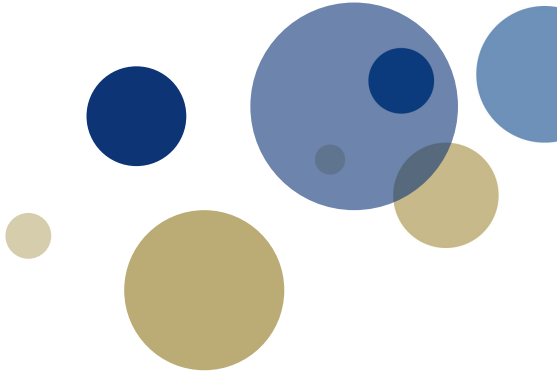
NTNU

Kunnskap for en bedre verden



Programmering og algoritmisk tenking

Matematisk modellering og IKT

Tjerand Silde - 17 Mars, 2021

Tjerand Silde

Doktorgradsstudent i matematikk ved NTNU, med spesialisering i kryptografi og datasikkerhet.

Tidligere:

- Bakgrunn i informatikk og matematikk fra NTNU
- Grunnla Kodeklubben Trondheim i 2013, og Kodeklubben Oslo i 2015
- Prosjektleder i Lær Kidsa Koding 2015 - 2017
- Foreleser i programmering ved HVL 2017 - 2020
- Foreleser i matematikk ved NTNU 2019 - 2020
- Arrangert 12 lærerkonferanser om programmering rundt om i hele landet

LÆR KIDSA KODING

Lær Kidsa Koding
kidsakoder.no

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.

Lær Kidsa Koding! vil hjelpe de unge til å ikke bare bli brukere, men også skapere med teknologien som verktøy.

Kodeklubben er en arena for utforskning, læring og kreativitet, ikke et sted hvor et orakel skal fortelle hvordan alt fungerer.

Matematikktimen er en arena for utforskning, læring og kreativitet, ikke et sted hvor et orakel skal fortelle hvordan alt fungerer.

PROGRAMMERING I SKOLEN

Kjerneelement i matematikkfaget

- Utforsking og problemløysing
- Modellering og anvendingar
- Resonnering og argumentasjon
- Representasjon og kommunikasjon
- Abstraksjon og generalisering

Programmering i matematikkfaget

- 2. trinn: lage og følgje reglar og trinnvise instruksjonar i leik og spel
- 4. trinn: lage algoritmar og uttrykkje dei ved bruk av variablar, vilkår og lykkjer
- 6. trinn: bruke variablar, lykkjer, vilkår og funksjonar i programmering til å utforske geometriske figurar og mønster
- 7. trinn: bruke programmering til å utforske data i tabellar og datasett
- 8. trinn: utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering
- 9. trinn: simulere utfall i tilfeldige forsøk og berekne sannsynet for at noko skal inntreffe, ved å bruke programmering
- 10. trinn: utforske matematiske eigenskapar og samanhengar ved å bruke programmering
- 1T: formulere og løyse problem ved hjelp av algoritmisk tenking, ulike problemløysingsstrategiar, digitale verktøy og programmering

Programmering i matematikkfaget

- 6. trinn: utforske og beskrive symmetri i mønster og utføre kongruensavbildingar med og utan koordinatsystem
- 9. trinn: berekne og vurdere sannsyn i statistikk og spel
- 10. trinn: modellere situasjonar knytte til reelle datasett, presentere resultata og argumentere for at modellane er gyldige
- 10. trinn: utforske samanhengen mellom konstant prosentvis endring, vekstfaktor og eksponentialfunksjonar
- 10. trinn: utforske og samanlikne eigenskapar ved ulike funksjonar ved å bruke digitale verktøy
- 1T: identifisere variable storleikar i ulike situasjonar, setje opp formlar og utforske desse ved hjelp av digitale verktøy

Programmering i andre fag

- K&H, 7. trinn: bruke programmering til å skape interaktivitet og visuelle uttrykk
- Naturfag, 7. trinn: utforske, lage og programmere teknologiske systemer som består av deler som virker sammen
- Naturfag, 10. trinn: utforske, forstå og lage teknologiske systemer som består av en sender og en mottaker
- Naturfag, 10. trinn: bruke programmering til å utforske naturfaglige fenomener

Fordeler og ulemper med programmering i skolen

1. Individuell: Skriv ned 3 fordeler og 3 ulemper

2. Grupper: Presenter egne fordeler/ulemper og bli enige om 3 bidrag fra gruppen i hver kategori

3. Plenum: Presenter gruppens bidrag

Systematisk tenkning & problemløsning

Arbeidsmetodikk og utholdenhet

Mange ulike måter å løse et problem på

Samarbeid

Glede av å lære nye ting

Øker forståelse for språk og matematikk

```
from random import randint

tall1 = randint(2, 12)
tall2 = randint(2, 12)

print('Hva er ' + str(tall1) + ' ganger ' + str(tall2) + '?')
svar = input()

if svar == tall1 * tall2:
    print('Ja, svaret er ' + svar)
else:
    print('Nei, det riktige svaret er ' + str(tall1 * tall2))
```

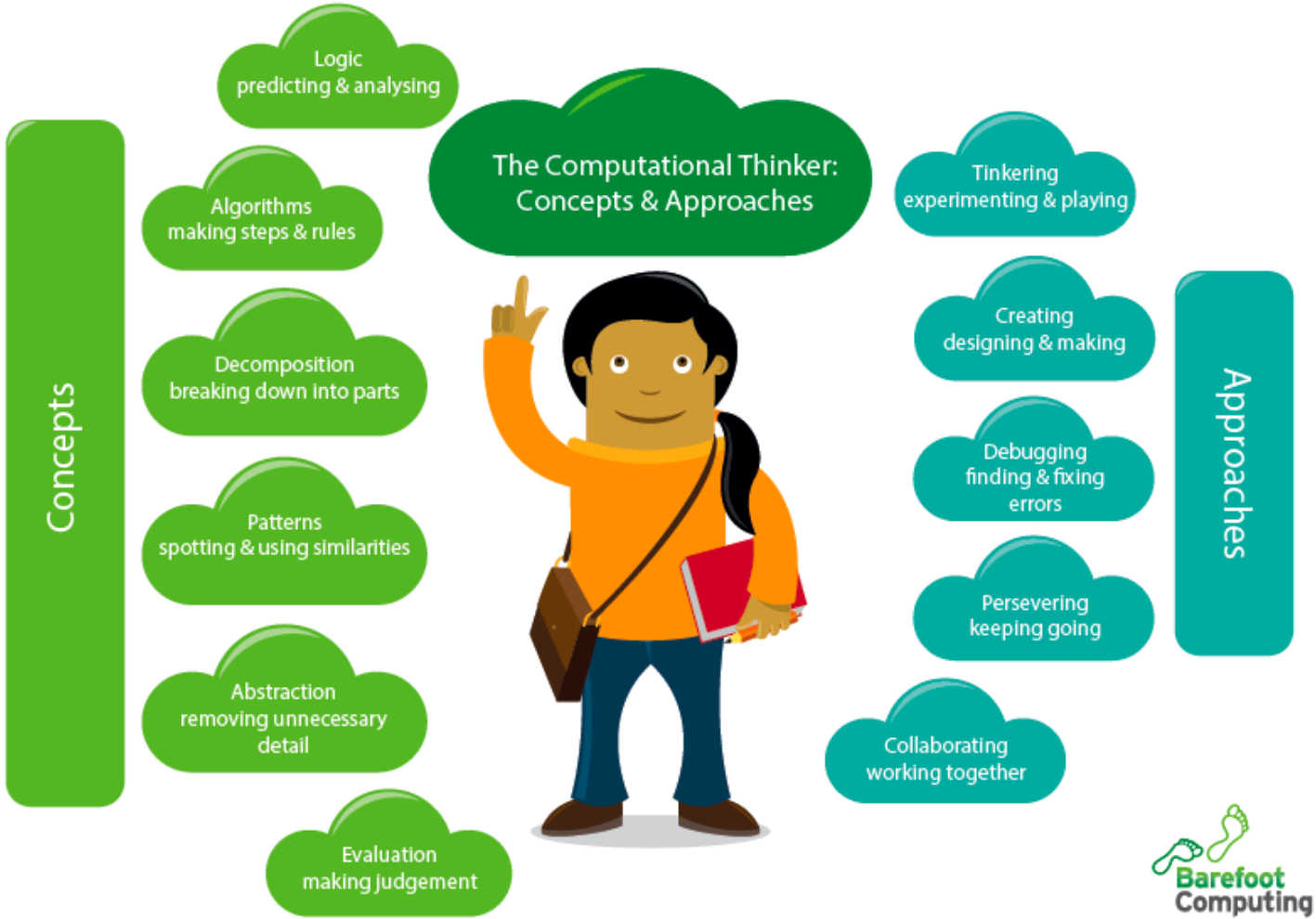
ALGORITMISK TENKING

Hva er algoritmisk tenkning?

1. Individuell: Skriv ned 3 punkter

2. Grupper: Presenter egne punkter til hverandre og bli enige om 3 bidrag fra gruppen

3. Plenum: Presenter gruppens bidrag



Barefoot Computing

Den algoritmiske tenkeren (UDIR)

https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning

Algoritmisk tenkning innebærer

... å bryte ned komplekse problem til mindre, mer håndterlige delproblemer som lar seg løse.

... å organisere og analysere informasjon på en logisk måte og å lage fremgangsmåter for å komme fram til ønsket løsning.

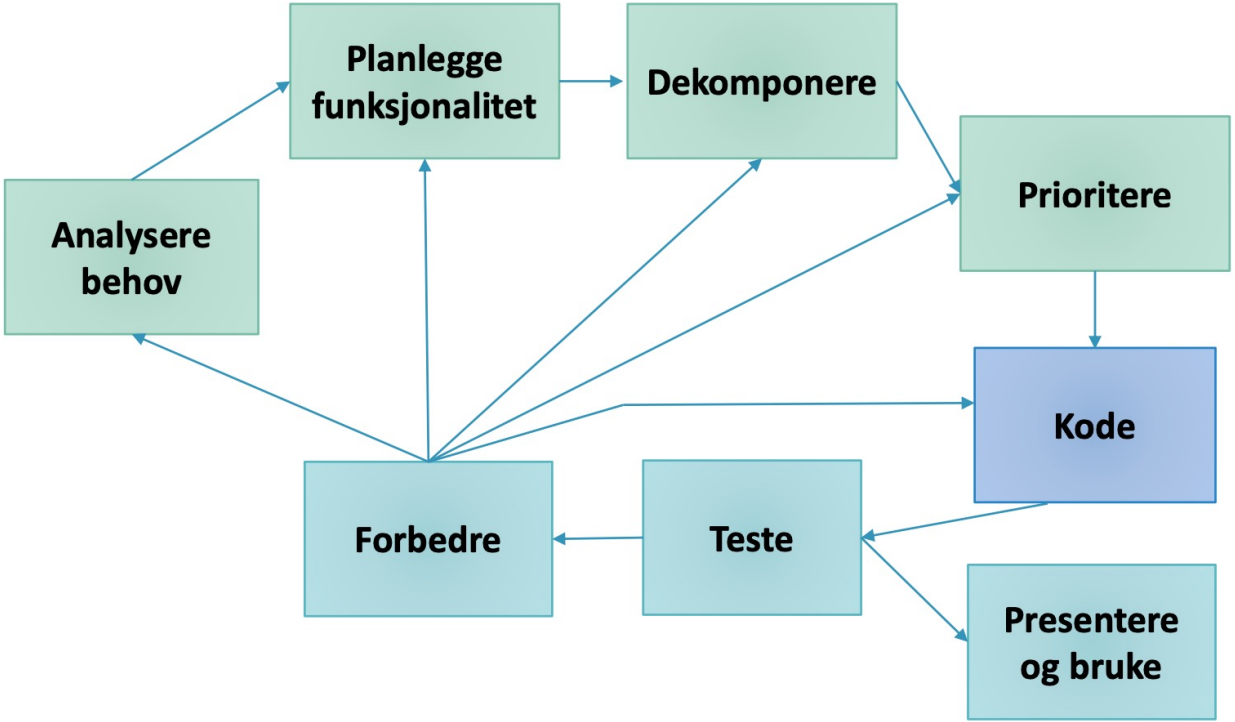
... å lage abstraksjoner og modeller av den virkelige verden ved å fjerne unødvendige detaljer og fokusere på det som er relevant for den aktuelle problemstilling og løsning.

Den algoritmiske tenkeren

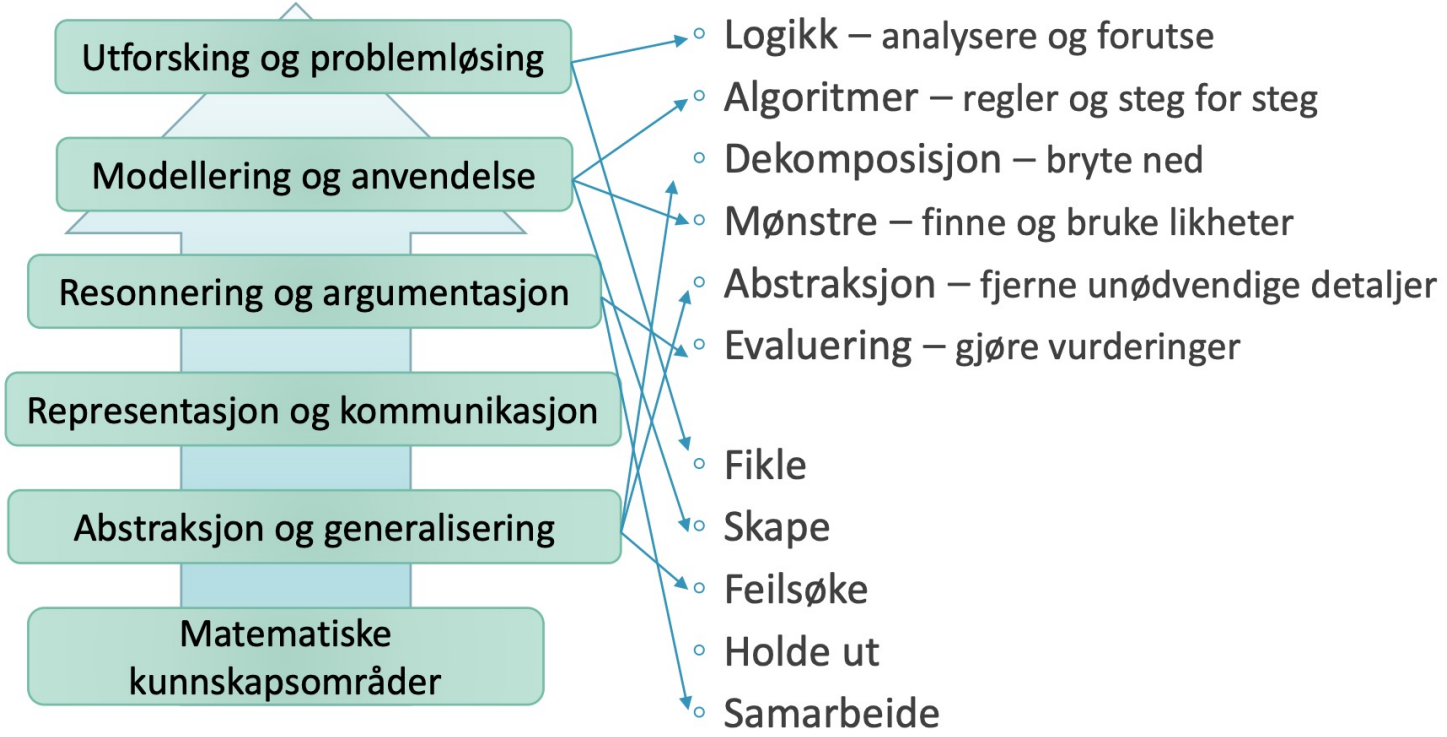
... må være systematisk og analytisk i sitt arbeid, men det er minst like viktig å være skapende, eksperimenterende og åpen for alternative løsninger.

... må ha en nysgjerrig og utforskende tilnærming til å formulere og løse problemer. Å gjøre feil underveis er en viktig del av prosessen, og den algoritmiske tenkeren må ha strategier for å oppdage at noe er feil og rette feilene.

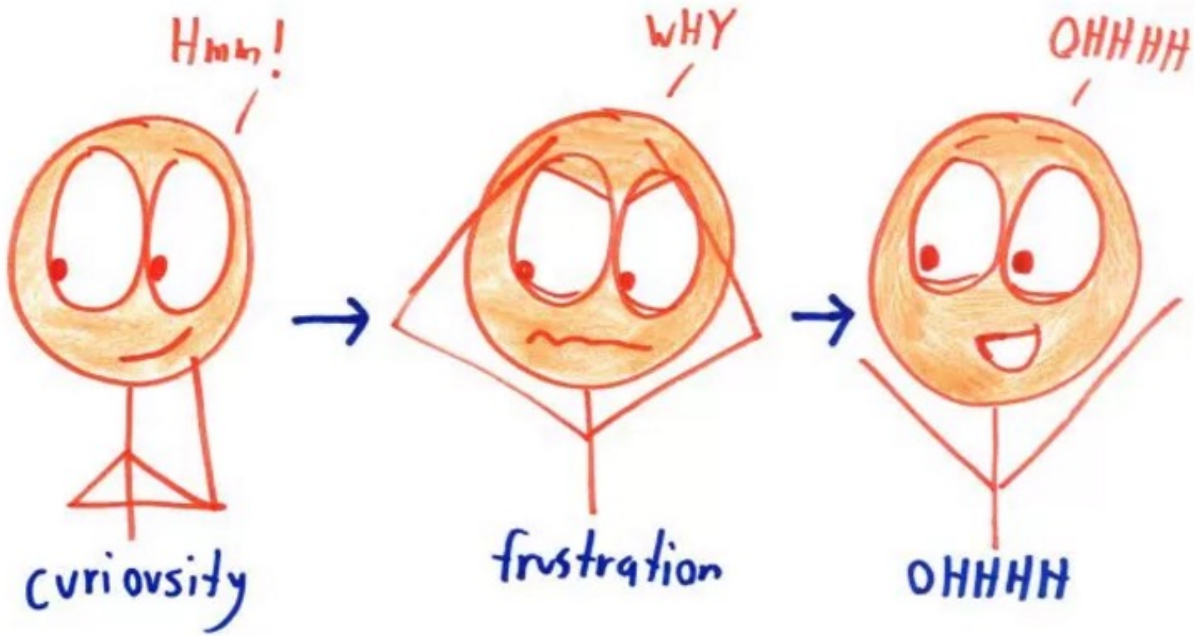
Hva innebærer programmering?

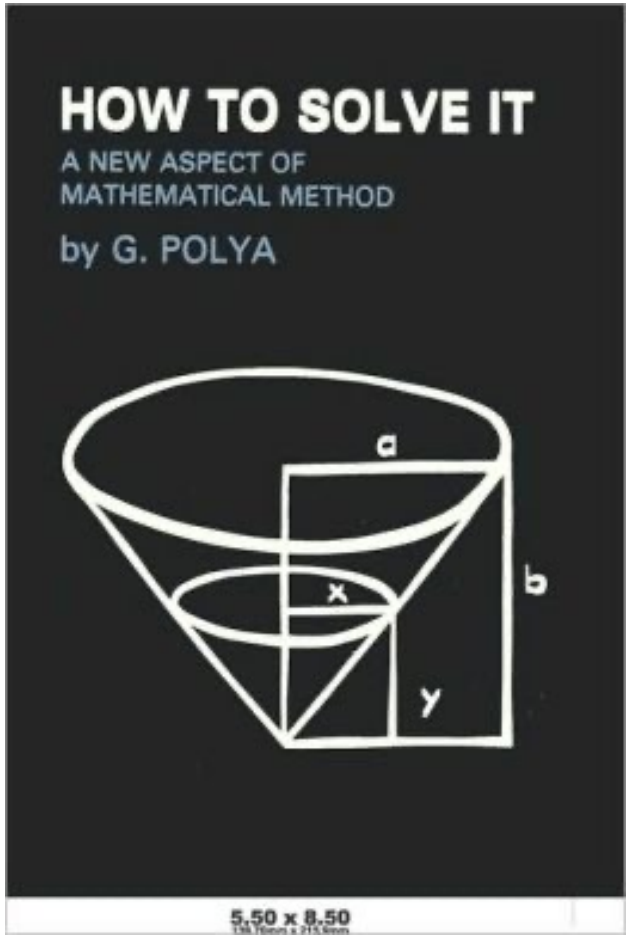


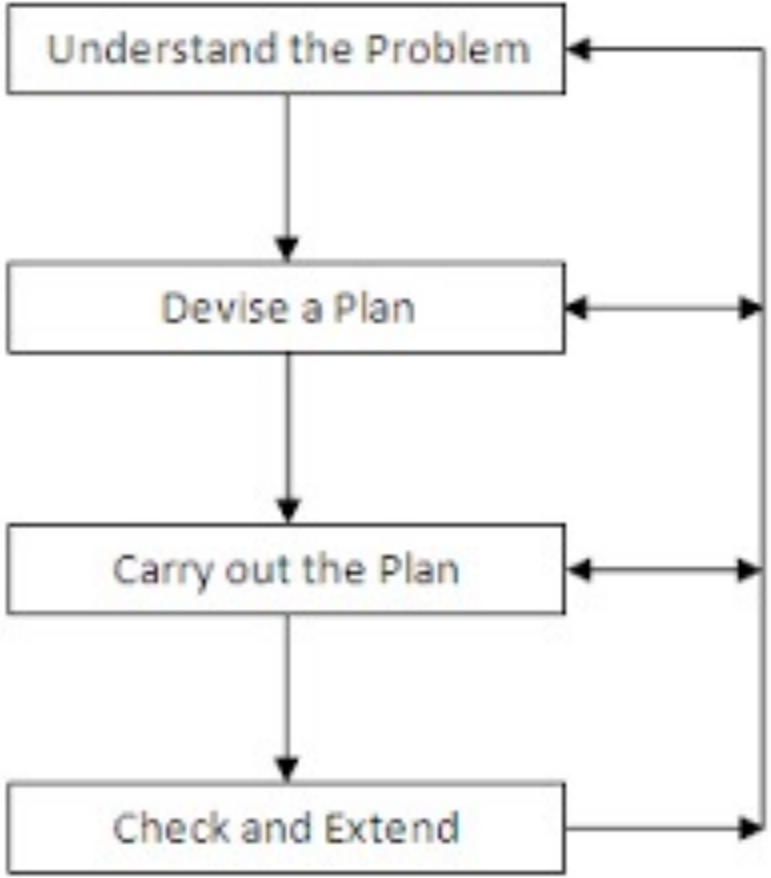
Kjerneelementer i matematikk

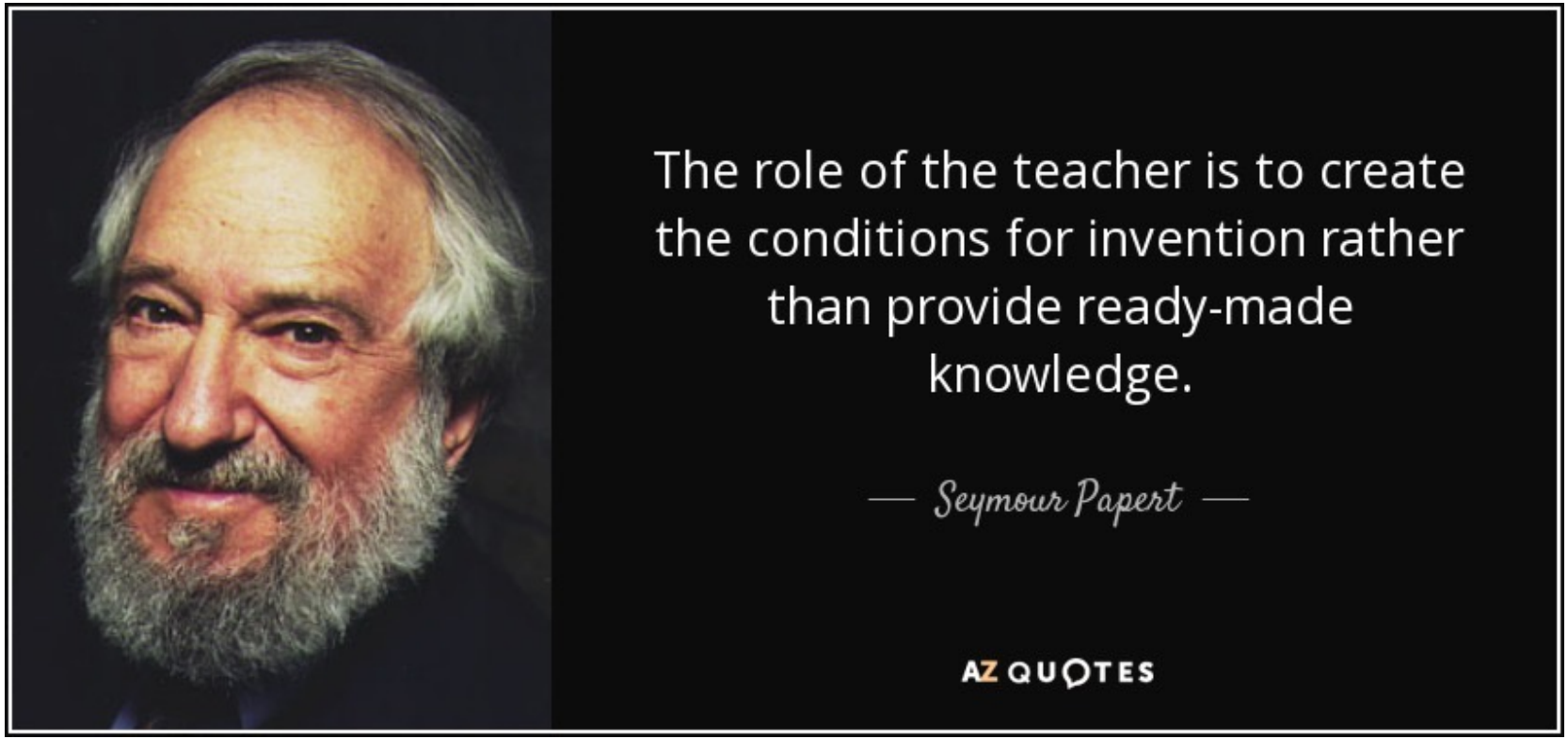


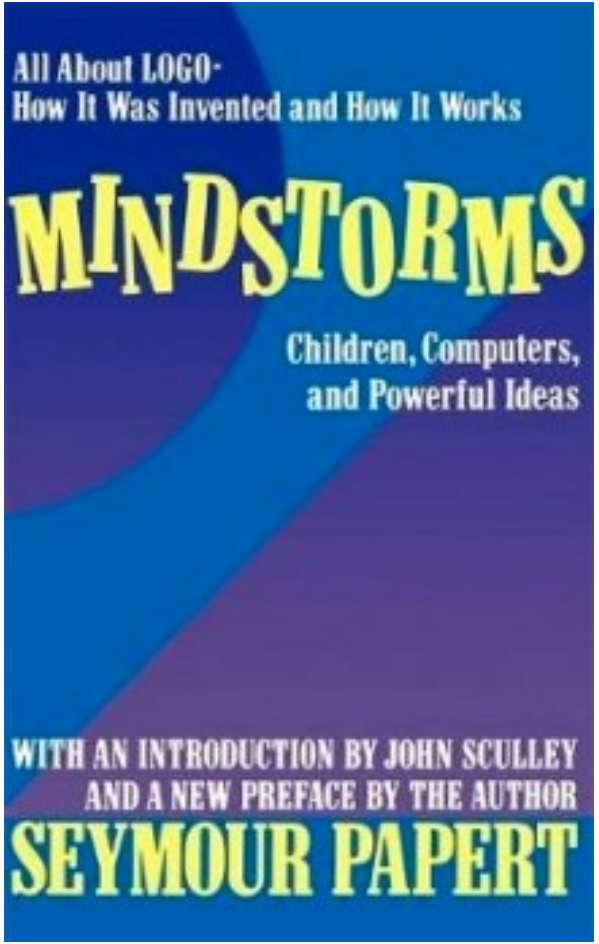
The Mathematics Three-Step



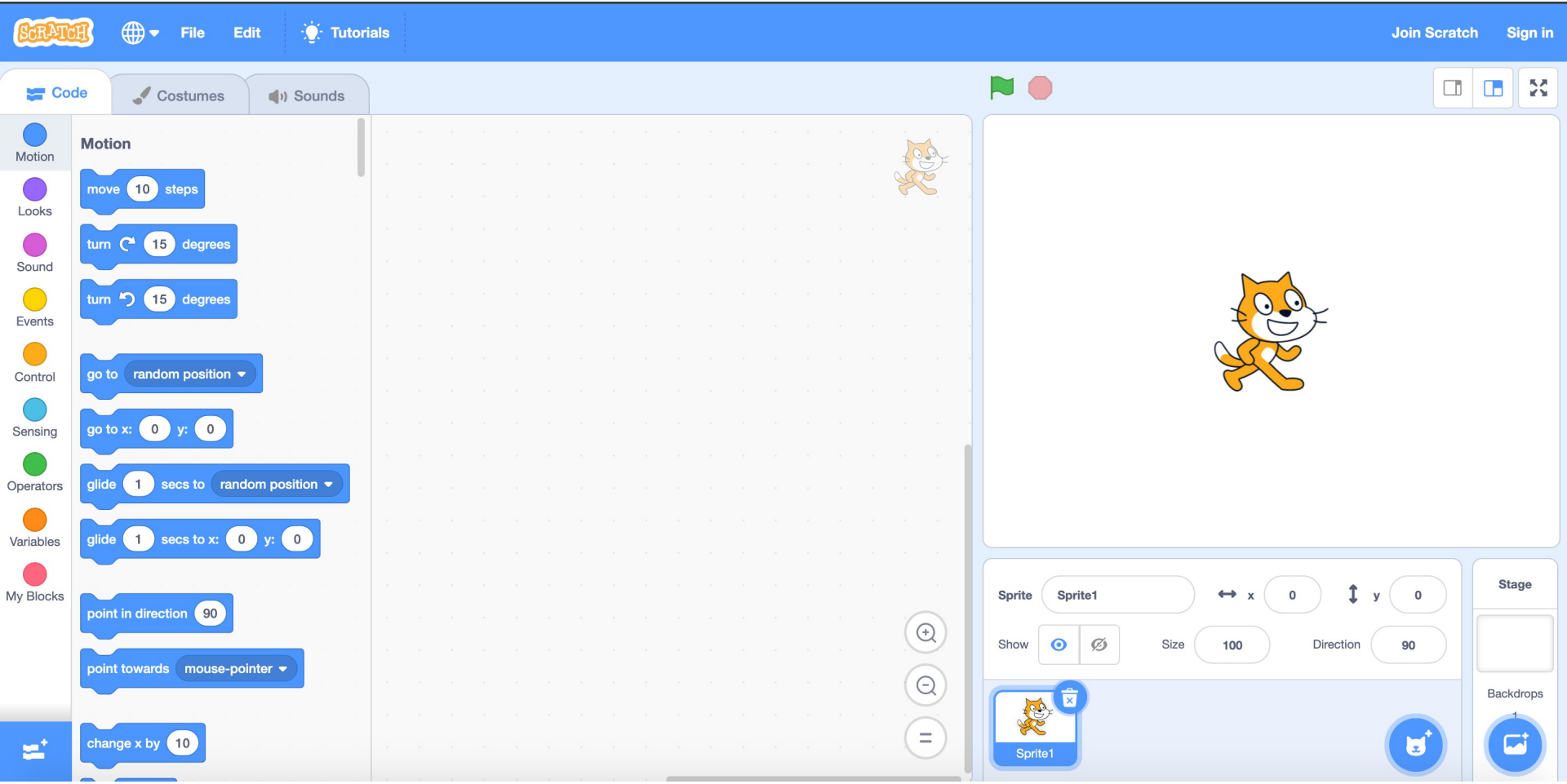

HOW TO SOLVE IT
A NEW ASPECT OF
MATHEMATICAL METHOD
by G. POLYA
a
b
x
y
5.50 x 8.50


Understand the Problem
Devise a Plan
Carry out the Plan
Check and Extend


The role of the teacher is to create the conditions for invention rather than provide ready-made knowledge.
— Seymour Papert —
AZ QUOTES


All About LOGO-
How It Was Invented and How It Works
MINDSTORMS
Children, Computers,
and Powerful Ideas
WITH AN INTRODUCTION BY JOHN SCULLEY
AND A NEW PREFACE BY THE AUTHOR
SEYMOUR PAPERT

EKSEMPLER PÅ KODING


File
Edit
Tutorials
Join Scratch
Sign in
Code
Costumes
Sounds
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
My Blocks
Motion
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 0 y: 0
glide 1 secs to random position
glide 1 secs to x: 0 y: 0
point in direction 90
point towards mouse-pointer
change x by 10
Sprite
Sprite1
x
0
y
0
Show
Size
100
Direction
90
Sprite1
Stage
Backdrops
1

5. trinn

Geometrisk kunst

Matematikk 5. Årstrinn:
lage og programmere algoritmar med bruk av variablar, vilkår og lykkjer

Begreper:
Algoritme
Løkke
Vilkår
Variabel

Verktøy: Scratch

Oppstart: Hva er en algoritme (se begrepsforklaring), og hvor finner vi algoritmer i det daglige?

Bygg algoritmen til høyre trinnvis sammen med elevene.

1) Begynn med "når grønt flagg klikkes" "og "gå ti steg". Trykk på grønt flagg. Hva skjer?
2) Legg på "snu 90 grader" og trykk grønt flagg fire ganger, sånn at dere fullfører et kvadrat.
3) Legg så på en "gjenta for alltid" rundt de to blokkene. Vi ser nå at "katta" fyker rundt i et vilt tempo.
4) For å roe ned katta litt, og gjøre bevegelsen større, legger vi en "gjenta 10 ganger"-løkke utenpå "gå"-blokka.

Tips: For å se bevegelsen til katta enda bedre, kan man importere penn-biblioteket og legge en "penn på"-kloss etter "når grønt flagg klikkes"!

Vi har nå blitt kjent med en "for alltid"-løkke, og en gjenta X ganger-løkke.

Vi skal nå bli bedre kjent med variabler og vilkår.

Vi skal fortsette å tegne kvadrater inntil videre, men nå skal vi telle hvor mange kvadrater vi har tegnet.

1) Da trenger vi en variabel vi kaller "telling". Trykk på "Variabler", "Lag ny", og kall den "telling", trykk OK. Du vil nå se fire oransje klosser.
2) Utvid algoritmen din som til høyre. Vær obs på den nye "gjenta 4 ganger"-blokken!

Nå vil du se at variabelen øker med 1 for hvert fullførte kvadrat.

Dette er i seg selv kanskje ikke så nyttig, men når vi knytter det til neste steg, ser vi at variabler er kjekke å ha.

Nå legger vi på en del praktiske klosser i tillegg, sånn at vi får en fungerende algoritme.

Før vi starter kvadrat-tegningen, ønsker vi at figuren vår skal gå til origo og peke oppover på skjermen.

Vi ønsker også å legge til penn-funksjonalitet.

Til slutt bygger vi på med noen vilkår, som sjekker hva variabelen "telling" er, og forandrer på figuren vår etter verdien på variabelen. **Klossene vi setter inn, ser du til høyre.**

Koden blir litt for lang til å vises enkelt i dette dokumentet, så følg denne lenken: bit.ly/kompmal5 for å se og teste hele algoritmen.

La gjerne elevene se hva sluttproduktet skal bli, og la dem prøve å resonnere seg frem til hvordan koden må se ut!

Videre arbeid:
Vi har nå sett på noe av funksjonaliteten til alle begrepene for 5. trinn. Nå er det på tide å la elevene leke seg litt med koden, og se om de kan lage andre kunstverk!

Kan de for eksempel tegne andre geometriske figurer? Kan de endre på roteringen og skape nye mønstre?

7. trinn

Geometrisk kunst med koordinater

Matematikk 7. årstrinn:
Bruke programmering til å utforske data i tabellar og datasett

Verktøy: Scratch

Begreper:
Algoritme
Variabel
Liste

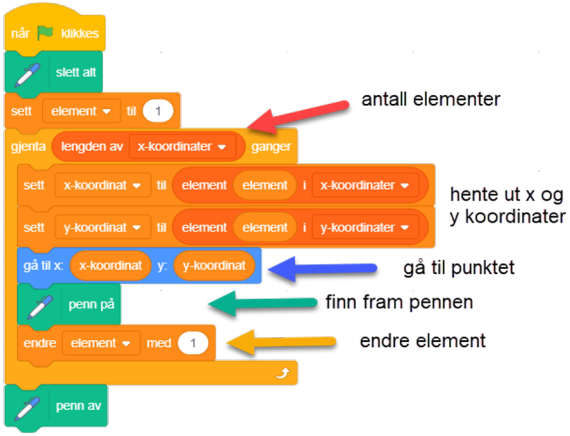
Koordinatene i dette eksempelet skal tilhøre geometriske figurer som f.eks trekanter, firkanter osv, men en kan også bruke koordinater til f.eks bokstaver og tall. Et eksempel er tilgjengelig på https://scratch.mit.edu/projects/356873611
Snakk en del om koordinater i Scratch før dere begynner. Snakk om origo (0,0) og spennet i x- (mellom i -240 og 240) og y-aksene (-180 og 180).

Lage lister i Scratch
Listene settes opp i Scratch under Variabler. I denne oppgaven lager jeg en liste som heter «x-koordinater» og en som heter «y-koordinater»

Elevene fyller inn ønsket x- og y-koordinat i de respektive listene. Oppføring 1 i x-koordinater, må samsvare for oppføring 1 i y-koordinater, osv.

Last inn tegneverktøyet
Denne oppgaven er ikke spesielt spennende eller morsom uten tegneverktøyet i Scratch. Last inn tegneverktøyet fra tilleggene i Scratch og gjør klar for litt tegning basert på x- og y-koordinater.

Snakk med elevene om hvordan Scratch skal kunne finne x- og y-koordinatene når de ligger i hver sin liste. Bygg koden sammen med elevene, og forklar hva som skjer for hvert trinn. Når koden er ferdigstilt, kan du i første omgang fylle inn noen x- og y-koordinater sammen med elevene. Merk: "gjenta (lengden av [x-koordinater]) ganger" vil telle hvor mange oppføringer det er i listen "x-koordinater", og gjenta algoritmen inni løkka så mange ganger. På denne måten kan listene ha så mange eller få oppføringer som vi selv ønsker, uten at koden trenger å endres.



Kjør og se hva som skjer. La elevene utforske hvilke figurer de klarer å tegne ved å bruke koordinater. Hvem klarer å lage den stiligste figuren?

NB! Det vil bli mye feilsøking på koordinater i denne oppgaven.

Eksempelkode: https://bit.ly/kompmal7

10. trinn

Matematikk 10. Trinn:
utforske matematiske eigenskapar og samanhengar ved å bruke programmering

Vi skal lage en algoritme som gjør seg nytte av formelen for Fibonaccis tallrekke, for å finne et hvilket som helst tall i tallrekken.

Til dette trenger vi en liste som tar vare på tallrekka, noen variabler som brukes for å kalkulere neste verdi i tallrekka, og ellers ei løkke, noen operatorer og litt output som gir oss verdien vi etterspør.

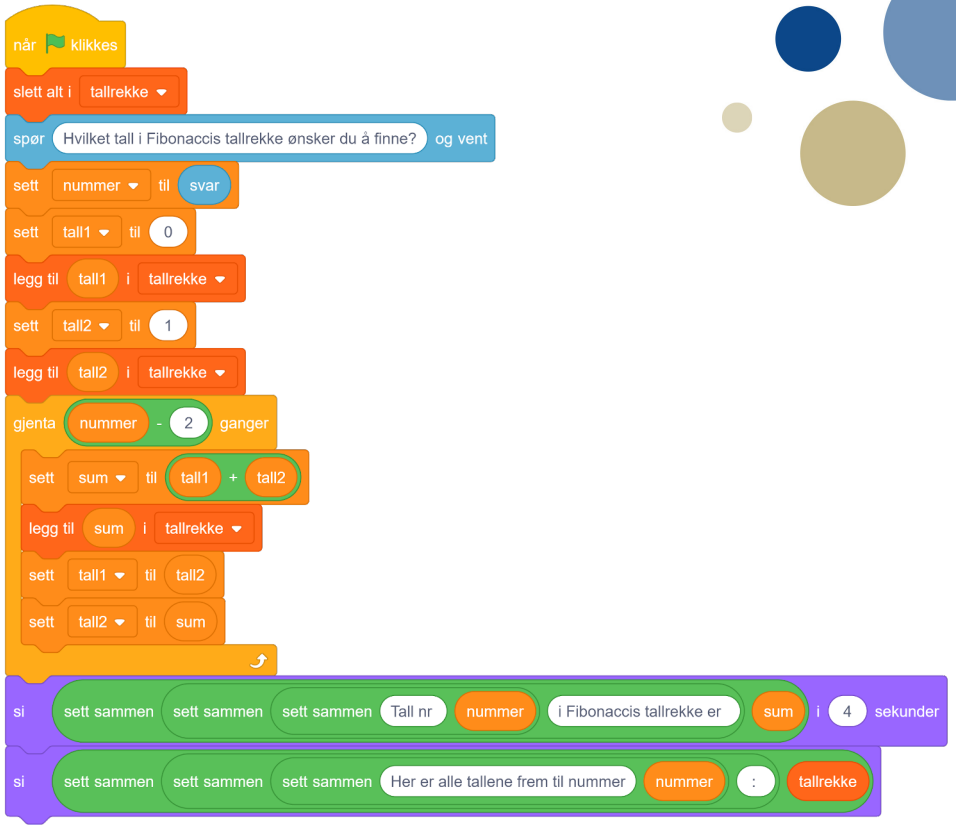
Liste: *tallrekke*: Lagrer alle summeringene i tallrekka for oss.

Variabler: *nummer*: Hvilket nummer i tallrekka ønsker du å finne?
Tall1: Førsteverdi i formelen. Settes til 0 i starten av programmet.
Tall2: Andreverdi i formelen. Settes til 1 i starten av programmet.
Sum: Summen av hver addisjon av tall1 og tall2 i løkka.

Løkke: Gjentas "*nummer*" - 2 ganger, ettersom de to første verdiene i tallrekka settes i starten av programmet

Vi anbefaler at denne oppgaven gjøres som en del av et større opplegg, for eksempel der man ser på praktisk anvendelse av Fibonaccis tallrekke, eller der elevene selv skal utvikle og beskrive tallrekker, som de så skal programmere og presentere.

Her er en lenke til et ferdig prosjekt: https://bit.ly/kompmal10



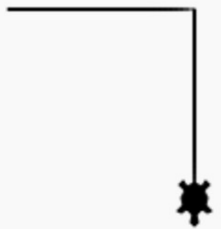
Put Turtle Graphics Anywhere on the Web

Customize the code below and Share!

Remix

main.py

```
1
2  from turtle import *
3
4  shape("turtle")
5  speed(2)
6  forward(100)
7  right(90)
8  forward(100)
9
10
```



https://trinket.io/turtle

Overview of available Turtle and Screen methods

Turtle methods

Turtle motion

Move and draw

```
forward() | fd()
backward() | bk() | back()
right() | rt()
left() | lt()
goto() | setpos() | setposition()
setx()
sety()
setheading() | seth()
home()
circle()
dot()
stamp()
clearstamp()
clearstamps()
undo()
speed()
```

Tell Turtle's state

```
position() | pos()
towards()
xcor()
ycor()
heading()
distance()
```

Pen control

Drawing state

```
pendown() | pd() | down()
penup() | pu() | up()
pensize() | width()
pen()
isdown()
```

Color control

```
color()
pencolor()
fillcolor()
```

Filling

```
filling()
begin_fill()
end_fill()
```

More drawing control

```
reset()
clear()
write()
```

Turtle state

Visibility

```
showturtle() | st()
hideturtle() | ht()
isvisible()
```

https://docs.python.org/3/library/turtle.html

RESSURSER

oppgaver.kidsakoder.no

Ferdige oppgaver tilpasset alle nivå

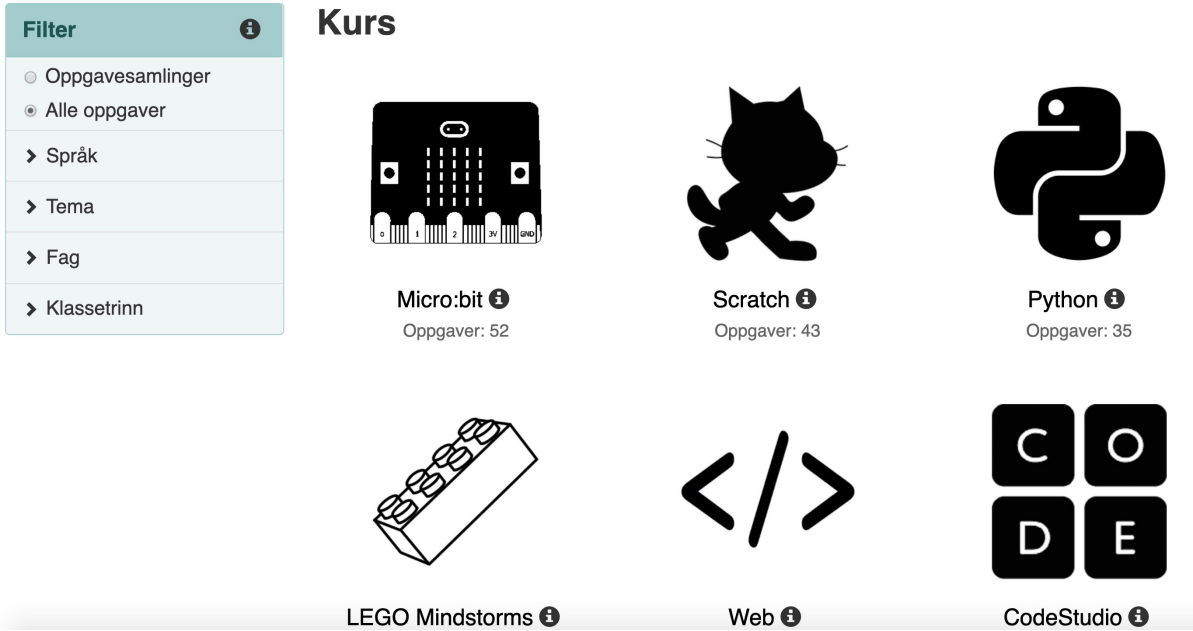
Over 200 veiledninger til 16 «språk»

Lærerveiledninger for gjennomføring

Filtrer på tema, fag og klassetrinn

Finnes på både bokmål og nynorsk

Produsert frivillig, gratis å bruke



Lær Kidsa Koding lærernettverk

Privat gruppe · 4,0 k medlemmer

Om Diskusjon Join Group

https://www.facebook.com/groups/laerernettverk

Lær Kidsa Koding
kidsakoder.no

Kom i gang

- Kodetimen
- super:bit
- Oppgaver i tråd med LK20
- Lær Kidsa Kodings oppgavesider
- Lær Kidsa Koding lærerkonferanser

Undervisningsopplegg

- Småtrinn
- Mellomtrinn
- Ungdomstrinn
- Videregående studiespes
- Videregående yrkesfag

Årsplaner

- Kompetansemål grunnskole
- Barneskole
- Ungdomsskole
- Videregående

https://www.kidsakoder.no/skole

Teknologi og programmering for alle

En faggjennomgang med forslag til endringer i grunnopplæringen- august 2016

For å møte fremtidens behov for kompetanse ser man *algoritmisk tenkning* som det å kunne

- abstrahere
- gjennomgå informasjon systematisk
- lære å lese og å forstå forskjellige representasjonsformer
- modularisere
- resonnere i iterative og parallelle strukturer

https://www.udir.no/tall-og-forskning/finn-forskning/rapporter/teknologi-og-programmering-for-alle

ALLE KURS OG ARRANGEMENTER

Programmeringsdidaktikk

22. MARS KL. 18:00–19:00 STREAMING, NETTBASERT

Høsten 2020 ble programmering en del av læreplanene i realfag. Hvordan kan vi utnytte styrken i programmering slik at vi får til gode diskusjoner i realfagene og oppnå at elevene lærer enda bedre? Og hvordan kan vi best lære elevene programmering?

Beskrivelse

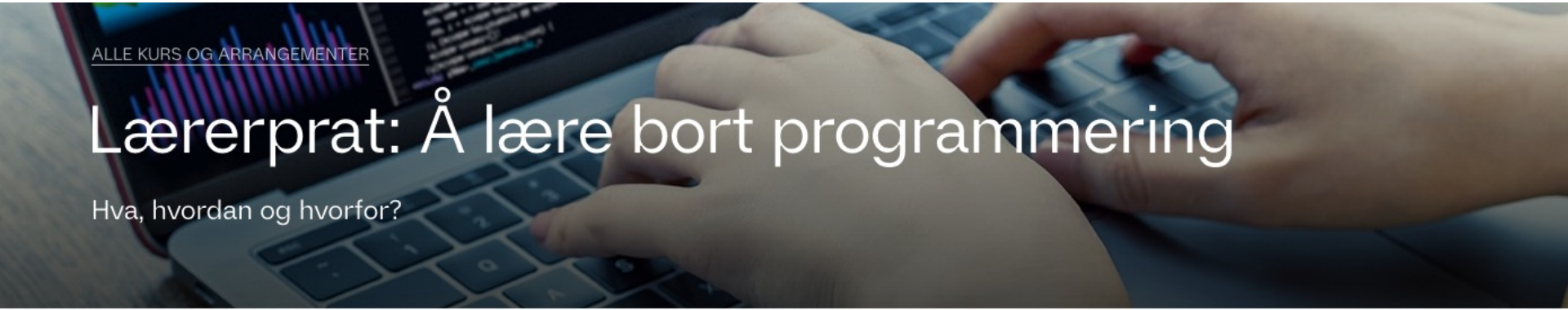
Morten Munthe forsker på programmeringsdidaktikk, ved siden av jobben som mattelærer på Oslo Handelsgym. På dette webinaret vil han snakke om forskningen rundt bruk av programmering i Norge og verden, og om egne funn og erfaringer. Det vil bli praktiske eksempler, og tid til å stille spørsmål.

PRIS

Gratis

Meld meg på

https://www.tekna.no/kurs/programmeringsdidaktikk-41312



ALLE KURS OG ARRANGEMENTER

Lærerprat: Å lære bort programmering

Hva, hvordan og hvorfor?

20. APR. KL. 18:30–19:30 ZOOM, NETTBASERT

Hva er egentlig programmering, og hvordan kan vi bringe det inn i klasserommet på en forståelig måte?

Beskrivelse

Med den nye læreplanen har programmering blitt introdusert som et kjerneelement i skolen. Men hva er egentlig programmering, og hvordan kan vi bringe det inn i klasserommet på en forståelig måte? På dette webinaret skal vi diskutere hva programmering er, hvorfor det er nyttig å lære programmering og forskjellige strategier du kan bruke i programmeringsundervisningen. Innholdet er basert både på forskning og Kodeskolens lange erfaring med å holde programmeringskurs for elever og lærere.

PRIS

Gratis

Meld meg på

https://www.tekna.no/kurs/larerprat-a-lare-bort-programmering-41473

UiT

NORGES
ARKTISKE
UNIVERSITET

Institutt for lærerutdanning og pedagogikk

Koding som digital grublis

En kvalitativ studie om hvordan elevenes læringsstrategier påvirkes gjennom programmering.

—

Susanne Iversen

Masteroppgave i lærerutdanning for 1.-7. trinn mai 2015

https://munin.uit.no/handle/10037/8089

Institutt for lærerutdanning og pedagogikk

Programmering + matematikk = sant?

En casestudie om overføringsverdien mellom programmering valgfag og matematikkfaget

—

Andrej Verstad

https://munin.uit.no/handle/10037/11601

Programming in School

An insight to the Norwegian programming pilot and the inclusion/exclusion of girls in computer programming education

Fay Pedersen Tveranger

http://bora.uib.no/handle/1956/16932

Increasing IT Interest Among Girls in Secondary School: Lectures and Workshops

Sarah Serussi

https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2564789

Hedda Louise Lang-Ree

"Vi må tenke og ikke bare tegne"

En kvalitativ studie om bruk av programmering som verktøy i arbeid med matematikk

Masteroppgave i matematikkdidaktikk 5-10

Veileder: Trygve Solstad

Trondheim, juni 2016

https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2396148

Digital kompetanse

En studie om begrepets fremtid i skolen

Lars Finnsønn Klingenberg

https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2450802

MASTEROPPGAVE

«Ok, da prøver vi …»

Elevers matematiske kompetanse gjennom programmering

Tonje Lindberg

https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/2606144

MASTEROPPGAVE

Hvilke ulike praksiser eksisterer for faget programmering i ungdomsskolen, og hvilke innholdskomponenter legger lærerne vekt på i utformingen av faget?

Ellen Marje Thorland

https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/2586400

MASTEROPPGAVE

Programmering og problemløsning i småskolen

Christina Hemnes

https://hvlopen.brage.unit.no/hvlopen-xmlui/handle/11250/2602950

Takk! Spørsmål?

tjerandsilde.no
tjerand.silde@ntnu.no