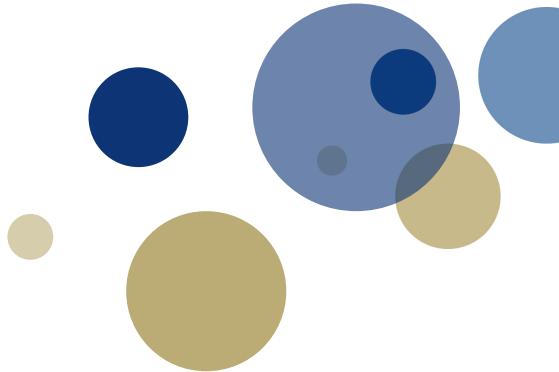




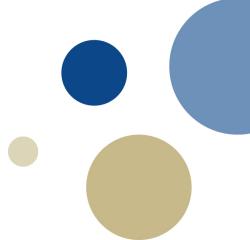
Kunnskap for en bedre verden



Algoritmisk tenkning

Tjerand Silde, 31 Januar 2020

Hva er algoritmisk tenkning?



1. Individuell: Skrive ned 3 punkter
2. Grupper: Presenter egne punkter til hverandre og bli enige om 3 bidrag fra gruppen
3. Plenum: Presenter gruppens bidrag

Concepts

- Logic predicting & analysing
- Algorithms making steps & rules
- Decomposition breaking down into parts
- Patterns spotting & using similarities
- Abstraction removing unnecessary detail
- Evaluation making judgement

The Computational Thinker: Concepts & Approaches



- Tinkering experimenting & playing
- Creating designing & making
- Debugging finding & fixing errors
- Persevering keeping going
- Collaborating working together

Den algoritmiske tenkeren (UDIR)



Algoritmisk tenkning innebærer

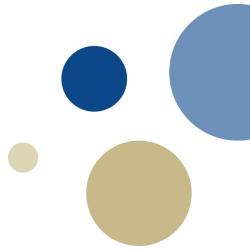


... å bryte ned komplekse problem til mindre, mer håndterlige delproblemer som lar seg løse.

... 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.

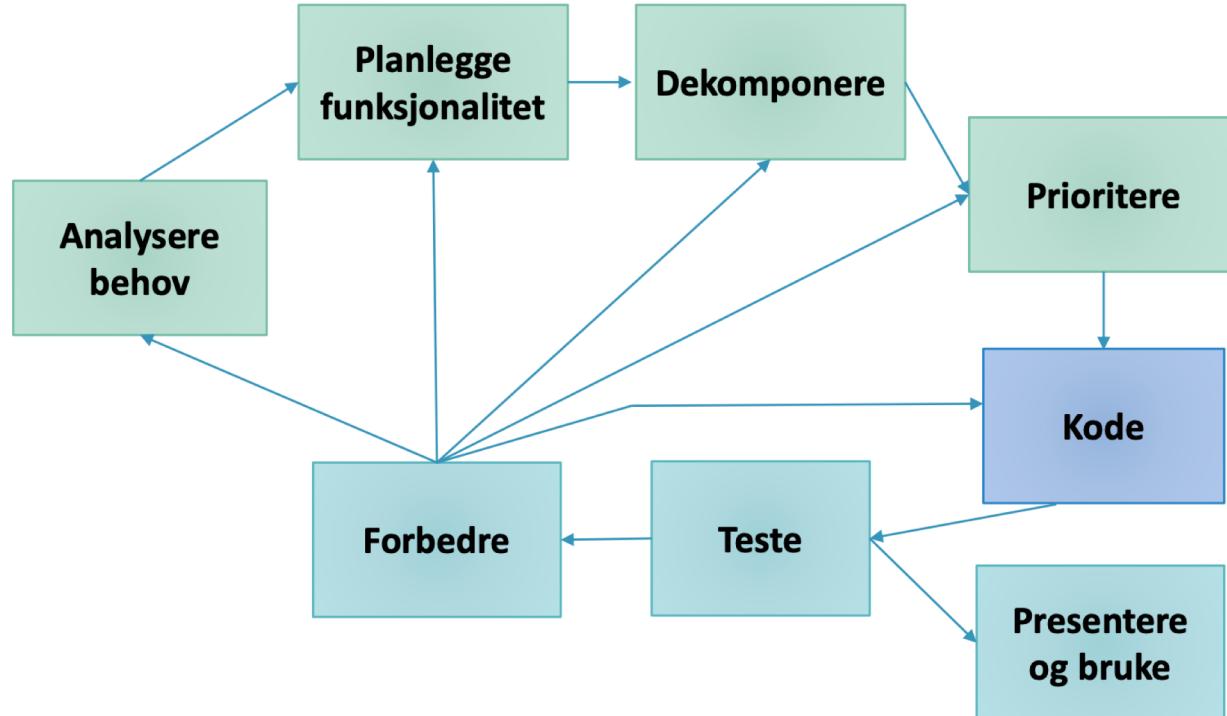
... må ha en nysgjerrig og utforskende tilnærming til å formulere og løse problemer. Å gjøre feil underveis er en viktig del av prosessen, og den algoritmiske tenkeren må ha strategier for å oppdage at noe er feil og rette feilene.

Programmering i matematikk

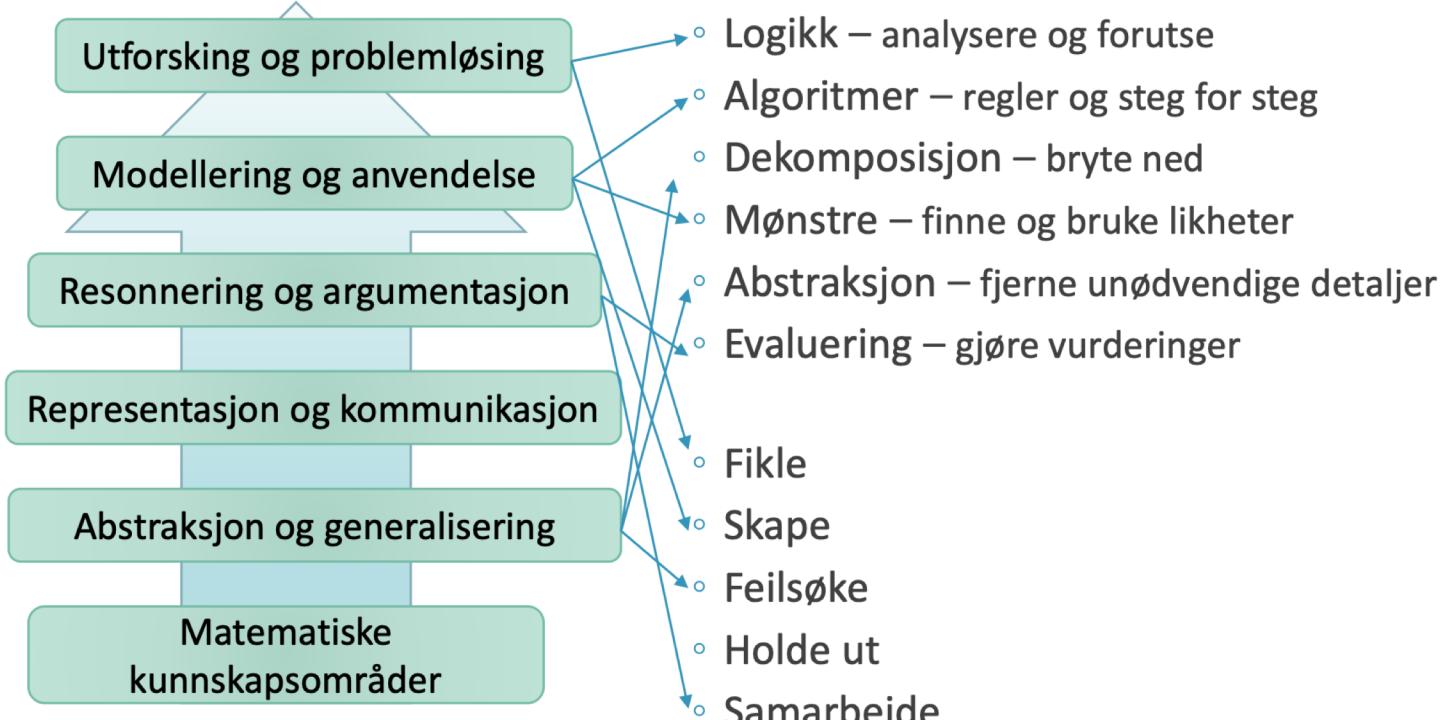


- lage og følgje reglar og trinnvise instruksjonar i leik og spel (knytt til koordinatsystemet)
2. trinn (3. trinn)
- lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse
5. trinn
- utforske korleis algoritmar kan skapast, testast og forbetrast ved hjelp av programmering
8. trinn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

Hva innebærer programmering?



Kjerneelementer i matematikk

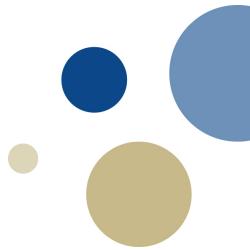


Teknologi og programmering for alle

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



KODING I SKOLEN

Fordeler og ulemper med programmering i skolen

- 
1. Individuell: Skrive 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

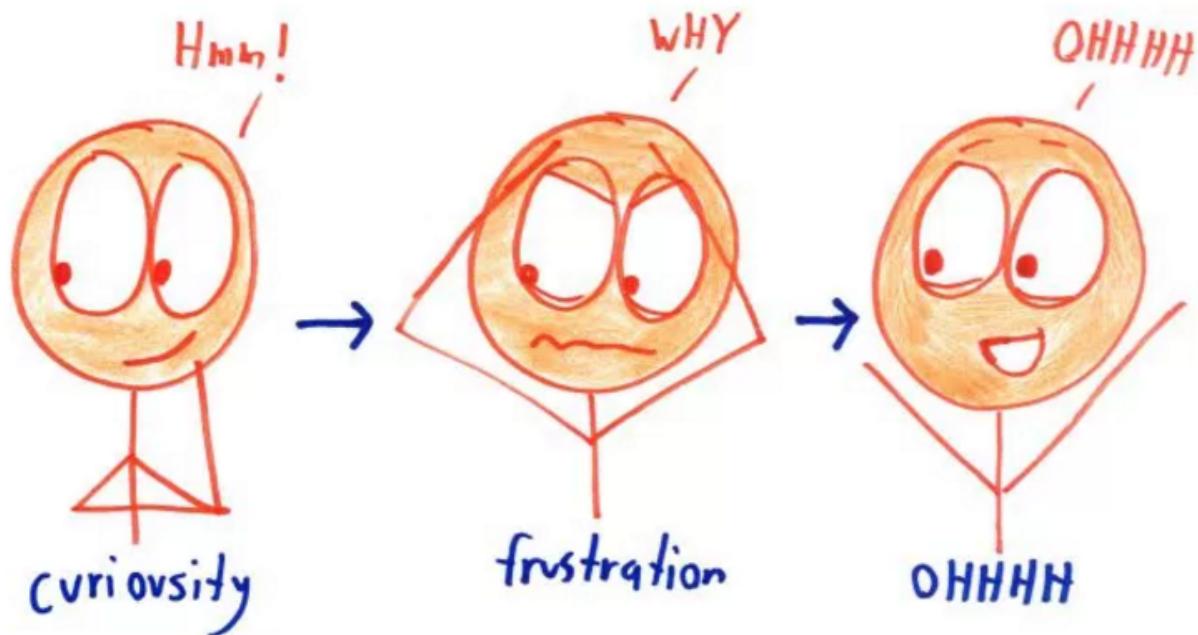
```
from random import randint

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))
```

The Mathematics Three-Step



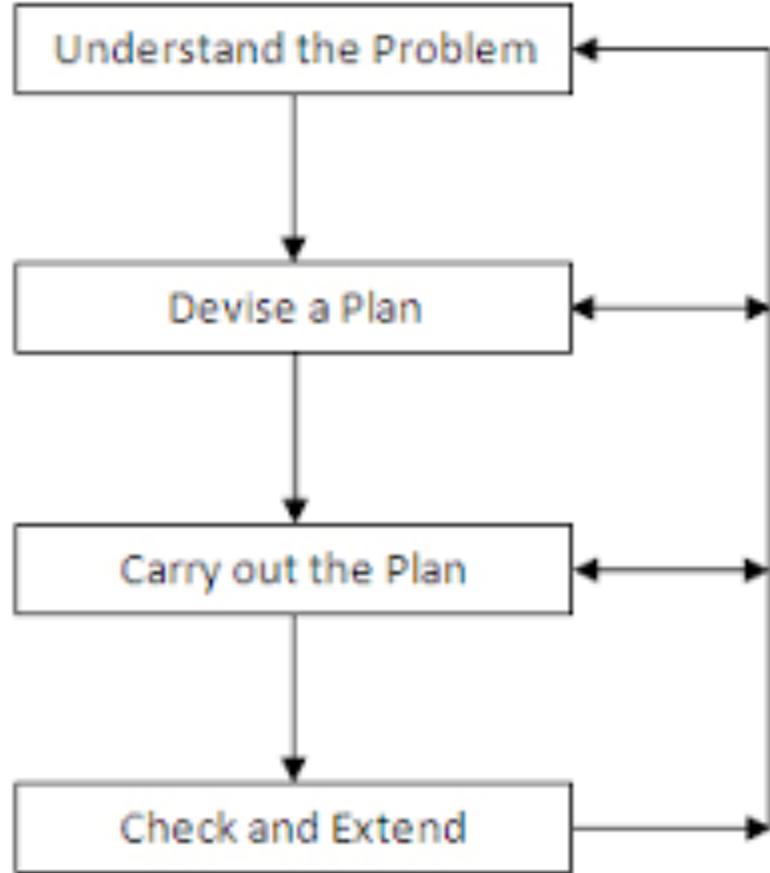
HOW TO SOLVE IT

A NEW ASPECT OF
MATHEMATICAL METHOD

by G. POLYA



5.50 x 8.50
139.7mm x 215.9mm

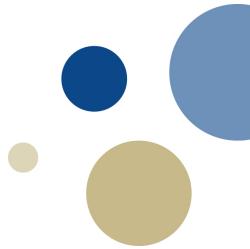




The role of the teacher is to create
the conditions for invention rather
than provide ready-made
knowledge.

— *Seymour Papert* —

AZ QUOTES



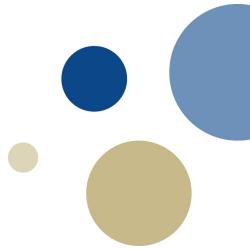
All About LOGO-
How It Was Invented and How It Works

MINDSTORMS

Children, Computers,
and Powerful Ideas

WITH AN INTRODUCTION BY JOHN SCULLEY
AND A NEW PREFACE BY THE AUTHOR

SEYMOUR PAPERT



EKSEMPLER PÅ KODING

Code

Costumes

Sounds



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
Size 100
Direction 90



Stage



Backdrops



repl.it/languages/python_turtle

TMA4115 Matemati... Security News: Cy... Technology - The... Zack Whittaker | Te... Infosider for ansatt... PCP#3 - Linearity... Papers

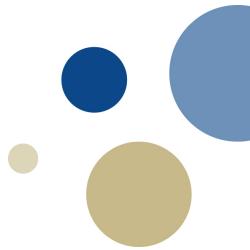
save run ► share + new repl talk Sign up

main.py saved

1 Not sure what to do? Run some examples (start typing to dismiss)

result console

?



MASTEROPPGAVER



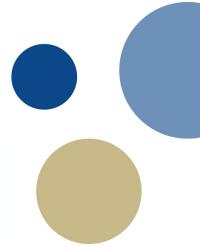
Institutt for lærerutdanning og pedagogikk

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



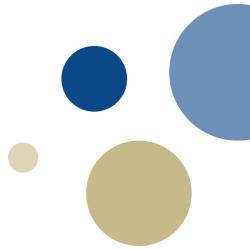
Institutt for lærerutdanning og pedagogikk

Programmering + matematikk = sant?

En casestudie om overføringsverdien mellom programmering valgfag og matematikkfaget

—

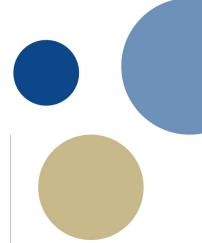
Andrej Verstad



Programming in School

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

Fay Pedersen Tveranger



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

Sarah Serussi

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



Digital kompetanse

En studie om begrepets fremtid i skolen

Lars Finnsønn Klingenbergs

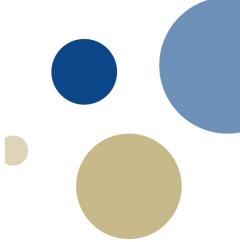
MASTEROPPGAVE



«*Ok, da prøver vi ...»*

Elevers matematiske kompetanse gjennom
programmering

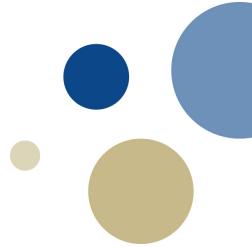
Tonje Lindberg



MASTEROPPGAVE

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

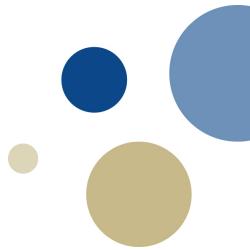
Ellen Marje Thorland



MASTEROPPGAVE

Programmering og problemløsning i
småskolen

Christina Hemnes



LÆR KIDSA KODING!



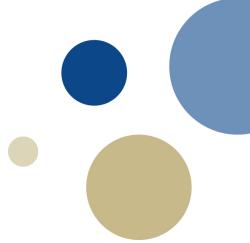
Lær Kidsa Koding
kidsakoder.no

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.



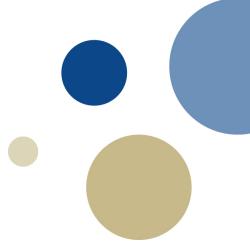
“



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

”

“



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

”

oppgaver.kidsakoder.no

Ferdige oppgaver tilpasset alle nivå

Over 200 veiledninger til 16 «språk»

Lærerveileddninger for gjennomføring

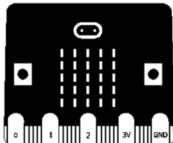
Filtrer på tema, fag og klassetrinn

Finnes på både bokmål og nynorsk

Produsert frivillig, gratis å bruke

Filter	i
<input type="radio"/> Oppgavesamlinger	
<input checked="" type="radio"/> Alle oppgaver	
> Språk	
> Tema	
> Fag	
> Klassetrinn	

Kurs



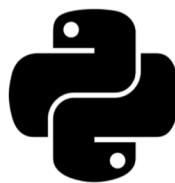
Micro:bit i

Oppgaver: 52



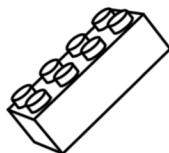
Scratch i

Oppgaver: 43



Python i

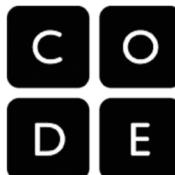
Oppgaver: 35



LEGO Mindstorms i



Web i



CodeStudio i

Lærerkonferanse i programmering



- Arrangeres i Trondheim mandag 23. mars
- Foredrag bl.a. om
 - «Reformasjon 2.0» ved Carl Andreas Myrland, Lær Kidsa Koding
 - «Programmering og fagfornyelsen» ved Kristine Sevik, UDIR
- Workshops:
 - Scratch, super:bit, Sphero, Adafruit, Tiles, KreTek, og Python
- Pris: kr 500,- inkl. lunsj.
- Nettside: kidsakoder.no/event/trondheim2020.



Takk! Spørsmål?

tjerandsilde.no/talks

tjerand.silde@ntnu.no