisk tankegang

Gjennomføre oppgave ved bruk av nøkkelelementer i algoritmisk tenkning:

Dekomposisjon	Hva er oppgaven? Hvilke deler består oppgaven av? Hvilke deler av oppgaven skal løses ved bruk av datamaskin? Trengs det utstyr for å løse oppgaven? Hva slags utstyr? Trengs det forberedelser før vi kan løse oppgaven?
Logikk	I hvilken rekkefølge skal vi løse oppgaven?
Mønstre	Finnes det en allerede kjent fremgangsmåte for denne oppgaven? Kan vi bruke den? Hva må/bør endres?
Abstraksjon	Hva må vi gjøre for å løse oppgaven, og hva kan vi vente med/kutte ut? Kan oppgaven løses på forskjellige nivåer? På hvilket nivå skal vi prøve å løse oppgaven?
Algoritmer	Med hvilken fremgangsmåte kan vi løse hvert steg i oppgaven? Vi lager en presis steg-for-steg-modell for løsningen av oppgaven. Vi kan lage en stikkordsliste, et flytskjema, en oppgaveløsningsramme osv.
Evaluering	Hvilke kriterier skal brukes for å kontrollere løsningen? Vi lager en tydelig oversikt over hva som kjennetegner en god løsning av oppgaven. Vi gjennomfører oppgaven ved hjelp av modellen vi har laget, og vurderer den. Hvis vi ikke har løst oppgaven slik vi ønsket, feilsøker vi ved å bruke stegene over: dekomposisjon, logikk, mønstre, abstraksjon og algoritmer, før vi evaluerer igjen.

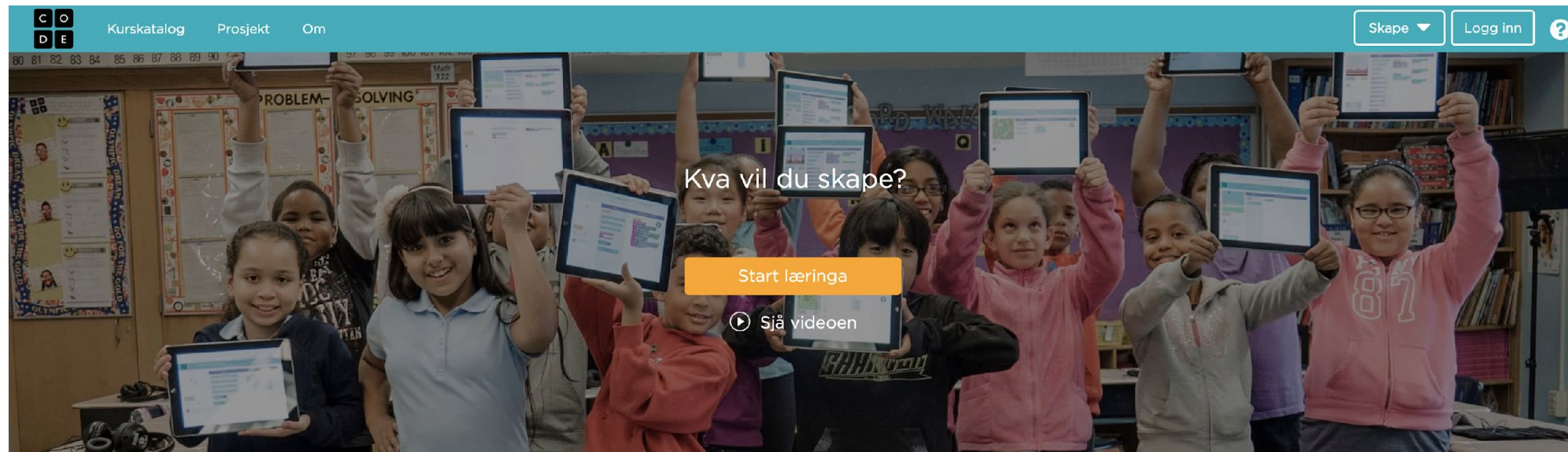


Vi har utarbeidet et hjelpeverktøy i veiledning av elever. Dette kan du bruke for å lære elever om fremgangsmåten ved algoritmisk tenkning. Du finner den på siden [Veiledning](#)



Code.org er en ideell organisasjon og har som visjon at hver elev i hver skole skal ha anledning til å lære datavitenskap, på lik linje som biologi, kjemi eller algebra.

Code.org blir støtta av Amazon, Facebook, Google, Microsoft og mange fleire.



Kva vil du skape?

Start læringa

Sjå videoen

Alle elevar på alle skular burde få høve til å lære koding

Støtt det

Elevar
Utforsk alle opplæringsprogramma våre

Lærarar
Undervis elevane dine


Kodetimen
Alle kan læra. Begynn i dag

Dansefest
Lag og del din eigen dansefest

Over 54 million Code.org projects created




C OD ENivå 4: Kunstnar: Sekvens1MEIRLogg inn?☰



Kjør

Treng du hjelp? Sjå desse videoane og tipsa



Kunstner introduksjon

Instruksjonar

Hei, jeg er en kunstner. Du kan skrive kode som får meg til å tegne nesten hva som helst. Få meg til å tegne over de grå linjene i bilde ved å bruke noen få blokker. (Hver linje er 100 piksler lang)

Blokker

Arbeidsområde: 1 / 4 blokker

når den køyrer

- move forward by 100 piksler
- turn right by 90 grader
- turn left by 90 grader
- hopp framover med 100 piksler
Flyttar kunstnaren sporlaust.
- set farge
- set farge tilfeldig farge

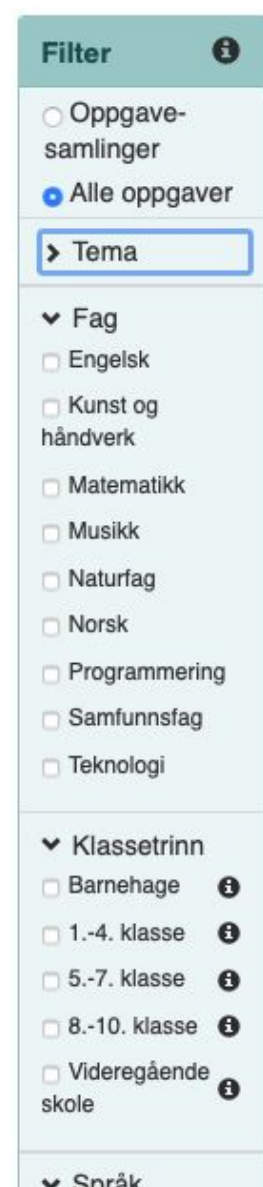
Norsk (Ny) | Personvern | Opphavsrett | Meir



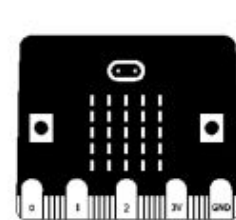
Lær Kidsa Koding! er en frivillig bevegelse som arbeider for at barn og unge skal lære å forstå og beherske sin egen rolle i det digitale samfunnet.

Her finner man mange opplegg laget på norsk tilpasset fag, klassetrinn og programmeringstype

NB: lærermodus



Kurs



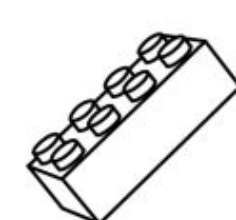
Micro:bit
Oppgaver: 50



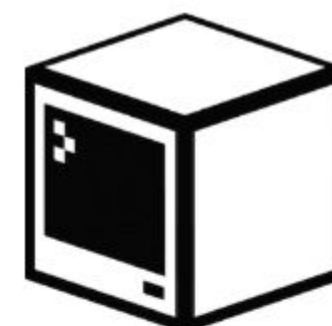
Scratch
Oppgaver: 43



Python
Oppgaver: 35



LEGO Mindstorms
Oppgaver: 24



ComputerCraft
Oppgaver: 8



App Inventor
Oppgaver: 5



Arduino
Oppgaver: 5



Learn To Mod
Oppgaver: 5



Web
Oppgaver: 20



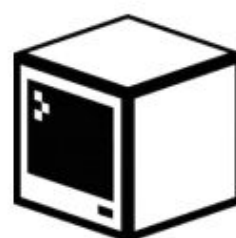
CodeStudio
Oppgaver: 15



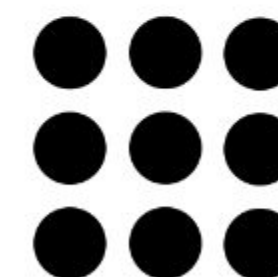
Processing
Oppgaver: 13



Eim
Oppgaver: 11



Uten datamaskin
Oppgaver: 5



Diverse
Oppgaver: 4



Dagens aktiviteter:

[Bit.ly/borkedalnett](https://bit.ly/borkedalnett)



Ressursside for Samling 4

Nivå 1

Vi foreslår at du prøver ut **code.org** og/eller noen av eksemplene på analoge eller digitale aktiviteter



Eksempler på aktiviteter

Eksempler på analog og digital programmering i skolen

Velg et verktøy



Code.org

Lær å kode med kodekurs

Mellomarbeid

1. Velg deg ut et av dine fag og ta for deg et tema der det kan egne seg å lage en aktivitet knyttet mot algoritmisk tenkning og/eller programmering.
2. Gå til nettressursen for samling 4 – gå så inn på [eksempler på aktiviteter](#).
3. Finn et opplegg som passer deg. Test ut med elevene.
4. Hvordan gikk det? Del tanker og refleksjoner med kollegaer - og vi tar det opp på neste samling i november.

Undersøkelse - Hva syntes du om samlingen i dag?

bit.ly/borkekart4





Nivå 2

Eksempler fra planverket hvor digitalisering blir konkret satt opp som kompetansemål

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kompetansemål etter 7. trinn

- bruke teknologi og digitale verktøy til å skape, øve inn og bearbeide musikk

Kompetansemål etter 4. trinn

- eksperimentere med rytmer, melodier og andre grunnelementer, sette sammen mønstre til komposisjoner, også ved bruk av digitale verktøy, og beskrive arbeidsprosesser og resultater

Kompetansemål etter 2. trinn

- leke med musikkens grunnelementer gjennom lyd og stemme, lage mønstre og sette sammen mønstrene til enkle improvisasjoner og komposisjoner, også med digitale verktøy

Kunst og håndverk

Kompetansemål etter 10. trinn

- utforske hvordan digitale verktøy og ny teknologi kan gi muligheter for kommunikasjonsformer og opplevelser i skapende prosesser og produkter
- visualisere form ved hjelp av frihåndstegninger, arbeidstegninger, modeller og digitale verktøy

Kompetansemål etter 7. trinn

- bruke digitale verktøy til å planlegge og presentere prosesser og produkter
- analysere visuelle virkemidler i ulike medier og belyse et aktuelt tema gjennom foto eller infografikk
- bruke programmering til å skape interaktivitet og visuelle uttrykk

Kompetansemål etter 4. trinn

- bruke enkle komposisjonsprinsipper i fotografi og digitale verktøy

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener
- utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker
- bruke og lage modeller for å forutsi eller beskrive naturfaglige prosesser og systemer og gjøre rede for modellenes styrker og begrensinger
- analysere og bruke innsamlede data til å lage forklaringer, drøfte forklaringene i lys av relevant teori og vurdere kvaliteten på egne og andres utforskinger

Kompetansemål etter 7. trinn

- designe og lage et produkt basert på brukerbehov
- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Kompetansemål etter 4. trinn

- utforske teknologiske systemer som er satt sammen av ulike deler, og beskrive hvordan delene fungerer og virker sammen
- designe og lage et produkt basert på en kravspesifikasjon

Kompetansemål etter 2. trinn

- presentere funnene sine og beskrive hvordan eleven har kommet fram til dem
- presentere egne ideer til teknologiske oppfinnelser

Norsk

Kompetansemål etter 10. trinn

- utforske og vurdere hvordan digitale medier påvirker og endrer språk og kommunikasjon
- bruke kilder på en kritisk måte, markere sitater og vise til kilder på en etterrettelig måte i egne tekster

Kompetansemål etter 7. trinn

- orientere seg i faglige kilder på bibliotek og digitalt, vurdere hvor pålitelige kildene er, og vise til kilder i egne tekster
- reflektere etisk over hvordan eleven framstiller seg selv og andre i digitale medier
- presentere faglige emner muntlig med og uten digitale ressurser
- utforske og beskrive samspillet mellom skrift, bilder og andre uttrykksformer og lage egne sammensatte tekster

Kompetansemål etter 4. trinn

- holde muntlige presentasjoner med og uten digitale ressurser

Kompetansemål etter 2. trinn

- lese med sammenheng og forståelse på papir og digitalt og bruke enkle strategier for leseforståelse
- lage tekster som kombinerer skrift med bilder

Samfunnsfag

Kompetansemål etter 10. trinn

- bruke samfunnsfaglege metodar og digitale ressursar i egne undersøkingar, presentere funn ved bruk av digitale verktøy og drøfte kor gyldige og relevante funna er
- vurdere på kva måtar ulike kjelder gir informasjon om eit samfunnsfagleg tema, og reflektere over korleis algoritmar, einsretta kjelder eller mangel på kjelder kan prege forståinga vår
- utforske korleis teknologi har vore og framleis er ein endringsfaktor, og drøfte innverknaden teknologien har hatt og har på enkeltmenneske, samfunn og natur
- utforske og reflektere over eigne digitale spor og høvet til å få sletta spora og å verne om retten ein sjølv og andre har til privatliv, personvern og opphavsrett
- utforske ulike plattformer for digital samhandling og reflektere over korleis digital deltaking og samhandling påverkar forma på og innhaldet i samfunnsdebatten

Kompetansemål etter 7. trinn

- gjennomføre ei samfunnsfagleg undersøking og presentere resultatata ved hjelp av eigna digitale verktøy
- presentere ei aktuell nyheits sak og reflektere over forskjellar mellom fakta, meiningar og kommersiell budskap i mediebiletet
- utforske korleis menneske i fortida livnærte seg, og samtale om korleis sentrale endringar i livsgrunnlag og teknologi har påverka og påverkar demografi, levekår og busetjingsmønster
- reflektere over korleis ein sjølv og andre deltek i digital samhandling, og drøfte kva det vil seie å bruke dømmekraft sett i lys av reglar, normer og grenser

Kompetansemål etter 4. trinn

- samtale om reglar og normer for personvern, deling og beskyttelse av informasjon og om kva det vil seie å bruke dømmekraft i digital samhandling
- utforske og presentere samfunnsfaglege spørsmål, søkje etter informasjon i ulike kjelder og vurdere kor nyttig informasjonen er til å belyse spørsmåla

Kompetansemål etter 2. trinn

- samtale om moglegheiter og utfordringar ved digital samhandling

Matte

Kompetansemål etter 10. trinn

- utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- bruke funksjonar i modellering og argumentere for framgangsmåtar og resultat
- modellere situasjonar knytte til reelle datasett, presentere resultatata og argumentere for at modellane er gyldige
- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering
- utforske og argumentere for korleis framstillingar av tal og data kan brukast for å fremje ulike synspunkt

Kompetansemål etter 8. trinn

- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett
- lage og vurdere budsjett og rekneskap ved å bruke rekneark med cellereferansar og formlar

Kompetansemål etter 6. trinn

Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Kompetansemål etter 4. trinn

- lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer

Kompetansemål etter 3. trinn

- lage og følgje reglar og trinnvise instruksjonar i leik og spel knytte til koordinatsystemet

Kompetansemål etter 2. trinn

- lage og følgje reglar og trinnvise instruksjonar i leik og spel

Eksempler fra planverket hvor programmering blir konkret satt opp som kompetansemål

Musikk

Kompetansemål etter 10. trinn

- skape og programmere musikalske forløp ved å eksperimentere med lyd fra ulike kilder

Kunst og håndverk

Kompetansemål etter 7. trinn

- bruke programmering til å skape interaktivitet og visuelle uttrykk

Naturfag

Kompetansemål etter 10. trinn

- bruke programmering til å utforske naturfaglige fenomener

Kompetansemål etter 7. trinn

- utforske, lage og programmere teknologiske systemer som består av deler som virker sammen

Matte

Kompetansemål etter 10. trinn

- utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Kompetansemål etter 9. trinn

- simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe ved å bruke programmering

Kompetansemål etter 8. trinn

- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering

Kompetansemål etter 7. trinn

- bruke programmering til å utforske data i tabellar og datasett

Kompetansemål etter 6. trinn

- bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster

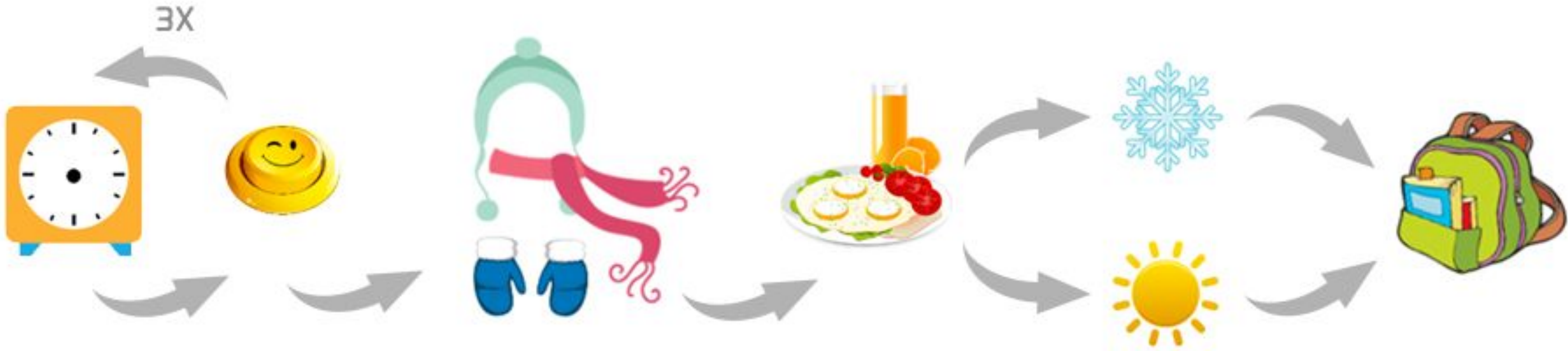
Kompetansemål etter 5. trinn

- lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Algoritmisk tenkning

En algoritme kan forklares som en meget detaljert trinn-for-trinn oppskrift som må følges i nøyaktig riktig rekkefølge.

Et eksempel på en algoritme



```
when Alarm Rings occurs
repeat 3
  Hit Snooze Button
  wait until Alarm Rings
Turn Off Alarm
Get Up
Shower
Get Dressed
Eat breakfast
if weather = sunny then
  Walk to School
else
  Ride to School with Parent
```

Bilde hentet fra: <https://www.tynker.com/blog/articles/ideas-and-tips/how-to-explain-algorithms-to-kids/>

Den algoritmiske tenkeren

Nøkkelbegrep

- 1** **Logikk**
Analysere og forutse
- 2** **Algoritmer**
Regler og steg-for-steg
- 3** **Dekomposisjon**
Bryte ned i mindre deler
- 4** **Mønstre**
Finne og bruke likheter
- 5** **Abstraksjon**
Fjerne unødvendige detaljer
- 6** **Evaluering**
Gjøre vurderinger



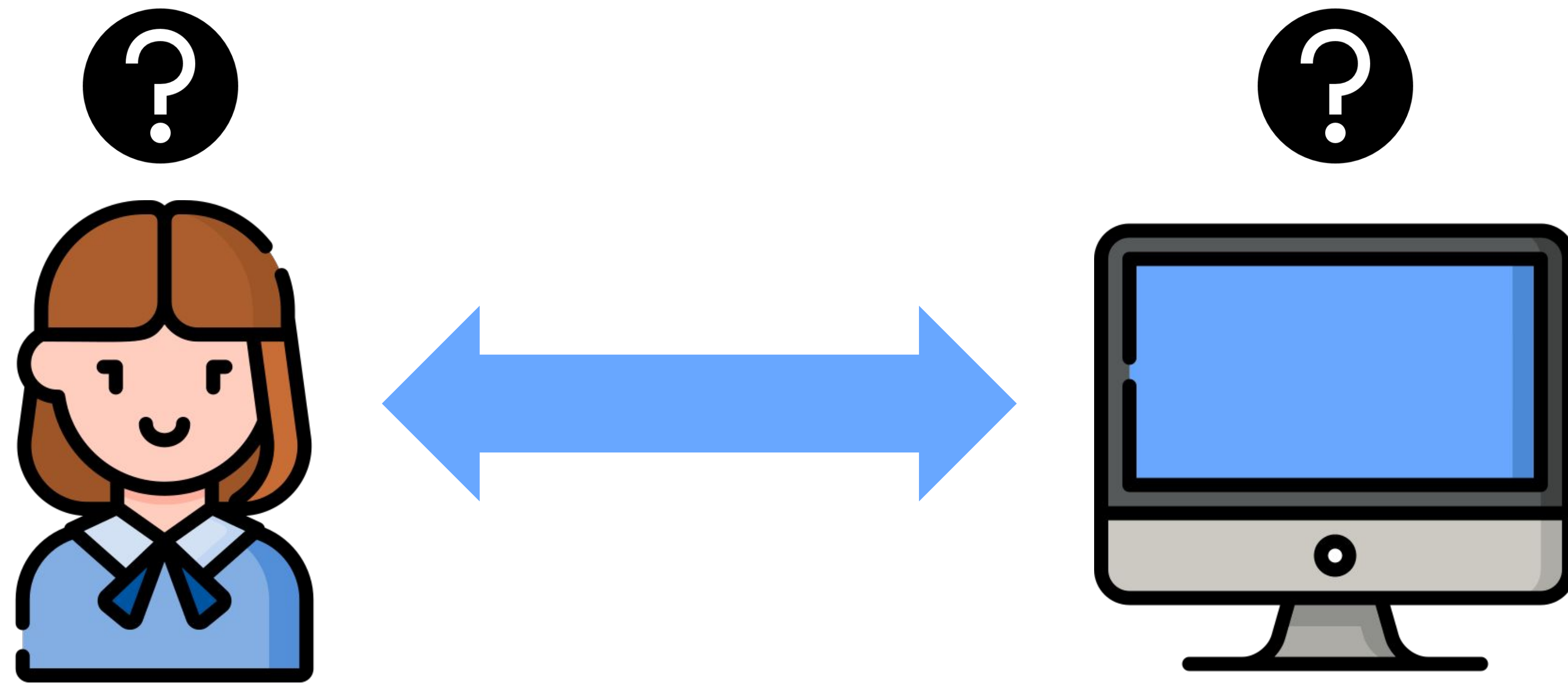
Arbeidsmåter

- ✓ **Fikle**
Utforske og eksperimentere
- ✓ **Skape**
Designe og lage
- ✓ **Feilsøke**
Oppdage og rette feil
- ✓ **Holde ut**
Fortsette og prøve igjen
- ✓ **Samarbeide**
Dele og jobbe sammen

Algoritmisk tenkning

- Problemløsningsmetode
- Systematisk
- Vurdering: teknologi vs. menneske
- *Algoritmisk tenkning er den norske oversettelsen av det engelske ordet computational thinking.*

Hva er programmering?



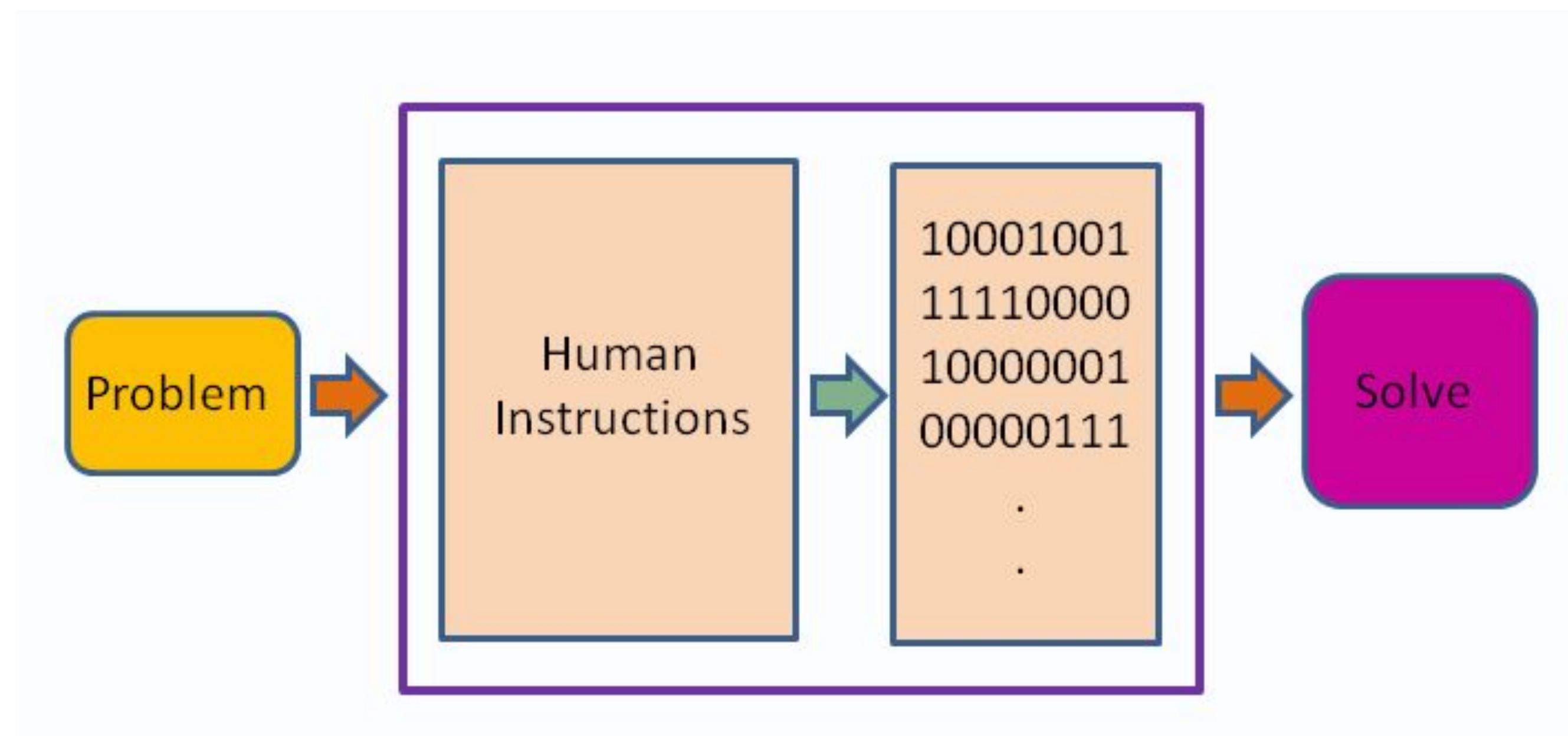


"Med programmering gjelder det å kunne formulere et problem og finne løsningen ved hjelp av et dataprogram."

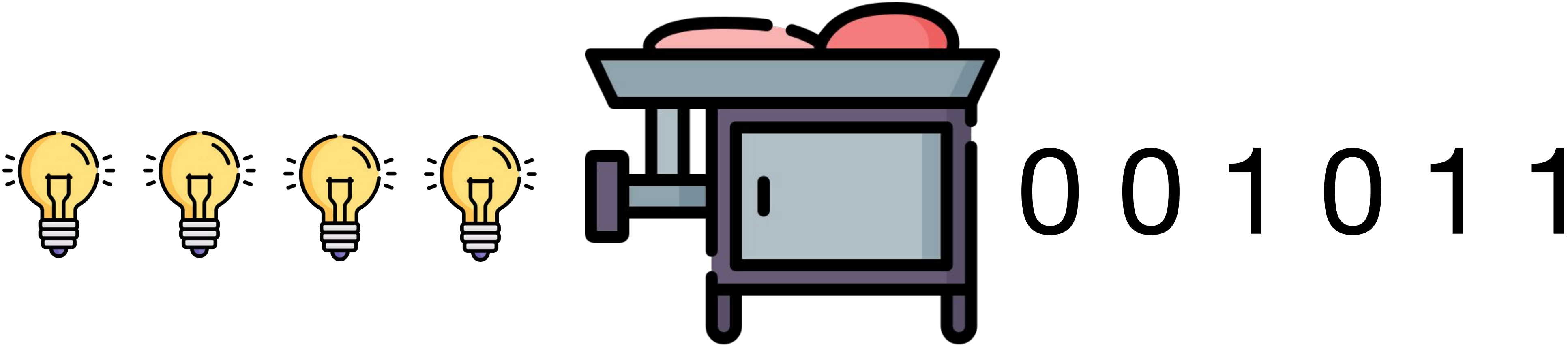
Hentet fra: <https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

«Selve prosessen med å skrive programmet kalles i IT-sjargong ofte for *koding*.»

Hentet fra: <https://www.hiof.no/lusp/slik/forskning/grupper/begynneropplering/aktuelt/aktuelt-2019/lererne-er-positive-til-programmering---men-mangle.html>



Er koding og programmering det samme?



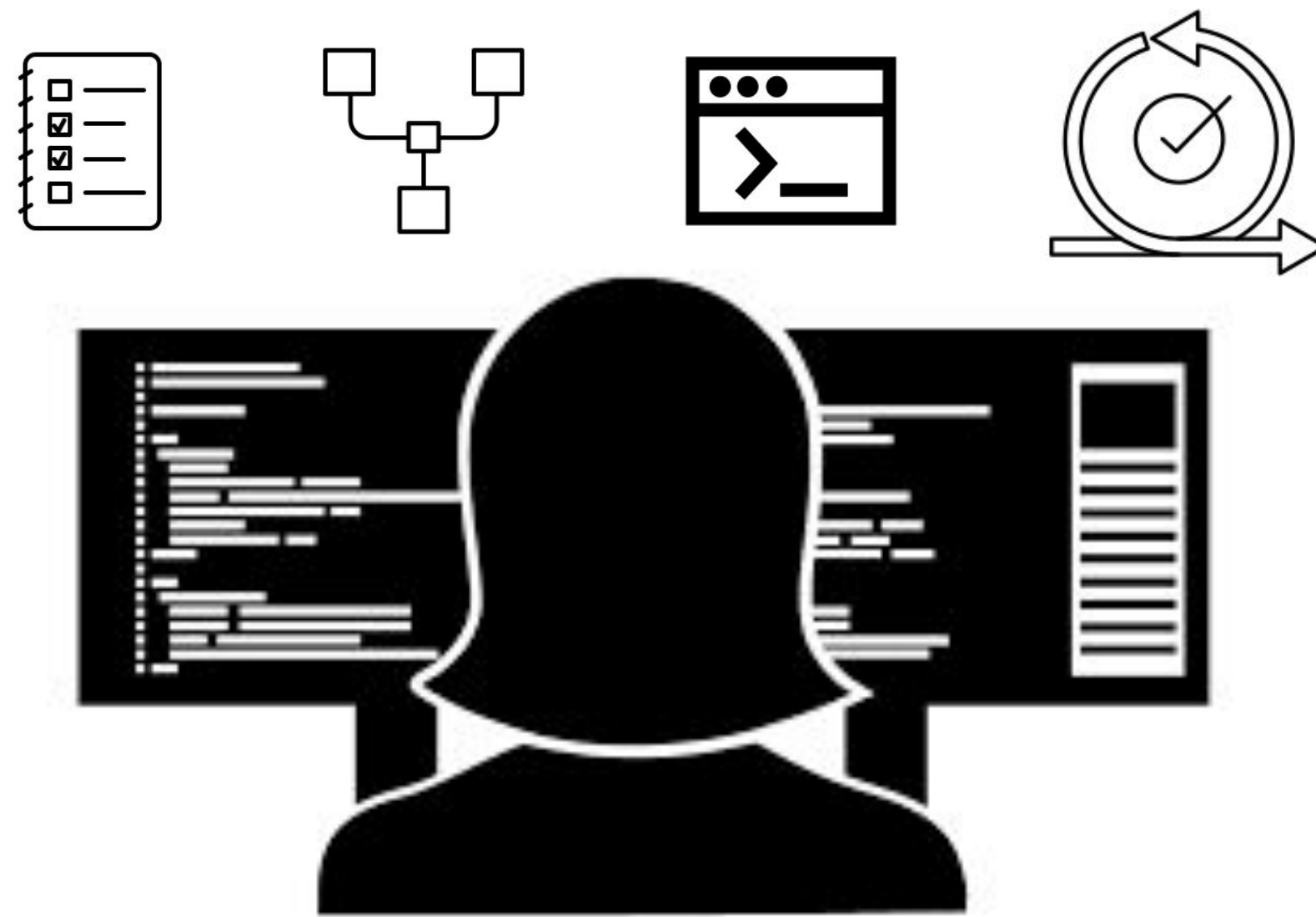
Å lære å kode handler om å lære seg det språket som skal til for å kommunisere med datamaskinen



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Programmering er derimot noe langt mer. Det involverer blant annet også tankeprosessene som går forut for kodingen.



Hentet fra:

<https://www.uv.uio.no/forskning/satsinger/fiks/om/aktuelt/aktuelle-saker/2018/programmering-er-mer-enn-koding%21.html>

Kodespråk

Tekstbaserte

```
1 /*
2  * This line basically imports the "stdio" header file, part of
3  * the standard library. It provides input and output functionality
4  * to the program.
5  */
6 #include <stdio.h>
7
8 /*
9  * Function (method) declaration. This outputs "Hello, world\n" to
10 * standard output when invoked.
11 */
12 void sayHello(void) {
13     // printf() in C outputs the specified text (with optional
14     // formatting options) when invoked.
15     printf("Hello, world!\n");
16 }
17
18 /*
19 * This is a "main function". The compiled program will run the code
20 * defined here.
21 */
22 int main(void)
23 {
24     // Invoke the sayHello() function.
25     sayHello();
26     return 0;
27 }
```

Visuelle



Klarer du å ta grisen?

Åpne nettadressen -> bit.ly/tagrisen



Eksempler på problemløsning med og uten datamaskin: algoritmisk tenking og programmering i fagfornyelsen

Robot: algoritmisk tenkning på barnetrinnet

Her skal barna lære å:

- omgjøre virkelige handlinger til instruksjoner
- få praktiske erfaringer i å kode med symboler
- få erfaringer med behov for presisjon innen koding



Analog programmering i kroppsøving

Oppgaven passer til:

Fag: Kroppsøving

Trinn: 1.-4. trinn men kan tilpasses eldre

Tema: Dans og koreografi

Mål for kodingen:

- kunne forstå retning, gjentakelser, rekkefølger og algoritmer.
- kunne lage en oppskrift med repetisjoner

Kompetansemål for kroppsøving:

- Utforske egen kroppslig bevegelse i lek og andre aktiviteter, alene og sammen med andre
- utforske og gjennomføre leker, idrettsaktiviteter, danser og andre bevegelsesaktiviteter
- øve på sammensatte bevegelser, alene og sammen med andre



KODEDANS, 1.-4. TRINN



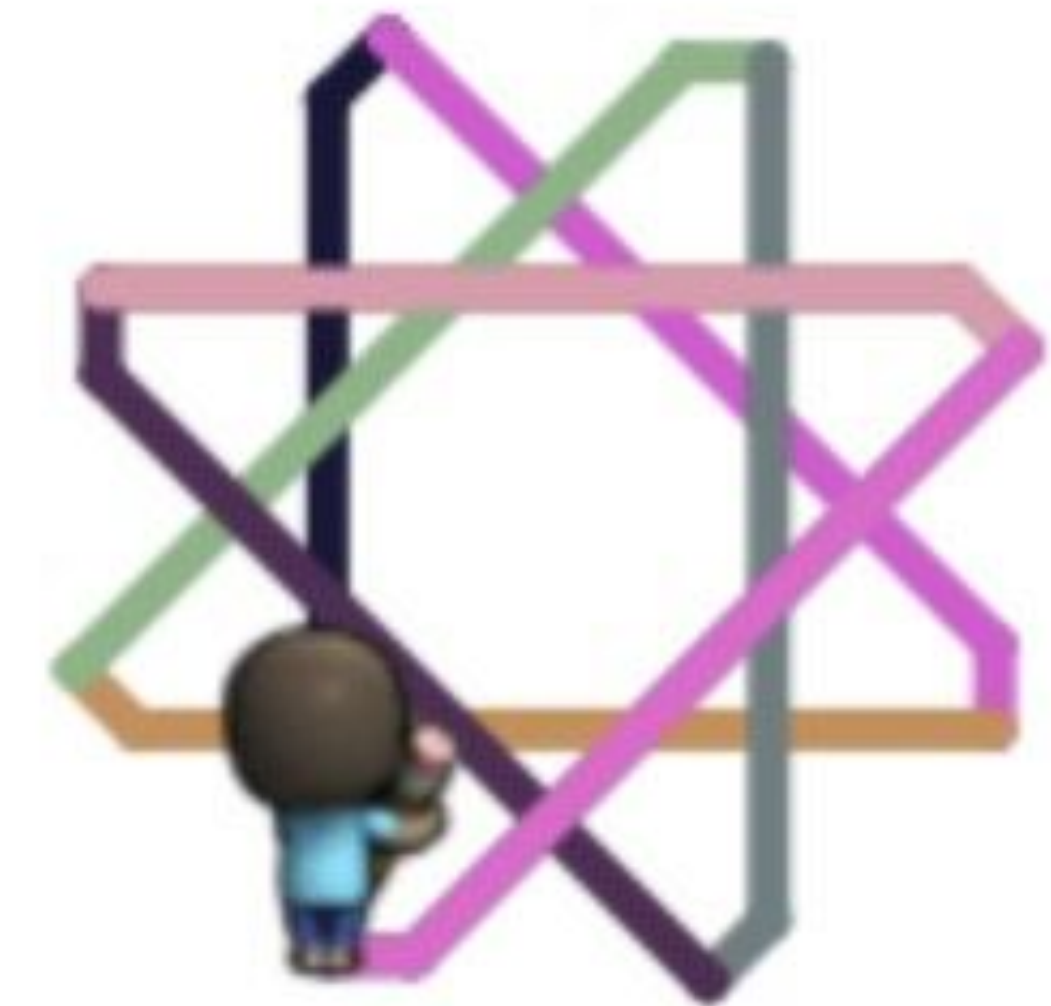
LÆRERVEILEDNING pdf

<https://skaperskolen.no/1-4-trinn/kodedans-1-4-trinn/>

Eksempel på programmering i matematikk

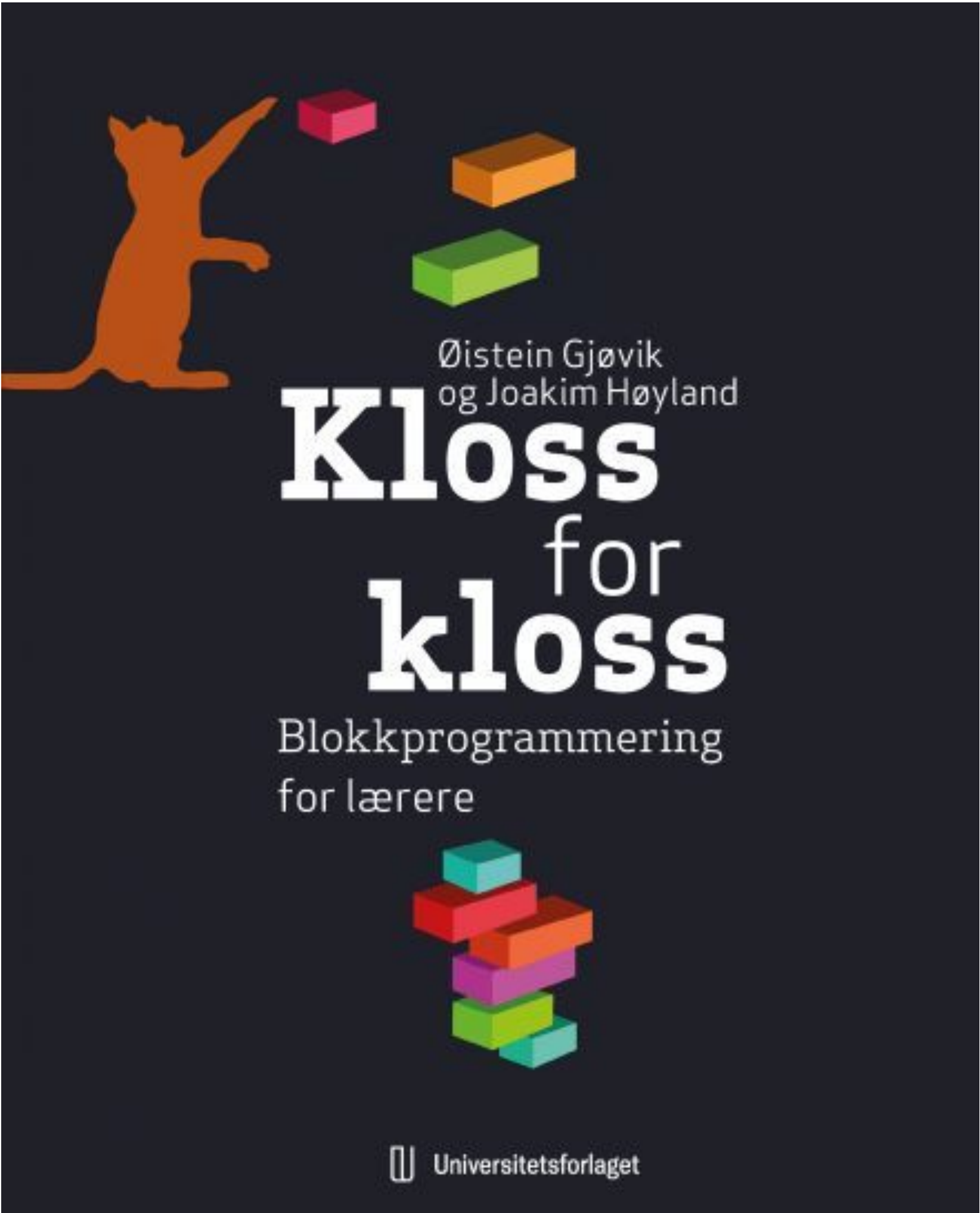
- lage og følge regler og trinnvise instruksjonar i leik og spel (knytt til koordinatsystemet)
2. trinn (3. trinn)
- lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse
5. trinn
- utforske korleis algoritmar kan skapast, testast og forbe­trast ved hjelp av programmering
8. trinn
- bruke programmering til å utforske matematiske eigenskapar og samanhengar
9. trinn

Artist: tegn en figur



<https://studio.code.org/s/artist/stage/1/puzzle/10>

Ressurser



Linda Liukas: Hei Ruby-bøkene



[Bit.ly/borkedalnett](https://bit.ly/borkedalnett)



Ressursside for Samling 4



Eksempler på aktiviteter

Her finner du både analoge, frakoblede programmeringsoppgaver og digitale programmeringsoppgaver.

I nettressursen for samling 4 finner du eksempler på analoge, frakoblede programmeringsoppgaver og digitale programmeringsoppgaver.

Hvordan gjennomføre oppgave ved bruk av nøkkelementer i algoritmisk tenkning

Bruk av algoritmisk tankegang

Gjennomføre oppgave ved bruk av nøkkelementer i algoritmisk tenkning:

Dekomposisjon	Hva er oppgaven? Hvilke deler består oppgaven av? Hvilke deler av oppgaven skal løses ved bruk av datamaskin? Trengs det utstyr for å løse oppgaven? Hva slags utstyr? Trengs det forberedelser før vi kan løse oppgaven?
Logikk	I hvilken rekkefølge skal vi løse oppgaven?
Mønstre	Finnes det en allerede kjent fremgangsmåte for denne oppgaven? Kan vi bruke den? Hva må/bør endres?
Abstraksjon	Hva må vi gjøre for å løse oppgaven, og hva kan vi vente med/kutte ut? Kan oppgaven løses på forskjellige nivåer? På hvilket nivå skal vi prøve å løse oppgaven?
Algoritmer	Med hvilken fremgangsmåte kan vi løse hvert steg i oppgaven? Vi lager en presis steg-for-steg-modell for løsningen av oppgaven. Vi kan lage en stikkordsliste, et flytskjema, en oppgaveløsningsramme osv.
Evaluering	Hvilke kriterier skal brukes for å kontrollere løsningen? Vi lager en tydelig oversikt over hva som kjennetegner en god løsning av oppgaven. Vi gjennomfører oppgaven ved hjelp av modellen vi har laget, og vurderer den. Hvis vi ikke har løst oppgaven slik vi ønsket, feilsøker vi ved å bruke stegene over: dekomposisjon, logikk, mønstre, abstraksjon og algoritmer, før vi evaluerer igjen.



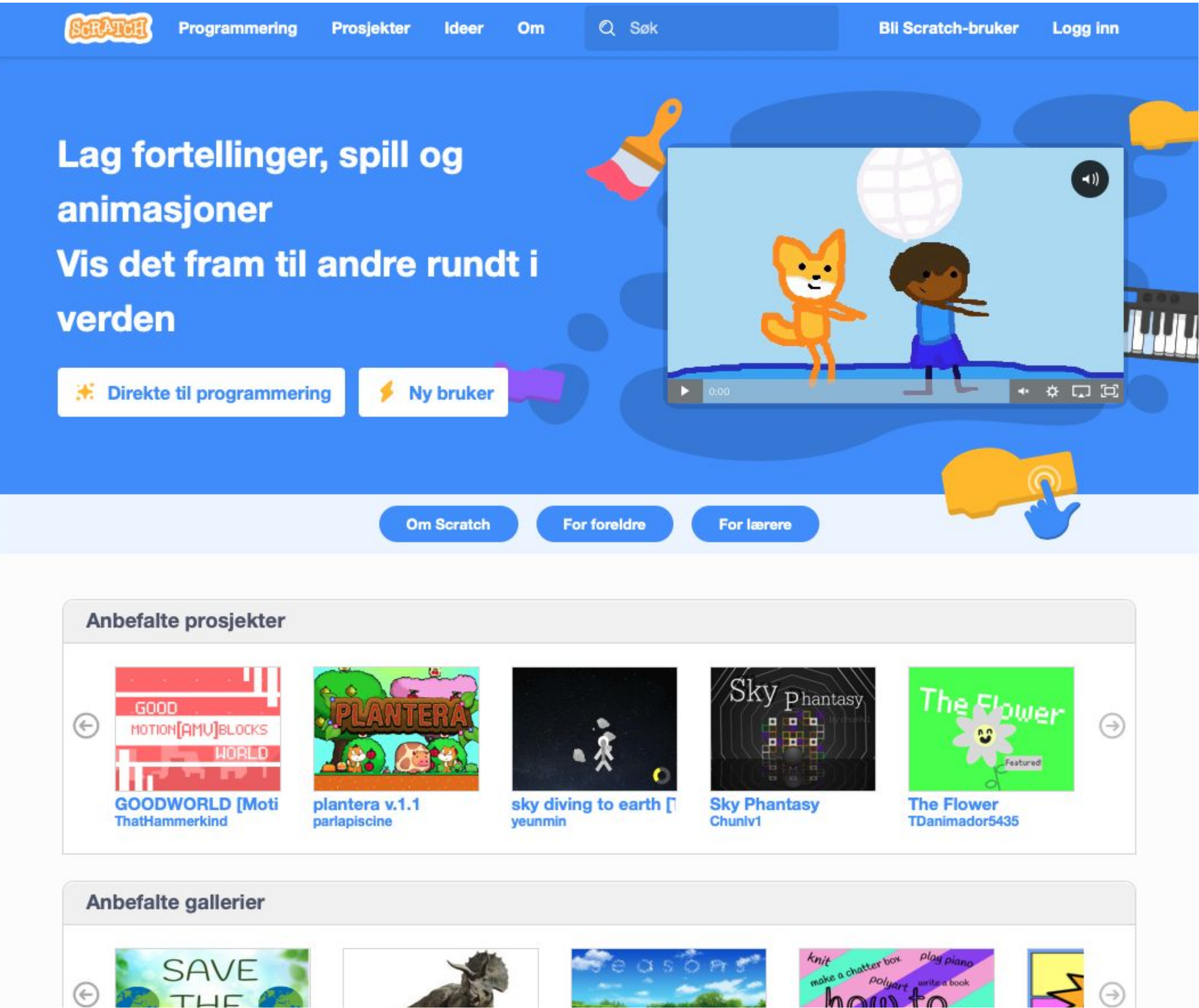
Vi har utarbeidet et hjelpeverktøy i veiledning av elever. Dette kan du bruke for å lære elever om fremgangsmåten ved algoritmisk tenkning. Du finner den på siden [Veiledning](#)

SCRATCH

Elever kan bruke Scratch til å programmere sine egne interaktive fortellinger, animasjoner og spill. Samtidig lærer de å være kreative, tenke systematisk og samarbeide med andre. I skolen brukes Scratch i mange forskjellige fag og på mange klassetrinn, fra barneskole til videregående.

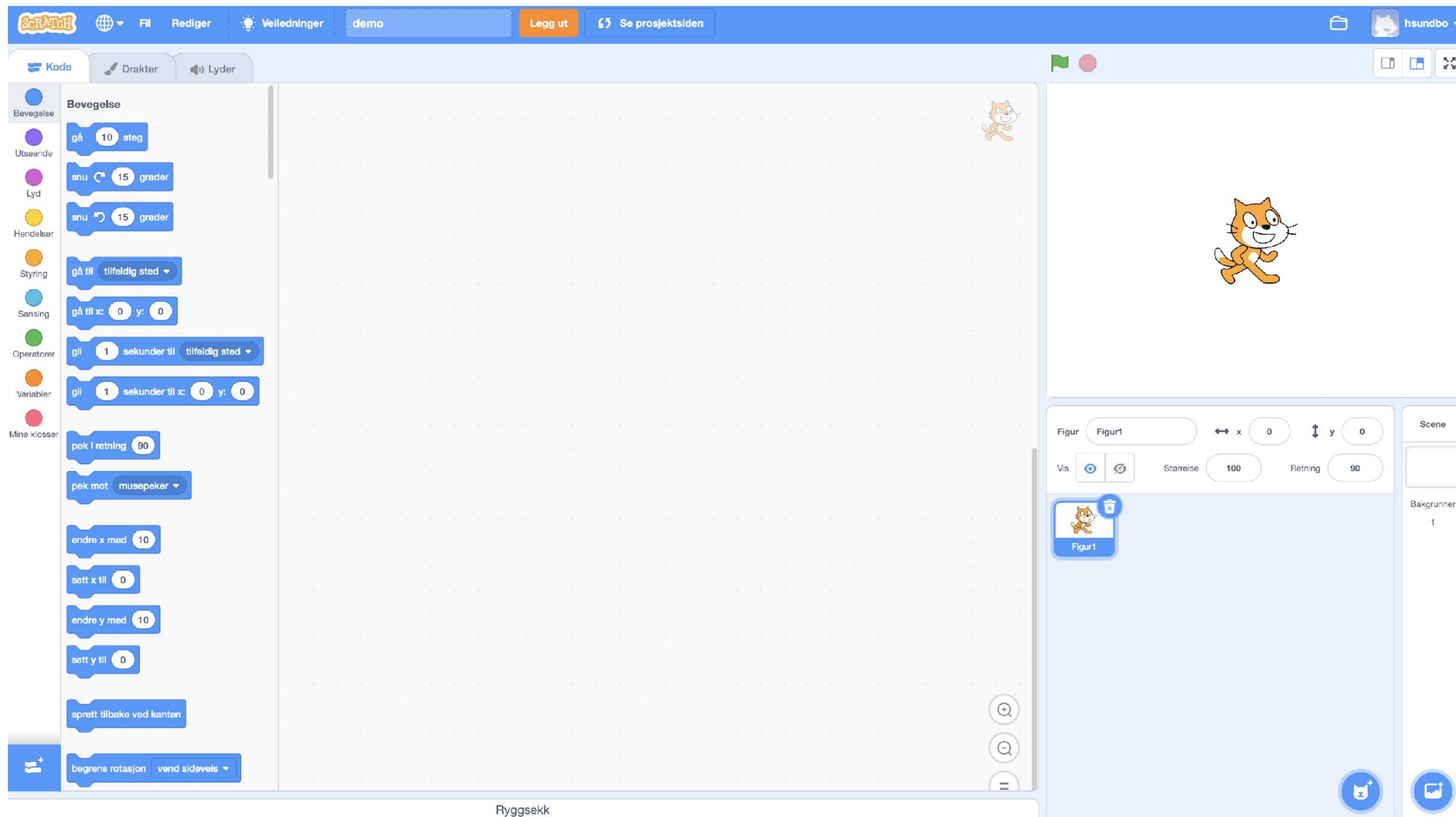


Introvideo om Scratch: <https://www.youtube.com/watch?v=98awWpkx9UM>



The image shows the Scratch website homepage. At the top, there is a navigation bar with the Scratch logo, menu items for 'Programmering', 'Prosjekter', 'Ideer', and 'Om', a search bar with 'Søk', and links for 'Bli Scratch-bruker' and 'Logg inn'. The main header features the text 'Lag fortellinger, spill og animasjoner' and 'Vis det fram til andre rundt i verden'. Below this are two buttons: 'Direkte til programmering' and 'Ny bruker'. A central video player shows a Scratch animation of a cat and a boy. At the bottom of the header are buttons for 'Om Scratch', 'For foreldre', and 'For lærere'. The main content area is divided into two sections: 'Anbefalte prosjekter' and 'Anbefalte gallerier'. The 'Anbefalte prosjekter' section displays five project thumbnails: 'GOODWORLD [Moti ThatHammerkind]', 'plantera v.1.1 parlapiscine', 'sky diving to earth [yeunmin]', 'Sky Phantasy Chunlv1', and 'The Flower TAnimador5435'. The 'Anbefalte gallerier' section displays five gallery thumbnails: 'SAVE THE', a gallery of animals, 'seasons', a 'how to' gallery, and a gallery with various craft projects.

Slik kan det se ut når du starter et tomt prosjekt i Scratch



Test ut Scratch-veiledningene som du finner her i toppmenyen

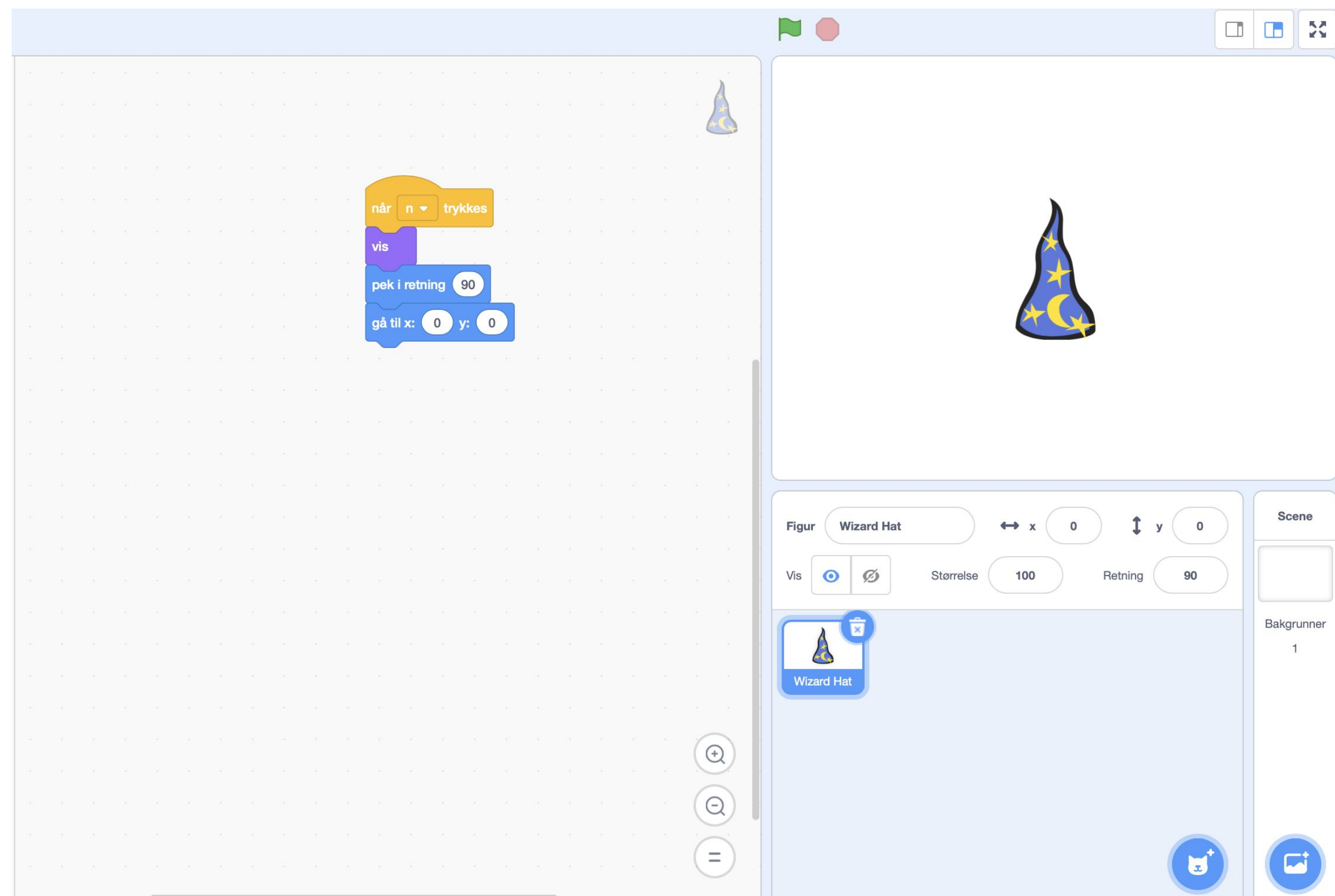
Ressurs:

Prøv veiledningen

«Kom i gang»

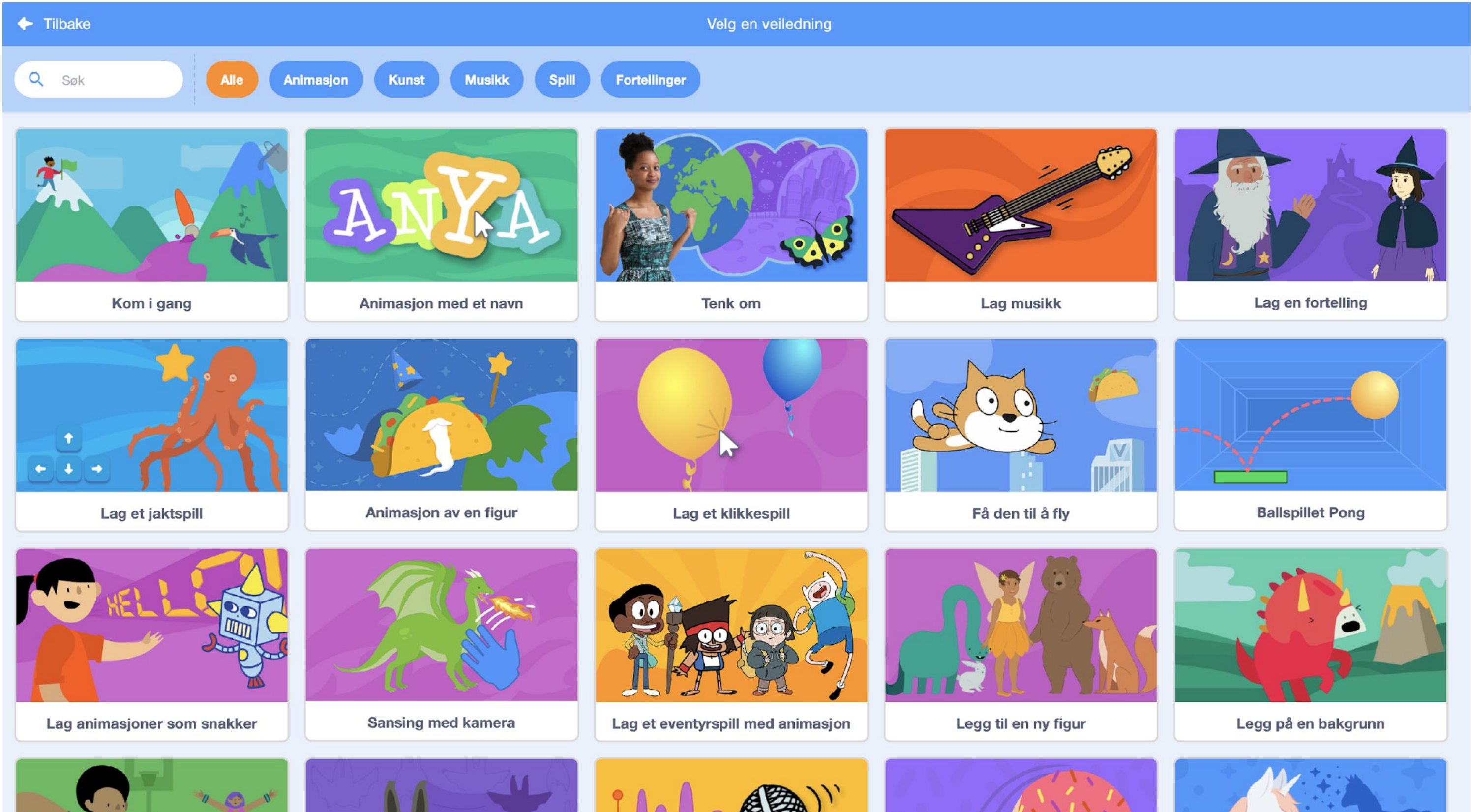
<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Prøv deg på denne oppgaven!



Rotasjon rundt egen akse

I Scratch finner du mange veiledninger



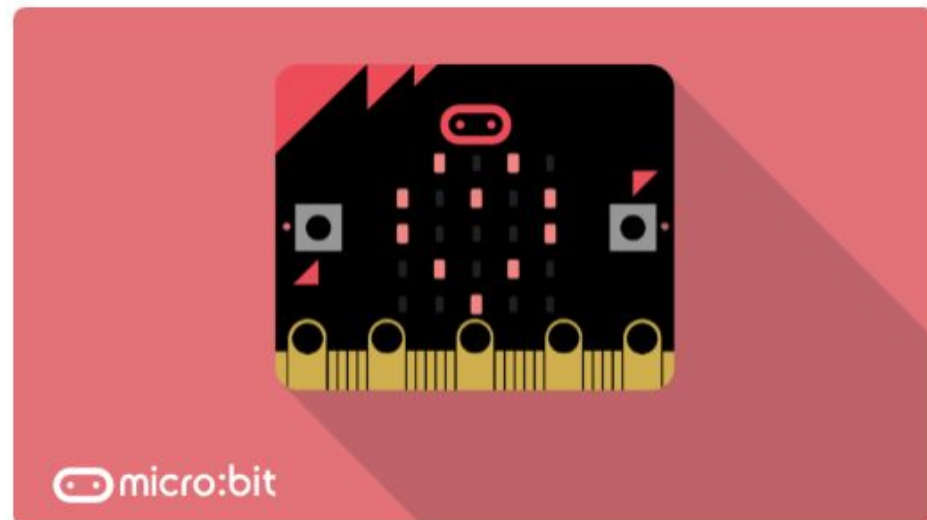
<https://scratch.mit.edu/projects/editor/?tutorial=all>

MakeCode

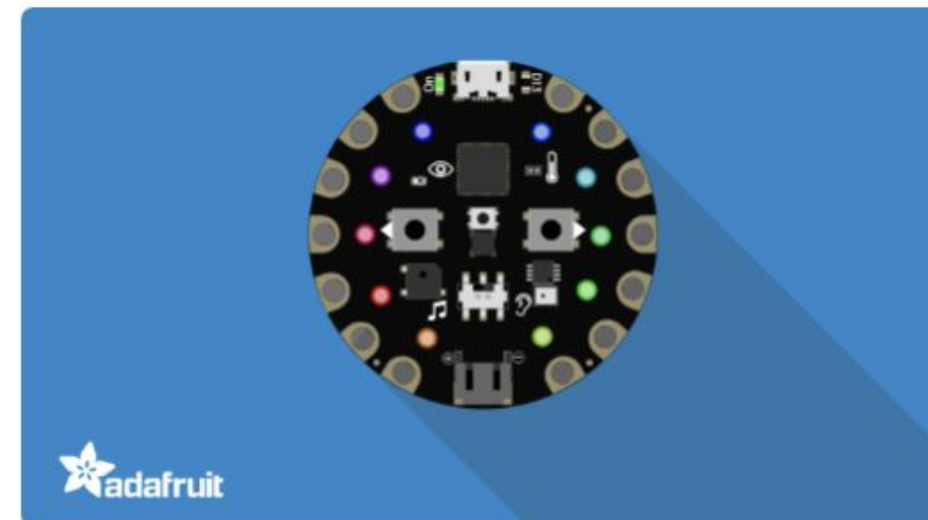
Microsoft MakeCode har som mål å gjøre datavitenskap engasjerende for alle elever med morsomme prosjekter, umiddelbare resultater og både blokk- og tekstredigering for elever på ulike nivåer.

Praktisk utdanning i databehandling

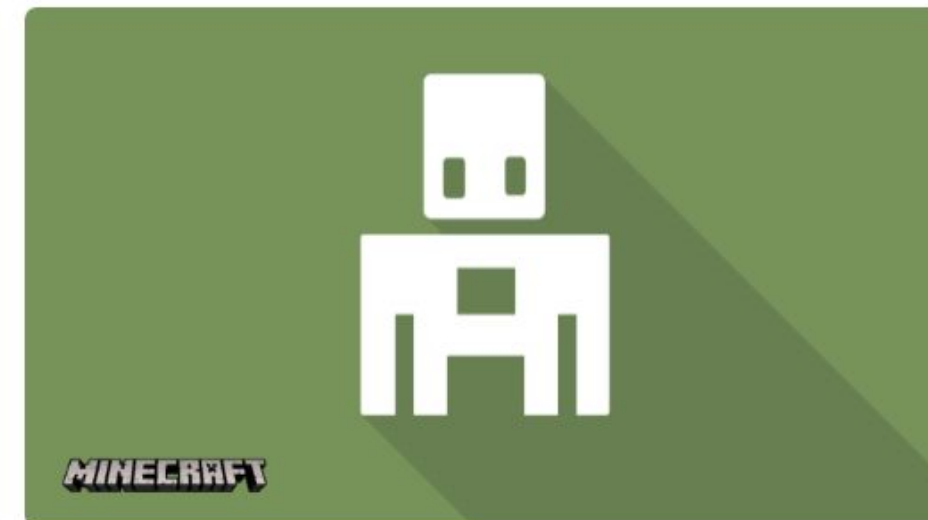
Microsoft MakeCode gjør datavitenskap engasjerende for alle elever med morsomme prosjekter, umiddelbare resultater og både blokk- og tekstredigering for elever på ulike nivåer.



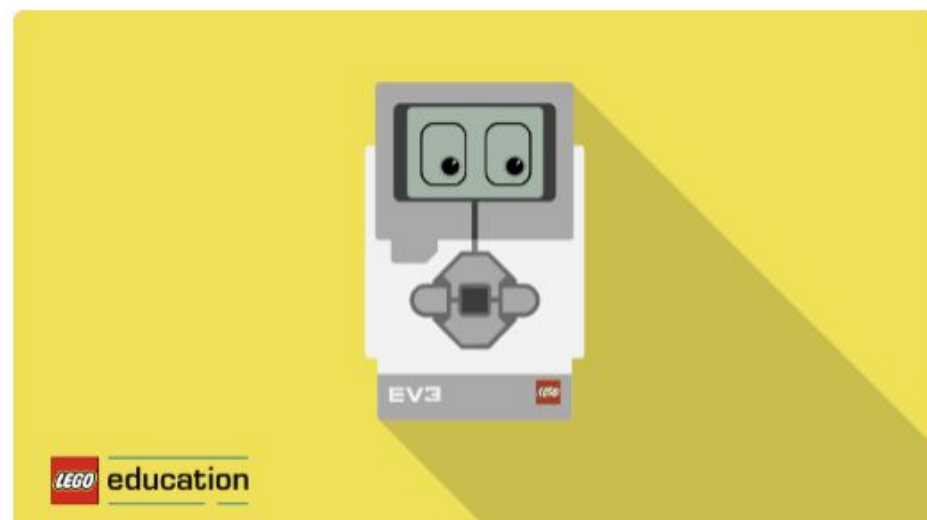
micro:bit
[Begynn å kode med micro:bit >](#)



Circuit Playground Express
[Begynn å kode med Circuit Playground Express >](#)



Minecraft
[Begynn å kode med Minecraft >](#)



LEGO® MINDSTORMS® Education EV3
[Begynn å kode med LEGO MINDSTORMS >](#)

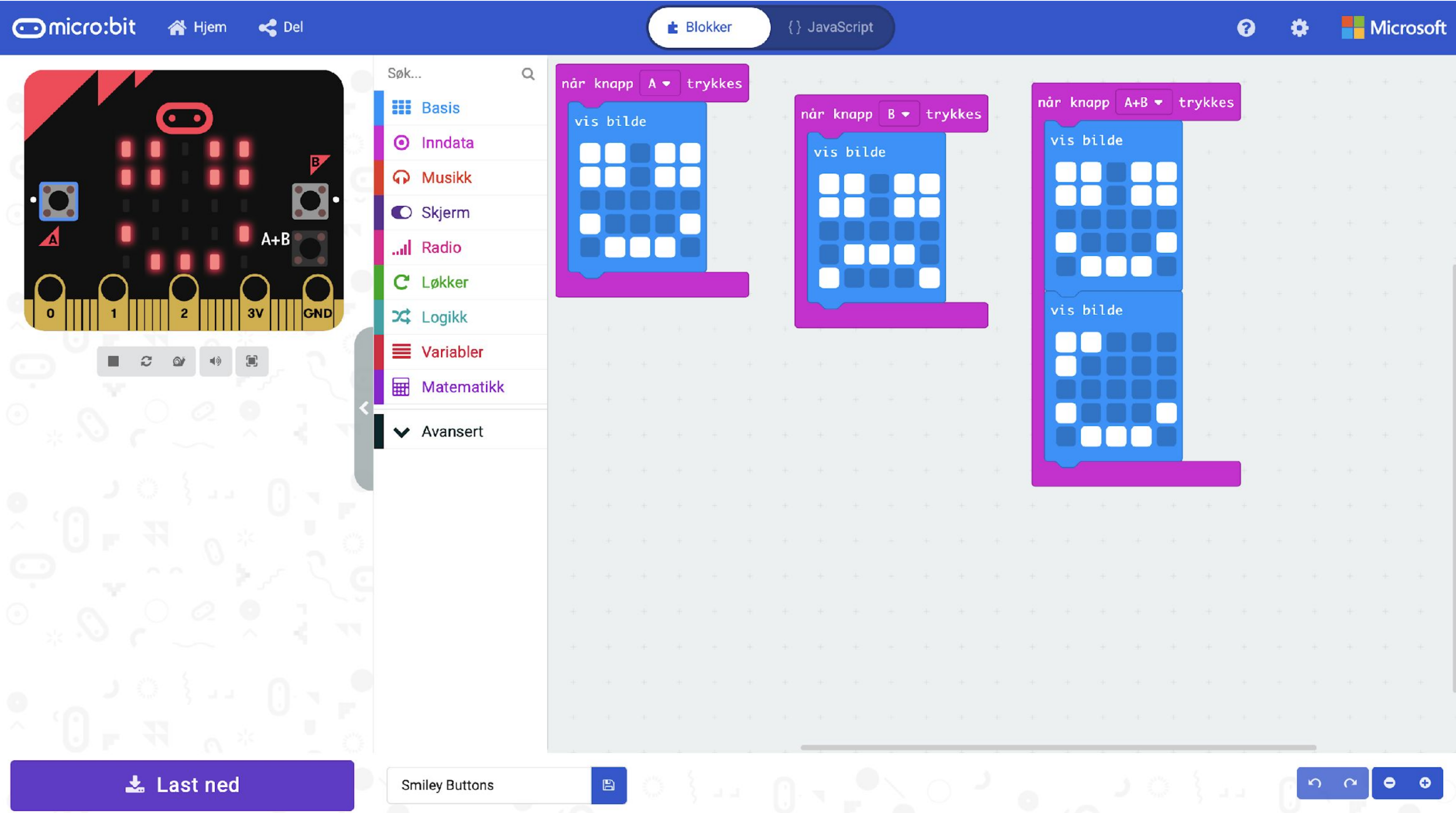


Cue
[Begynn å kode med Cue >](#)



BETA
Arcade
[Begynn å kode med Arcade >](#)

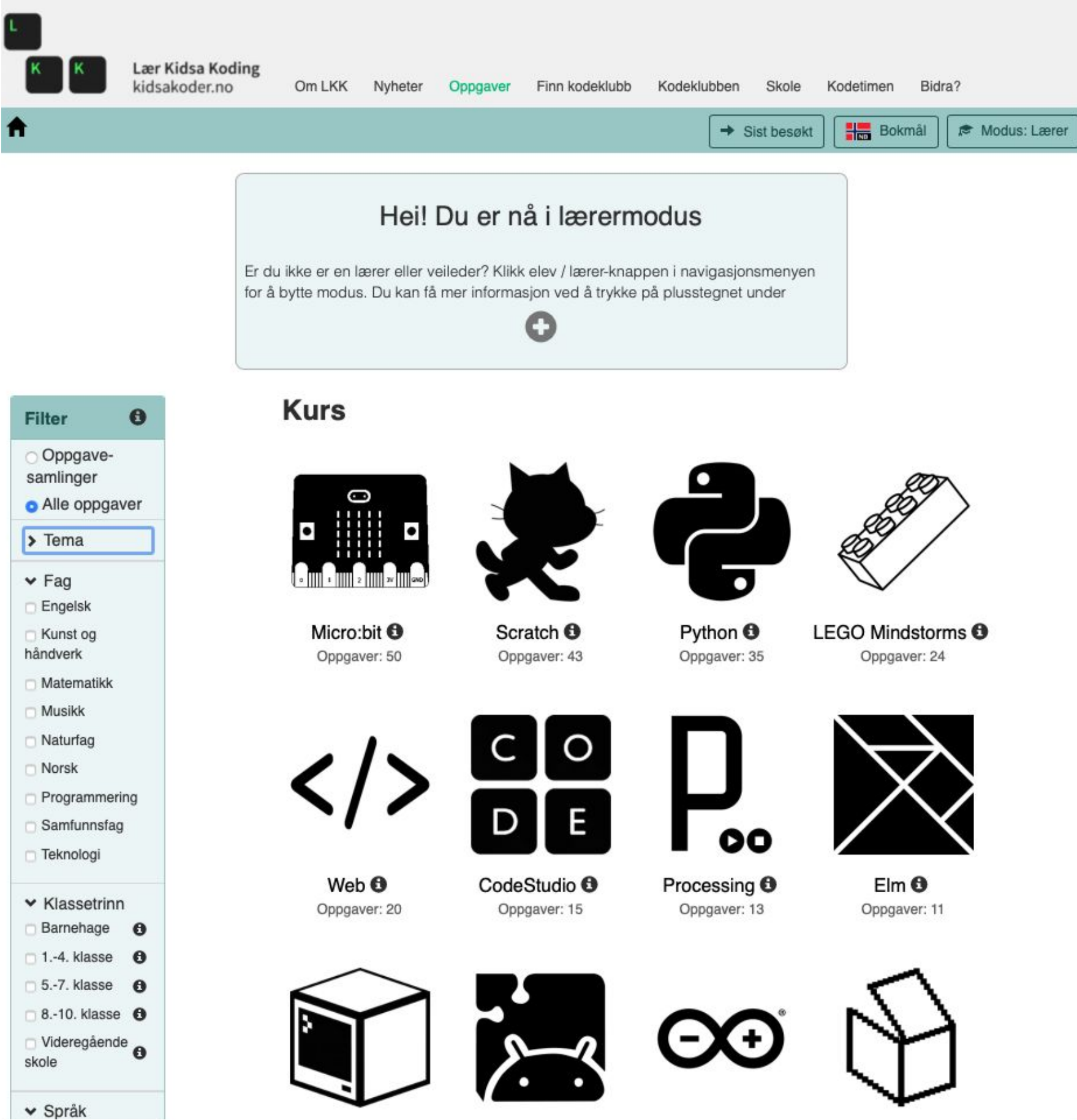
Slik kan det se ut når en er inne i MakeCode og lager et lite program for micro:bit





Lær Kidsa Koding! er en frivillig bevegelse som arbeider for at barn og unge skal lære å forstå og beherske sin egen rolle i det digitale samfunnet.

Her finner man mange opplegg laget på norsk tilpasset fag, klassetrinn og programmeringstype



NB:
lærermodus

Dagens aktiviteter

Nivå 2

Her står du mere fritt til å velge verktøy og oppgaver fra nettressursen. Vi foreslår at du prøver å lage et lite program i Scratch.

[Bit.ly/borkedalnett](https://bit.ly/borkedalnett)

Ressursside for Samling 4



Eksempler på aktiviteter

Eksempler på analog og digital programmering i skolen



Scratch
Lær å kode

Nivå 3

For deg som har mye erfaring med temaet.

Her står du fritt til å jobbe selvstendig med alle ressursene

Praktisk programmerings-oppgave

1. Gå inn på code.org eller scratch.mit.edu
 1. Ta for deg et kurs/veiledning og test ut. Får du laget et lite program?
2. Gå til oppgaver.kidsakoder.no og aktiver lærermodus. Da får du tilgang til oppgaveveiledninger og hvilke læreplanmål oppgavene er knyttet opp mot.
 1. Finn et opplegg som passer deg. Test ut!

Mellomarbeid

1. Velg deg ut et av dine fag og ta for deg et tema der det kan egne seg å lage en aktivitet knyttet mot algoritmisk tenkning og/eller programmering.
2. Gå til nettressursen for samling 4 – gå så inn på [eksempler på aktiviteter](#).
3. Finn et opplegg som passer deg. Test ut med elevene.
4. Hvordan gikk det? Del tanker og refleksjoner med kollegaer - og vi tar det opp på neste samling i november.

Undersøkelse - Hva syntes du om samlingen i dag?

bit.ly/borkekart4





Nivå 3

Oppgave Nivå 3

Del 1:

- Velg deg en målgruppe og et fag (eks 1.-4. og norsk)
- Lag en progresjonsplan for hvordan dine elever skal introduseres til, forstå og bruke algoritmisk tenkning i faget
- Lag en plan for hvilke type verktøy elevene skal introduseres for og til hvilken tid sammen med dine kolleger (type analogt, code.org, scratch, micro bit, python, annet)

Del 2:

- Lag et undervisningsopplegg der elevene skal lage sitt eget program analogt eller digitalt (lekse/vurderingssituasjon) og vise for deg eller klassen.

Alternativ oppgave Nivå 3

Aktivitet: Verktøy/Ressurser Utforsking og Deling

- Utforsk - Gå i grupper
 - Velg et verktøy eller en ressurs dere ønsker å teste ut
 - Prøve ut forskjellige funksjoner, eksperimentere og få en følelse av hvordan verktøyet eller ressursen fungerer
 - Hjelpe hverandre og diskutere dine funn innenfor gruppen

- Del - Presentere det digitale verktøyet/ressursen dere har utforsket for resten av kollegaene
 - Hvilke funksjoner og bruksområder har verktøyet/ressursen?
 - Hvordan kan dette verktøyet/ressursen være nyttig profesjonelt?
 - Hva likte dere spesielt godt med verktøyet eller ressursen, og var det noen utfordringer?
 - Har dere funnet noen ressurser eller tips som kan hjelpe andre som ønsker å bruke dette verktøyet eller ressursen?

Lenke til Padleten der vi deler: https://padlet.com/haakon_sundbo/borkedalén_prog

Mellomarbeid

1. Velg deg ut et av dine fag og ta for deg et tema der det kan egne seg å lage en aktivitet knyttet mot algoritmisk tenkning og/eller programmering.
2. Gå til nettressursen for samling 4 – gå så inn på [eksempler på aktiviteter](#).
3. Finn et opplegg som passer deg. Test ut med elevene.
4. Hvordan gikk det? Del tanker og refleksjoner med kollegaer - og vi tar det opp på neste samling i november.

Undersøkelse - Hva syntes du om samlingen i dag?

bit.ly/borkekart4

