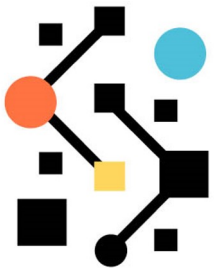


TEKFAG

Læreruddannelsens
udviklingslaboratorier



Udviklingslaboratorie 2

Computationel tænkning i tekster, tekster i computationel tænkning

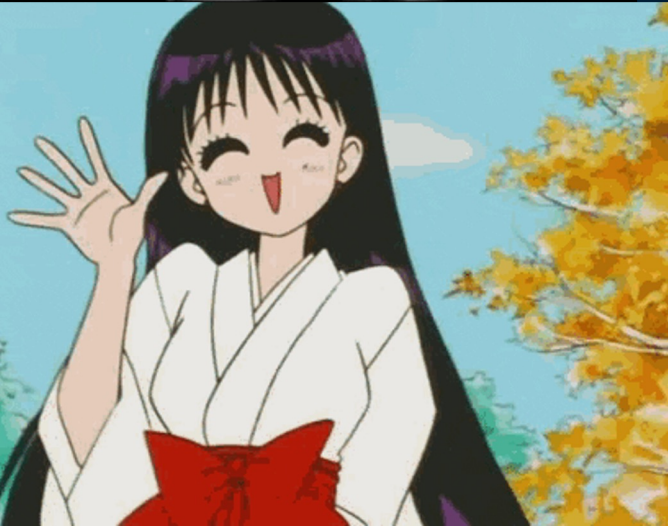


VIA University
College





Welcome



Dagens program - med ca.-tider...

09:00-09:05	Velkommen
09:05-09:15	TEKFAST-model - Hvor står vi i dag?
09:15-09:40	Mød en tænkning - algoritmer
09:40-09:50	Pause
09:50-10:30	Computational tænkning møder danskfaget og rammesætning
10:30-10:45	Pause
10:45-11:45	Undersøgelse 1: Flowcharts og fiktion
11:45-12:45	Undersøgelse 2: Tekster i twine
12:45-13:30	Frokost
13:30-14:30	Videndeling og opsamling



Kompetenceområder - indhold og samspil

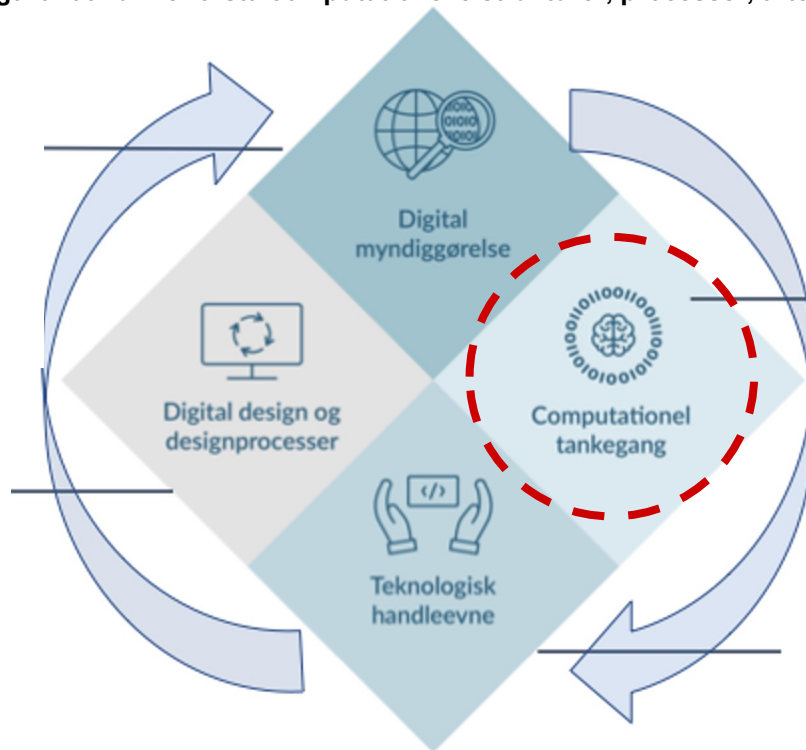
Omsætte processer fra omverdenen til algoritmer.

Evnen til abstraktion udvikles, herunder anvendelsen af strukturer og mønstre.

Nødvendigt for at kunne forstå computationelle strukturer, processer, artefakter og systemer.

Teknologianalyse
Formålsanalyse
Brugsstudier
Konsekvensvurdering
Redesign

Rammesættelse
Idegenerering
Konstruktion
Argumentation
Introspektion

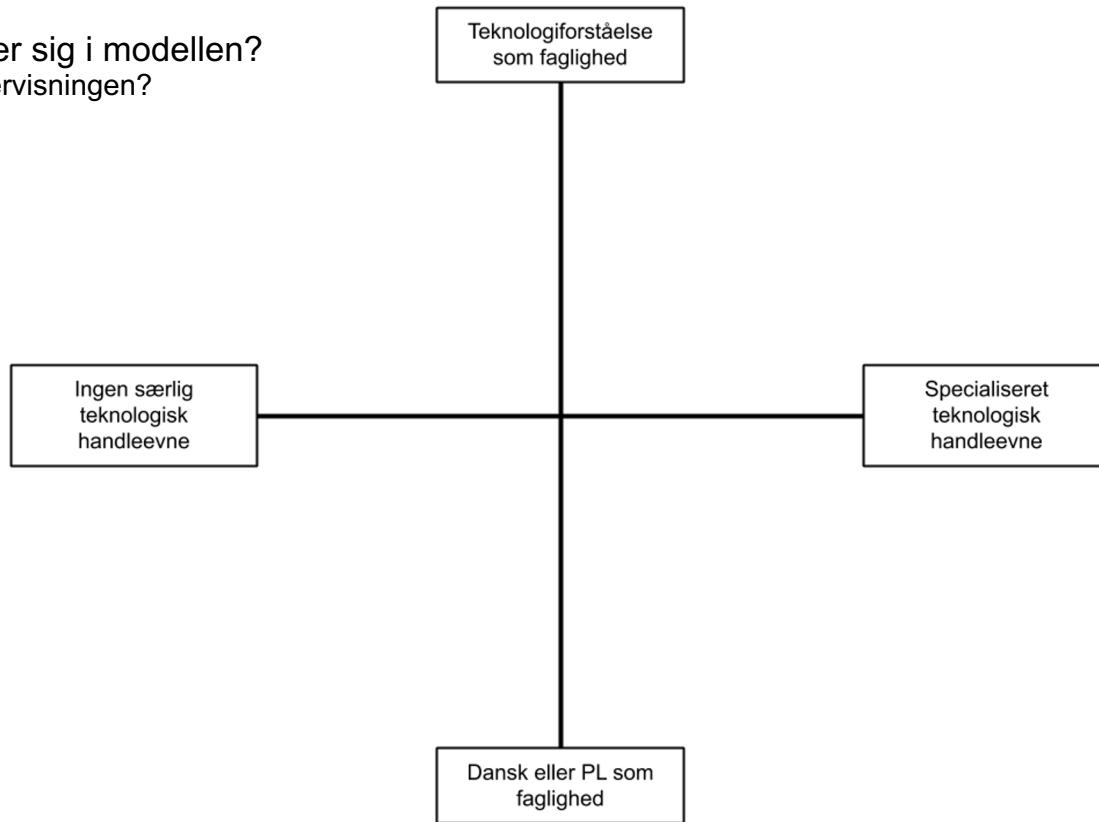


Data
Algoritmer
Strukturering
Modellering

Computersystemer
Netværk
Programmering
Sikkerhed

TEKFAST-model

Hvor tænker I at dette laboratorie placerer sig i modellen?
Hvad kan jeg selv, og hvad kan jeg bruge i undervisningen?



Tandbørstning. Mød en teknologi..eller skal vi sige en tænkning..

Instruerende tekster

1. 2. 3.

Tillykke I har nu skabt en algoritme, argumenter for hvorfor jeres algoritme er velfungerende og hvad der er afgørende for, at den kan være det!

1. 2. 3. 4. 5. 6.

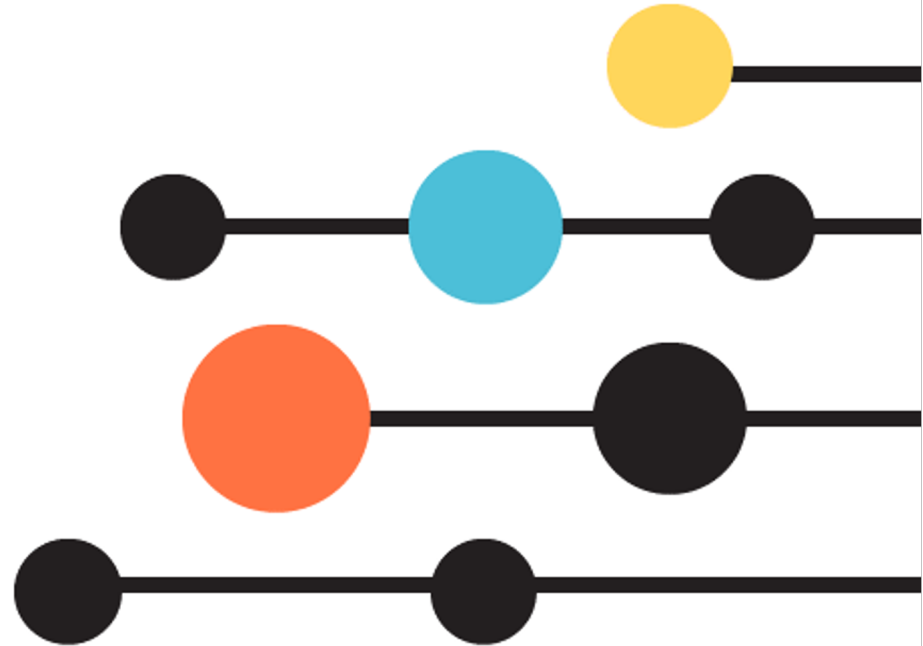
3 min **Tænke og skriv alle ord** og den viden du har om tandbørstning ned
3 min overvej hvornår der er **gentagelser** i det, du gør, eller du **gør det samme**, lav en lille oversigt
2 min se på dine noter, så om du kan gøre dem mere **simple eller overordnede**
4 min Du skal nu på baggrund af dit arbejde lave **en opskrift** på tandbørstning til din kollega, du kan gøre det på papir eller skærm; skriv, tegn, farv osv. for betydning. Du bestemmer
10 min iteration og udbygning af **opskrift - algoritme** i fællesskab

I skal afprøve jeres opskrift- algoritme på jeres kollega, husk at når du er den der modtager opskriften må du KUN gøre som der står i opskriften. Udbyg og udvikl jeres opskrift undervejs. Sammenlign og diskuter...

Se også denne video:

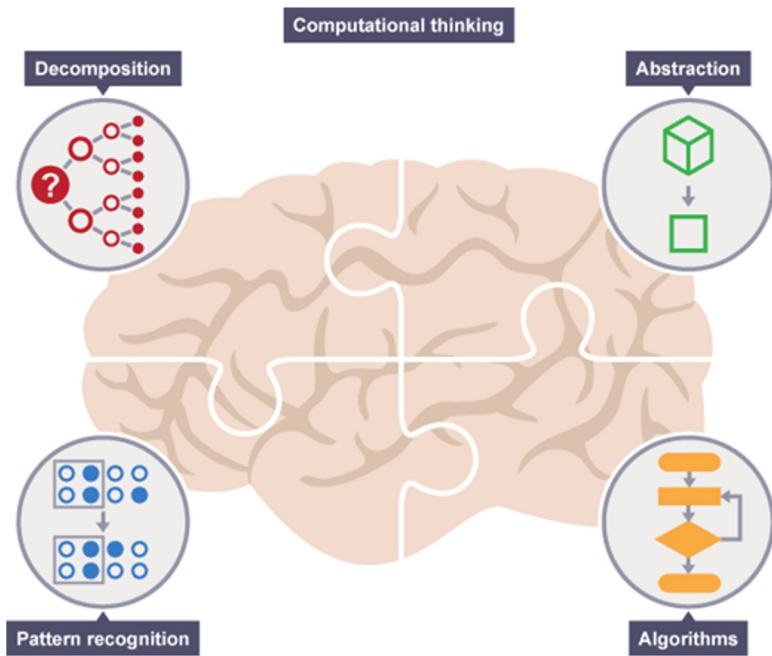
<https://www.youtube.com/watch?v=Ct-IOOUqmyY>

Rammesætning



Hvad gjorde vi lige? Gennemlysning af arbejdsproces i computationel tænkning

- *If this then that*



Computational tænkning indebærer at tage komplekse problemer og nedbryde det i en række små, mere håndterbare problemer (**dekomposition**). Hver af disse mindre problemer kan derefter ses på individuelt, i betragtning af hvordan lignende problemer tidligere er blevet løst (**mønstergenkendelse**) og kun fokuseret på de vigtige detaljer, mens man ignorerer irrelevant information (**abstraktion**). Dernæst kan enkle trin eller regler til løsning af hvert af de mindre problemer designes (**algoritmer**).

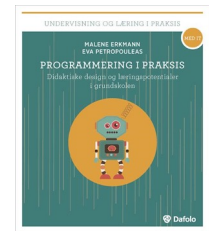
Endelig bruges disse enkle trin eller regler til at programmere en computer til at hjælpe med at løse det komplekse problem på den bedste måde.

Barr & Stephenson, 2011	Lee et al., 2011	Grover & Pea, 2013	Selby & Woollard, 2013	Angeli et al., 2016
Abstraction	Abstraction	Abstractions and pattern generalizations	Abstraction	Abstraction
Algorithms & procedures		Algorithmic notions of flow of control	Algorithmic thinking	Algorithms (including Sequencing and Flow of control)
Automation	Automation			
	Analysis			
		Conditional logic		
Problem Decomposition		Structured problem decomposition (modularizing)	Decomposition	Decomposition
		Debugging and systematic error detection		Debugging
		Efficiency and performance constraints	Evaluation	
			Generalizations	Generalization
Parallelization		Iterative, recursive, and parallel thinking		
Simulation		Symbol systems and representations		
		Systematic processing of information		

Et andet dansk bud...

1. Logisk ræsonnement
2. Algoritmisk tænkning
3. Dekomposition
4. Abstraktion
5. Mønstre og generaliseringer
6. Evaluering

*Malene Erkmann
& Eva
Petropouleas:
Programmering i
praksis (2017)*



Algoritme og data - kerneområder i computationel tænkning

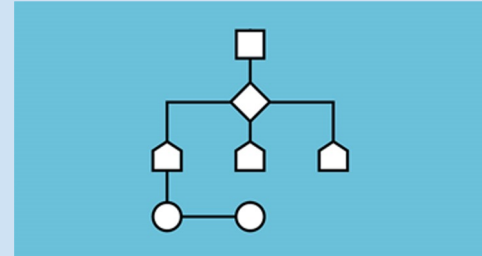
Data

Information, der kan lagres og behandles af en computer



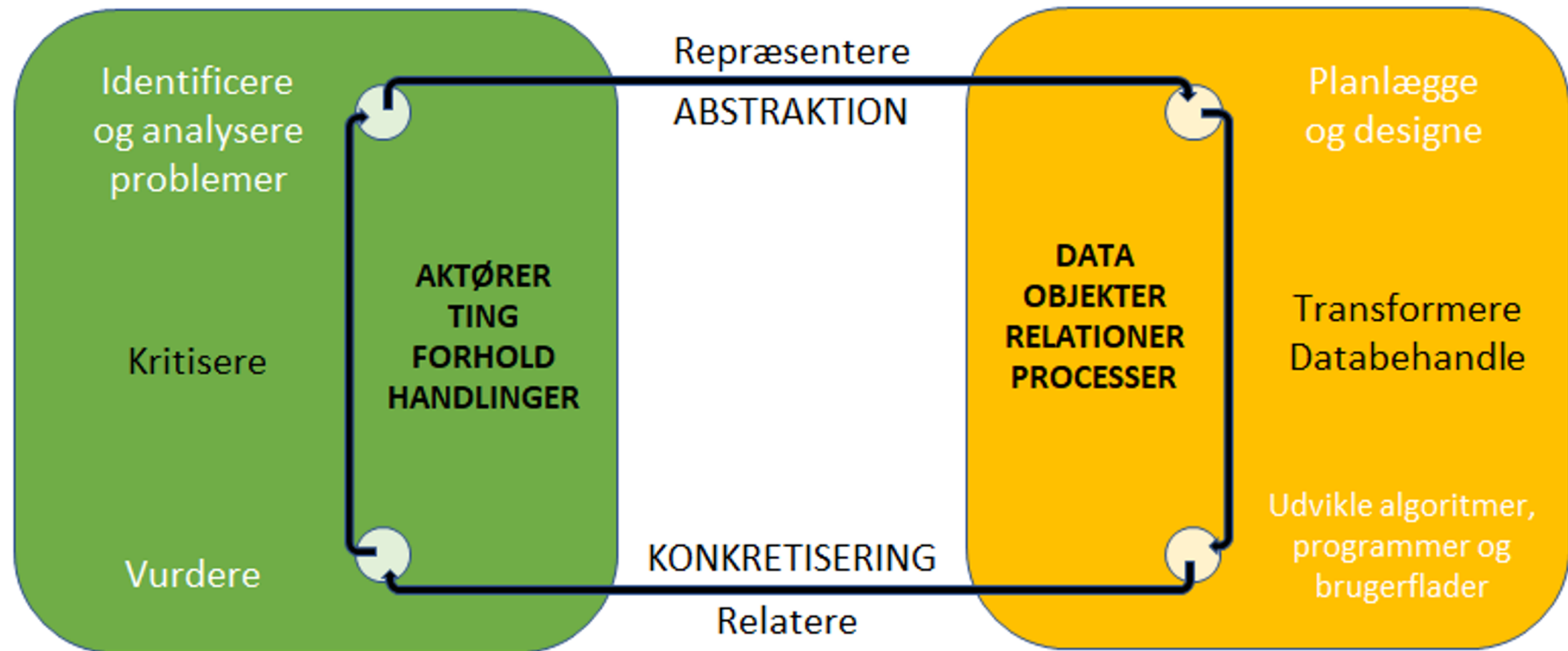
Algoritmer

„Opskrift“
Handlinger



Verden

Model



GUNILLA WOLDE

Totte bager



CARLSEN

FIXED LAYOUT

Til Tottes kage skal du bruge:
1 æg, 1 kop sukker, 1 kop mel og
3 store teskefulde smeltet smør.
Bages ved 200 grader i ca. 20 minutter.



Totte bøgerne fortæller om små og store
hverdagsoplevelser, som børn føler glæde ved
at genkende og tale om.

CARLSEN



Totte tager først ægget og knækker det
på skålens kant. Så hælder han det,
der er inde i ægget, ud i skålen.



Hovsa! Det røg ud i en fart!
Sikke et held, at det hele
kom i skålen.



83

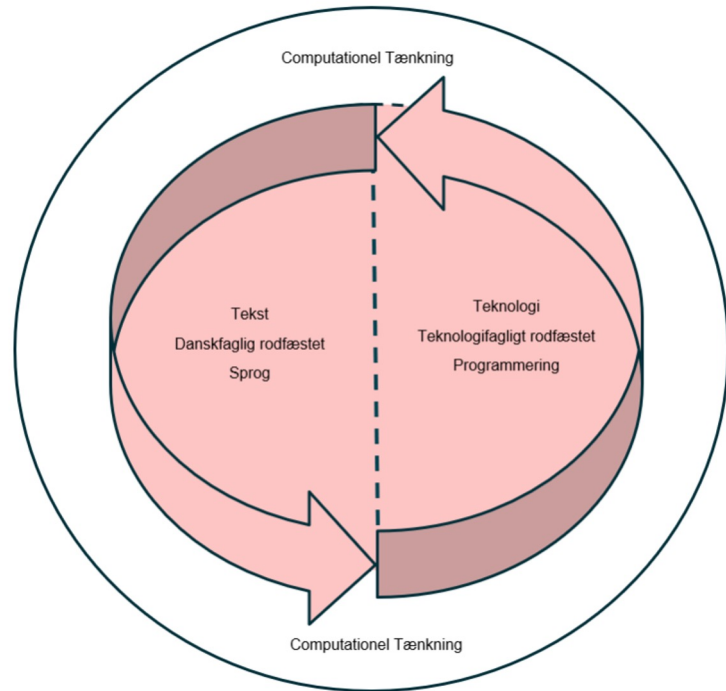
If this then that...

Computational tænkning møder danskfaget og rammesætning

Hvad: Mødet mellem fiktiv tekst (sprog) og computational tænkning (proces, programmering og algoritmer) Computational tankegang og narration drives af overlappende logikker omkring **if this then that**

Hvorfor: For at blive klogere på hvordan dette teknologifaglige indholdsområde, hvori en stor del af teknologiforståelsesfagets fagsprog bor, kan spille sammen med og potentielt berige danskfagets kerneområder (*se også slide 20)

Hvordan: Vi har forsøgt at understøtte det udforskende og laboratorierettede igennem forskellige typer af understøttelse som kan frisætte jer i en arbejdsproces. Ressourcer som man kan til- eller fravælge undervejs i laboratorierne. Ressourcer: Handouts, tekster og videoer til hvert laboratorie



Mennesket i programmet

“Programs are meant to be read by humans and only incidentally for computers to execute.”

A programmer is ideally an essayist who works with traditional aesthetic and literary forms as well as mathematical concepts, to communicate ...”

Donald E. Knuth

Sprog & programmering



Begrundelser for Computational Thinking

Tre begrundelser for at beskæftige sig med CT

- 1) et øget behov for at uddanne sig til et arbejdsmarked, det efterspørger denne kompetence (Wings argument)
- 2) denne tænkemåde er nyttig i almindelighed også selv om man ikke skal være IT-ingeniør (også Wings argument, selv 'the arts' kan bruge CT!)
- 3) det at interagere med verden gennem computationelle processer er en kreativ og undersøgende måde at forstå verden på (Paperts 1970'er-perspektiv: "learn to code, code to learn")



Computational Thinking og literacy

Ligesom afkodning og sætningsforståelse ikke er endemålet for danskundervisningen, men en vej til at opbygge vores kommunikative handlemuligheder, er CT en vej til noget andet end blot programmering.

Programmering er en færdighed, man må tilegne sig, hvis vi for alvor vil bruge computeren til kreative og undersøgende læreprocesser.

“[...] forholdet mellem grammatik og kommunikativ handlen er det samme som forholdet mellem reglerne for at skrive et program i et givet programmeringssprog og det - som jeg i mangel af bedre - vil kalde computationel handlen.” Ulf Dalvad Berthelsen

Det dur ikke hvis teknologiforståelse/CT lukker sig om sig selv. Ligesom det heller ikke dur, hvis grammatiske færdigheder lukker sig om sig selv i undervisningen...



Verden

Det teknologifaglige fagsprog

Model



Parole
Sprog i brug

INFORMATION
TING
FORHOLD
HANDLINGER

Reference

Repræsentere

Langue
Sprog
som
system



DATA
OBJEKTER
RELATIONER
PROCESSER

Algoritmer

Signifiant
&
Signifié

Påvirke

NB Den digitale myndiggørelse og designtænkningens rolle

Teknologiens mikroverden

En mikroverden er ifølge Seymour Papert (1984):

"it's a little world"

"a little slice of reality"

"It's strictly limited"

"a safe place for exploring"

"You will never get into trouble"

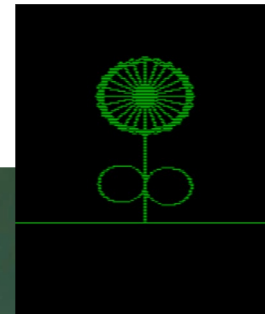
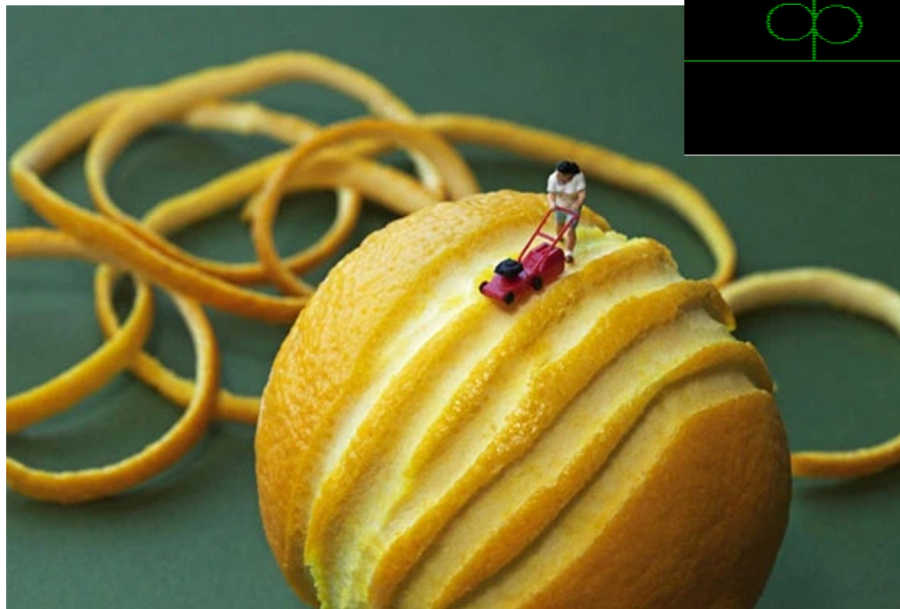
"You will never feel "stupid.""

"it will never fall to pieces or bite you or give you a low grade."

"You are totally safe in this little world. "

"Designed to be discovery-rich - little nuggets of knowledge have been

scattered around in it for you to find. "



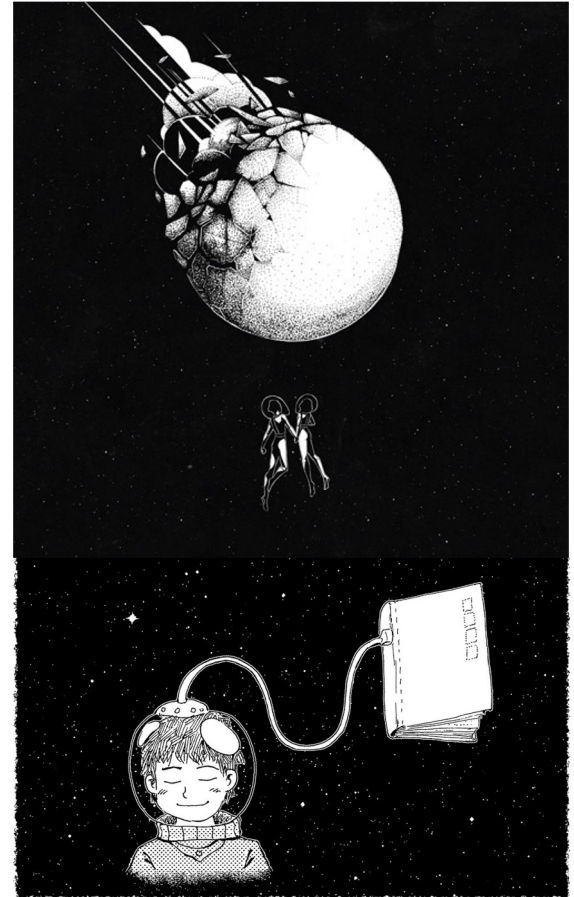
Tekstens mikroverden

Teknologiens mikroverden møder Tekstens mikroverden. Vi har valgt fiktion fordi det er...sværest

Hvilken verden udgør en fiktiv tekst?

- forestillinger om verden gennem en som-om-virkelighed
- en sansebåret tilgang til verden, som stemmer en (æstetik)
- muligheden for at føle den andens følelser (empati)
- indlevelsen i en historie (det narrative)
- forudsætter fortolkning, omsætning fra del til helhed, at læse med fordobling

Potentialet er så, at man ved at dykke ned i tekstens mikroverden, møder et komplekst spejl af virkeligheden, som måske kan gøre en klogere, mere sensitiv, anderledes....



Teknologiens sprog: Pseudokode, flowcharts og programmering (3 steps)

Flowchart en måde at afbilde algoritmer, arbejdsgange eller processer

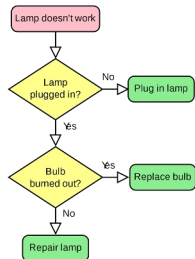
Kræver ikke, man kan et programmeringssprog

Fokus på rækkefølge

Fokus på muligheder af valg

Grafiske former er bærere af betydning men ingen fast syntaks

Kan laves analogt og digitalt



Pseudokode fungerer som et alternativ til rutediagrammer og kommunikerer også en algoritmisk ide

Kræver ikke, at man kan et programmeringssprog

Skriftsproglig modalitet

Uformel skitsering af et computerprogram men har ingen syntaks

Kan laves analogt og digitalt

```
Input tryk på knap A
  Hvis tryk vælg vilkårligt mellem
  tekst (PLAT)
  eller
  tekst (KRONE)
```

Programmering (kodning eller udvikling) dækker over udvikling af software og applikationer m.m.

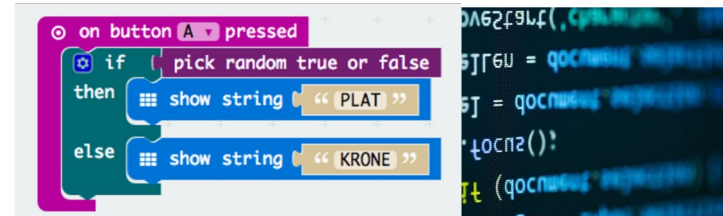
Det du gør for at få applikationen til at besidde funktioner

Proces: få datainput, udføre job, levere output

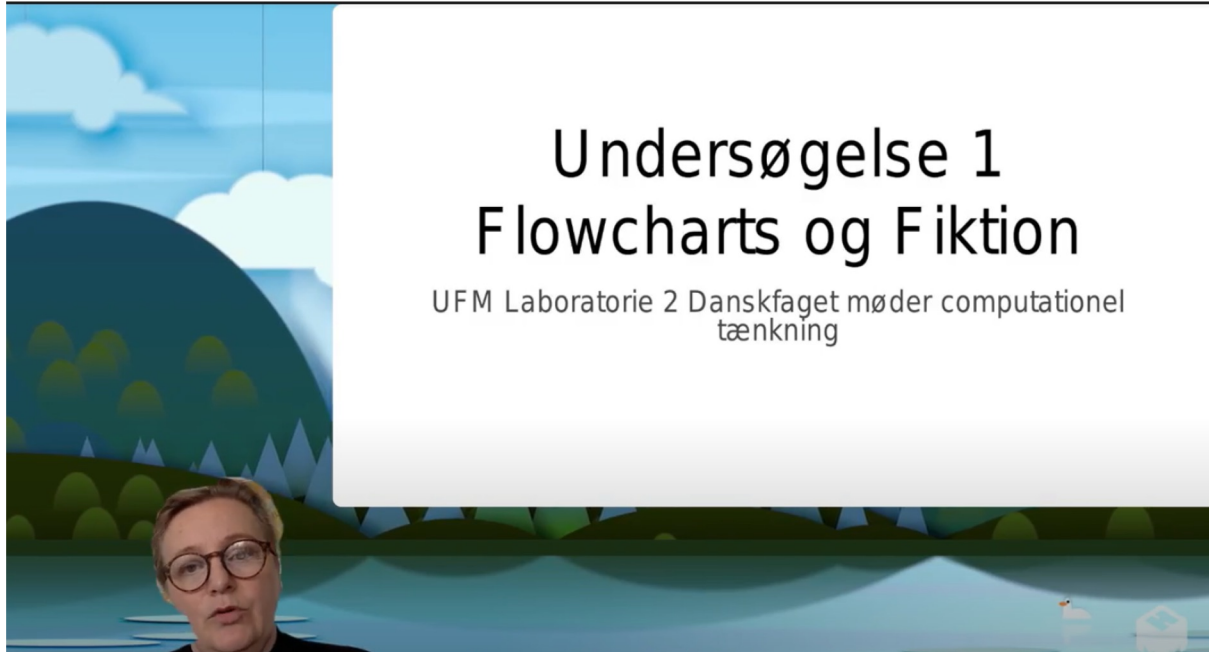
Kunstigt sprog, fokus på funktion, kommer i flere varianter

Forstås af computeren

Kan laves digitalt til digitale artefakter



Se video om Flowcharts og fiktion



Undersøgelse 1 - Flowcharts og Fiktion

I skal **skabe flowcharts** af fiktionstekster for at **gennemlyse tekstens algoritme**.

I kan arbejde på **makroniveau** i de narrative strukturer eller på **mikroniveau** på sætnings og betydnings plan. I bestemmer selv hvor tro I er mod teksten, hvor tæt på teksten I vil være. Jo tættere, jo større udfordring

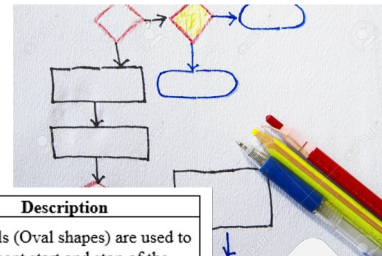
I kan gøre det **digitalt** eller **analogt**

I skal gøre brug af **flowchartets grundlæggende alfabet**, men I kan opfinde nye figurer, som I tillægger betydning, så længe I kan forklare det. Brug farver m.m. til at skabe betydningsklynger (alle processer er f.eks. blå etc.)

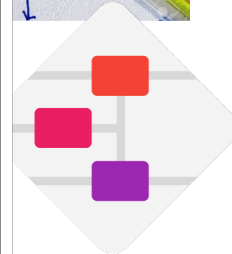
Se i ressourcer til *Flowcharts*



miro



Flowchart Symbol	Symbol Name	Description
	Terminal (Start or Stop)	Terminals (Oval shapes) are used to represent start and stop of the flowchart.
	Flow Lines or Arrow	Flow lines are used to connect symbols used in flowchart and indicate direction of flow.
	Input / Output	Parallelograms are used to read input data and output or display information
	Process	Rectangles are generally used to represent process. For example, Arithmetic operations, Data movement etc.
	Decision	Diamond shapes are generally used to check any condition or take decision for which there are two answers, they are, yes (true) or no (false).
	Connector	It is used connect or join flow lines.
	Annotation	It is used to provide additional information about another flowchart symbol in the form of comments or remarks.



www.draw.io

Undersøgelse 2 - Tekster i Twine

I skal **skabe transduktioner/interaktive fortællinger af fiktionstekster.**

I kan arbejde på **makroniveau** i de narrative strukturer og/eller på **mikroniveau** på sætnings- og betydningsplan.

I bestemmer selv, hvor tro I er mod teksten, og hvor tæt på teksten I vil være.

I skal gøre det **digitalt** på twinery.org

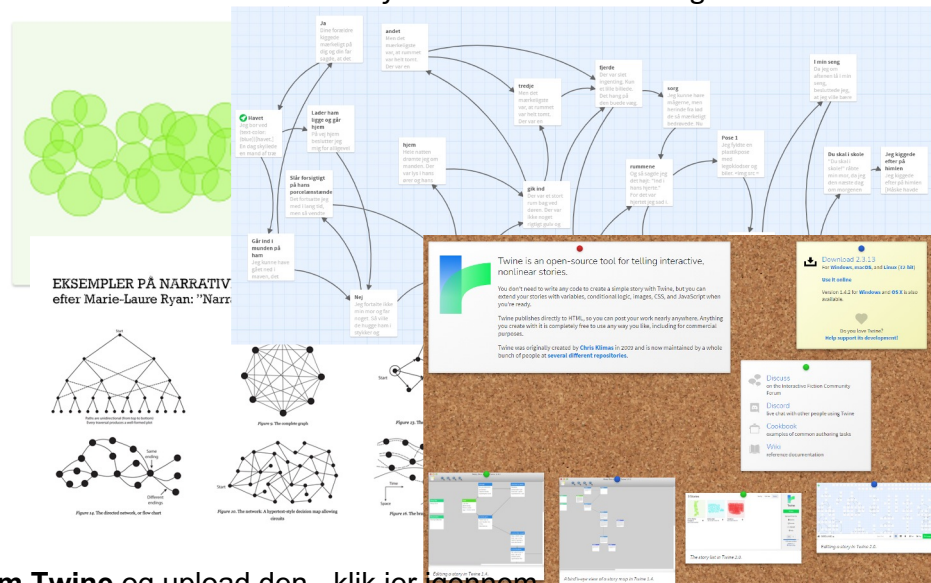
På baggrund af den valgte tekst skal I lave den **hypertekst**, der ligger til grund for transduktionen/den interaktive fortælling.

Arbejdsgang

Se også dokumentet: Generelle koder i Twine

- Hent filen **Twine om Twine** og upload den - klik jer igennem (**Højreklik, download, åbn**)
- Se de to videoer(1+2) om vores oplevelser med Twine i mappen
- Hent filen **Den strandede mand** og upload den - klik jer igennem
- Se generelle videoguiden til Twine
- Gå på opdagelse i mappen og lav jeres egen Twine-fortælling ud fra de fiktive tekster

For at få et overblik over en interaktiv fortælling, altså strukturen for hypertekstens netværk af forbundne tekster, illustreres *hypertekstens struktur* eller *fortællestrukturen* ofte grafisk





Se videoer om twine i ressourcer

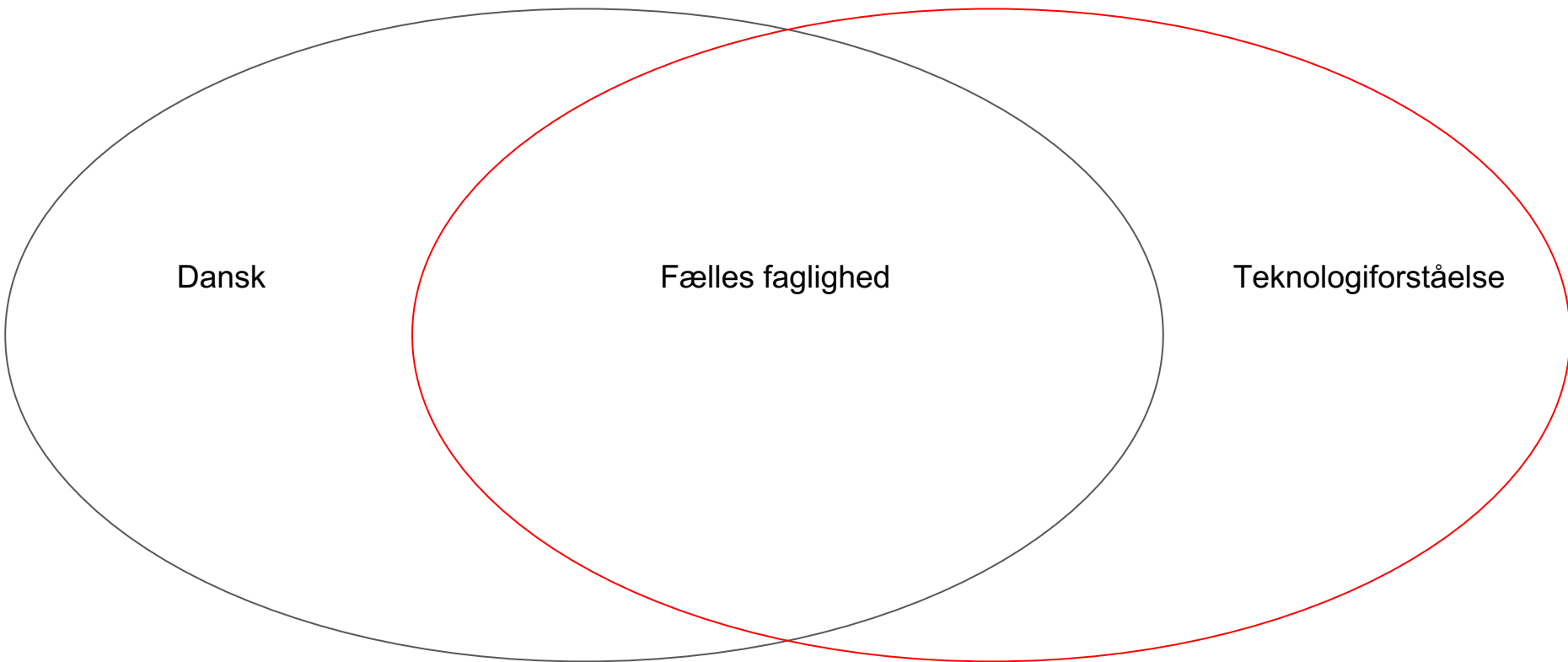
Videndeling



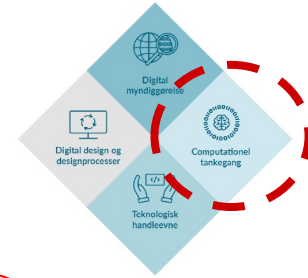
Forstyrrelsesspørgsmål - til yderligere drøftelse

- Hvad er forskellen på at lave et flowchart og en feltanalyse?
- Hvad er forskellen på at kigge nærmere på en tekst i Twine og i en berettermodel?
- Sæt tre ord på: Hvad betyder processen for din oplevelse af værket?
- Er der nogle fiktive tekster, som egner sig bedre til en transduktion i Twine end andre? Hvis ja, hvad kendetegner disse tekster?
- Hvad betyder tekstarbejdet i Twine/flowchart i forhold til dit fænomenologiske møde med værket?
- Hvad betyder tekstarbejdet i Twine/flowchart i forhold til din opfattelse af teksten som struktur?
- Hvorfor bøvl med det her? Er det for at blive bedre tekstlæsere eller bedre programmører?
- Hvordan passer eller passer følgende udsagn ikke i forhold til at omsætte tekster til Twine/flowchart? *Any interpretation given of a certain portion of a text can be accepted if it is confirmed, and must be rejected if it is challenged, by another portion of the same text. In this sense, the internal textual coherence controls the otherwise uncontrollable drift of the reader.* (Eco)
 - Er det her vold mod teksten?

Fælles faglighed



Et mødested mellem fagligheder



Dansk

Fortolkning
Læsning
Fiktion
Lyrik
Instruerende tekster

Fælles faglighed?



Teknologiforståelse

Computational tænkning
Algoritmer
Flowharts
Programmering

Afrunding

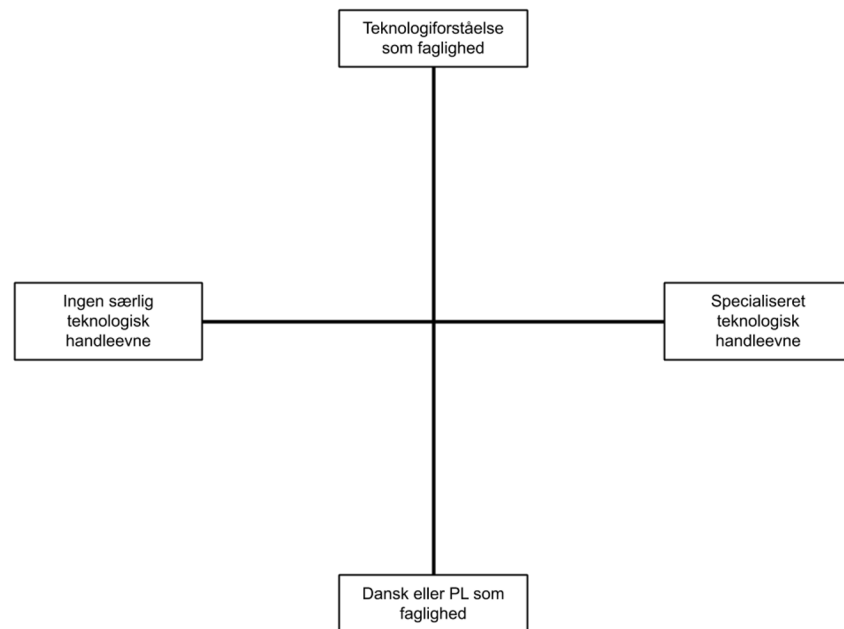
Placer dig selv - med navn

Placer dagens aktiviteter

Algoritmer og tandbørstning - **rødt** kryds

Flowcharts og fiktion - **grønt** kryds

Tekster i twine - **blåt** kryds



Central litteratur brugt i slides og videoer

- Dalvad Berthelsen, Ulf *Computational tænkning - mennesker maskiner og pædagogik* i KVAN nr 117 teknologiforståelse
- Erkmann, Marlene & Petropouleas, Eva (2017) Programmering i praksis Dafolo
- Fink Lorentzen, Rasmus (2009) *Kompetenceudvikling i mødet med det fiktive* i Kompetencer i dansk Gyldendal
- Papert, S *Microworlds: Transforming Education*, note MIT
- Illum Hansen, T (2020) *Teknologiforståelse som praktisk klogskab - om variation og virksomhedsformer i teknologiforståelse som fag* i Unge Pædagoger 1
- Brodersen, P, Illum Hansen, T, Ziehe, T (2020) *Oplevelse fordybelse og virkelyst. Noter til æstetik i undervisningen* KBH: Hans Reitzel
- Eva-Sophie Katterfeldt, Nadine Dittert, Heidi Schelhowe, (2015). Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops, *International Journal of Child-Computer Interaction*,
- Kress G, Bezemer J (2015) *Multimodalitet, Learning and Communication* Taylor And Francis

Ideer til fiktive tekster

- Den strandede mand - L. Jensen
- Sommerfugledalen - Inger Christensen
- Dorthe Nors Kort over Canada - Wild Swims
- Møtrikker - M. Brenøe
- Tove Jansson Sommerbogen - Spøgelseskoven
- Fædrelandssang - H. Nordbrandt
- I Danmark er jeg født - H. C. Andersen
- Mikro - Schack Staffeldt Aftenhøitid
- Kongens fald -Mikkel synker
- Ædepigen (uddrag) J. Wung-Sung
- Katrine Marie Guldager Styrt