JACOB WITT-LARSEN & KRISTIAN N. HJULER

SKAL BØRN VIRKELIG LÆRE AT KODE

- ELLER VAR DET BARE EN FASE?

HVORFOR ER DET EGENTLIGT AT ELEVERNE SKAL LÆRE AT PROGRAMMERE?

LIDT OM OS

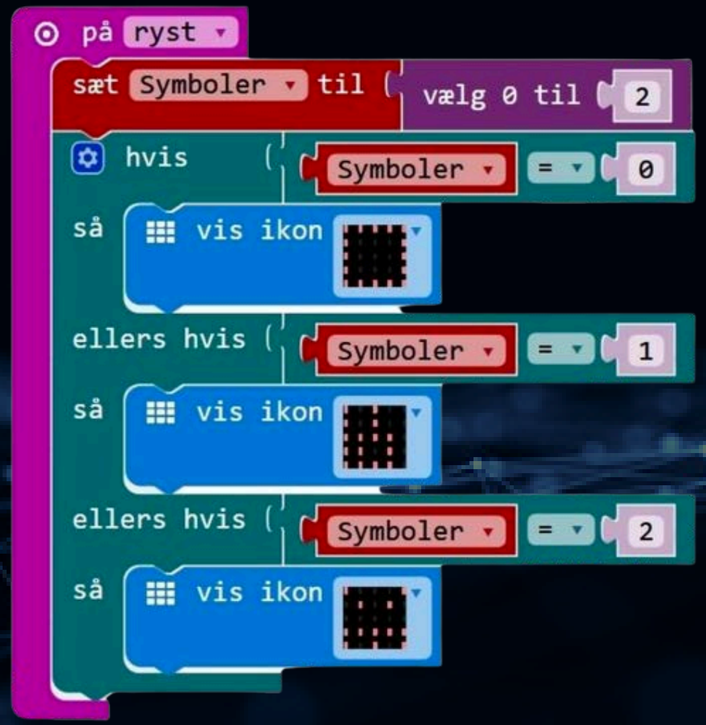
Kristian Neubert Hjuler.
Email: krnh@pha.dk
linkedin.com/in/kristian-neubert-hjuler-a565aa68
Pæd. konsulent for teknologiforståelse
CFU Absalon

Jacob Witt-Larsen
Email: jaco227e@lollandskoler
IT-vejleder,
Makerspaceressourceperson
Byskolen Nakskov, Lolland Kommune

MIL STUDERENDE - 4.SEMESTER

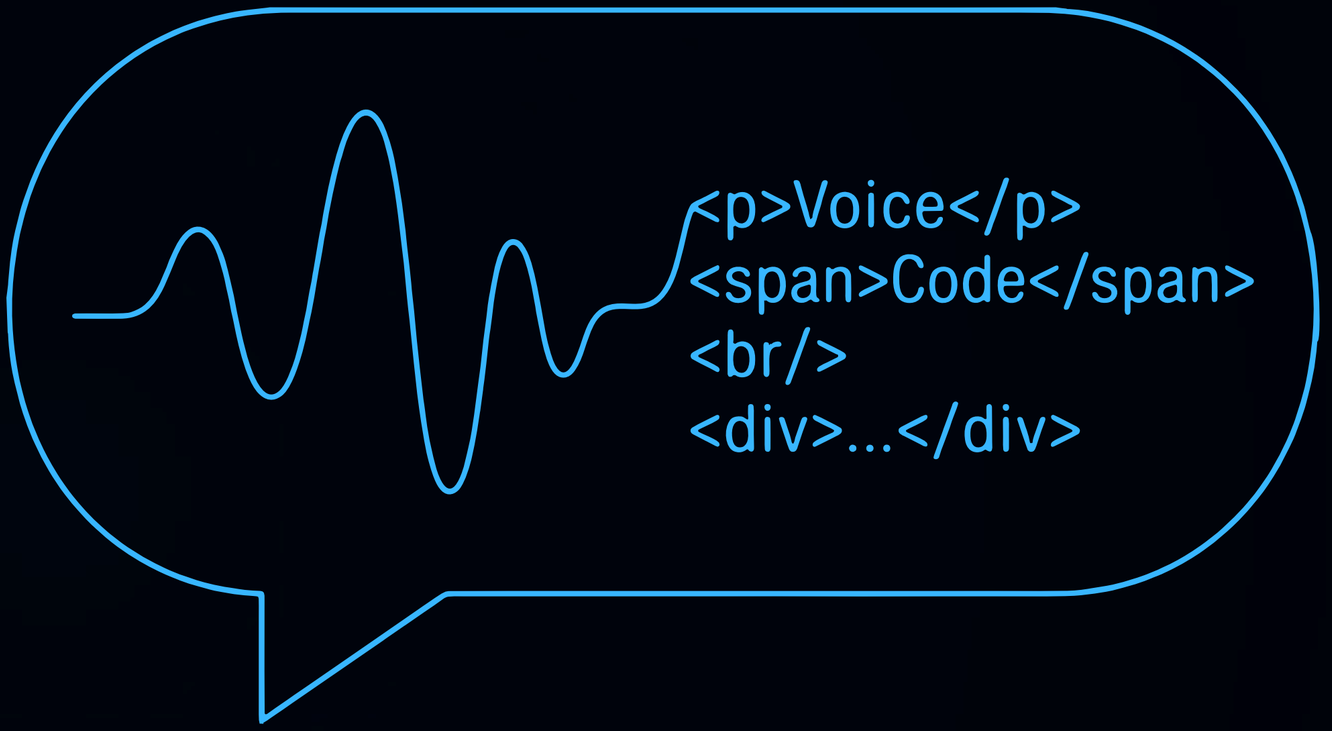
MIL PROJEKTER:
HTTPS://WWW.DIALOGKOMPASSET.DK/OCEANIEN 2084

VEJLEDNING - STEN-SAKS-PAPIR - OPGAVE 1

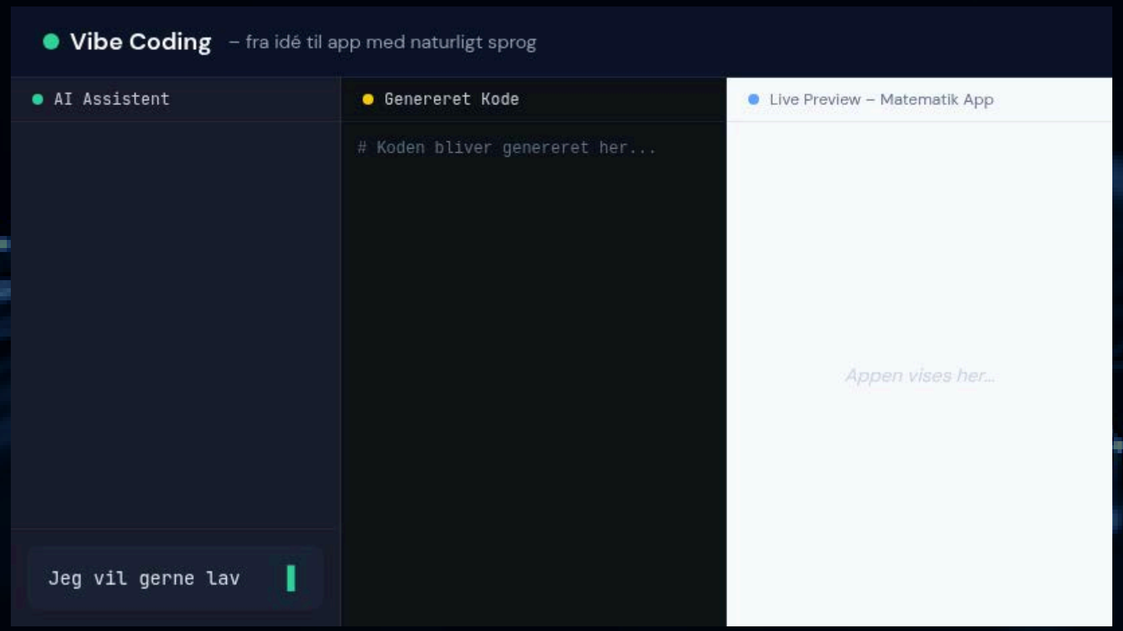


https://www.dr.dk/skole/vejledning-sten-saks-papir-opgave-1

VIBE CODING

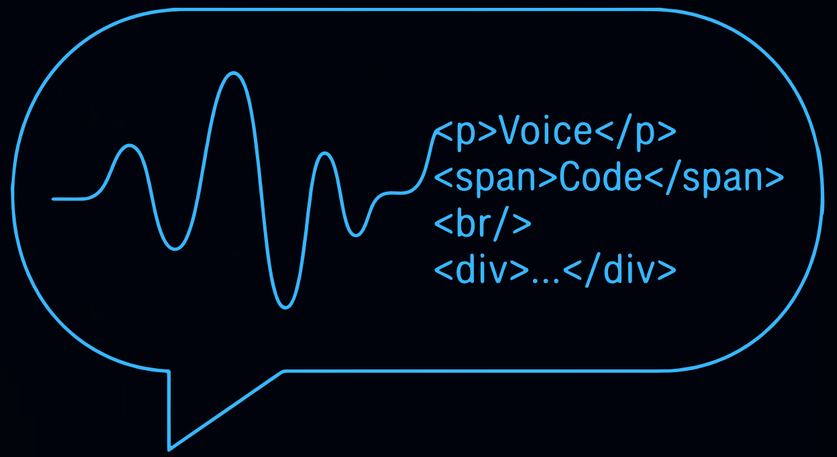

Voice
Code

...


Vibe Coding – fra idé til app med naturligt sprog
AI Assistent
Genereret Kode
Koden bliver genereret her...
Live Preview – Matematik App
Appen vises her...
Jeg vil gerne lav

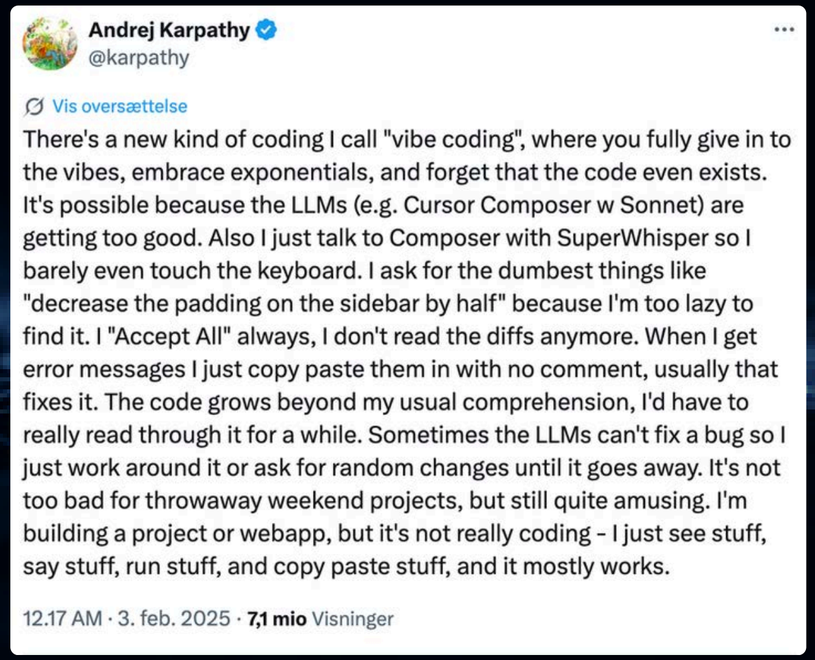
VIBE CODING

ANDREJ
KARPATHY



“I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.”

"Vibe coding", et begreb opfundet af AI-forskeren Andrej Karpathy i begyndelsen af 2025, er en tilgang til softwareudvikling, hvor man bruger kunstig intelligens til selve kodeskrivningen. Frem for at programmere manuelt, beskriver du blot dit mål i almindeligt sprog, hvorefter AI’en genererer et udkast. Processen fungerer som en itterativ proces. Du prompter, tester koden og beder AI’en om at rette til.



Andrej Karpathy
@karpathy

Vis oversættelse

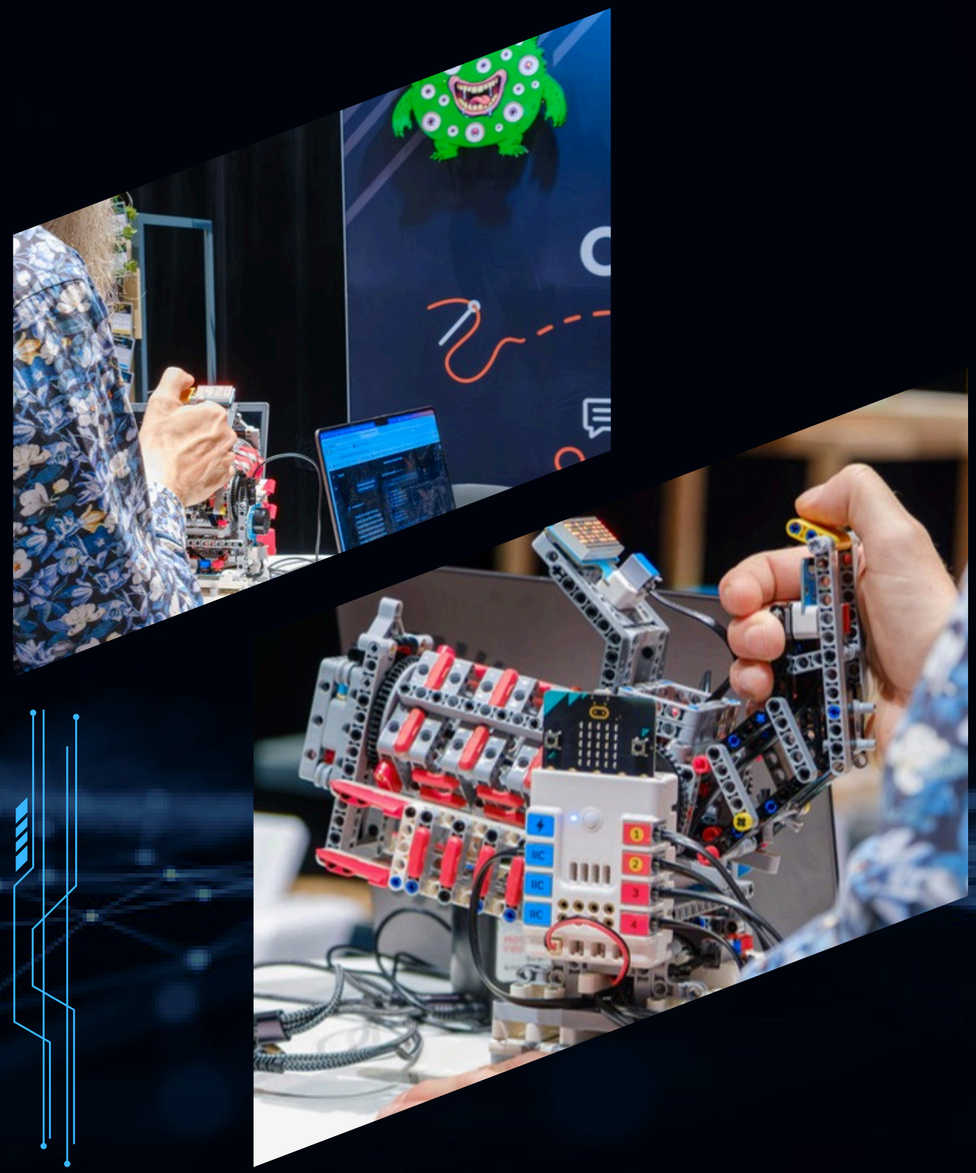
There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

12.17 AM · 3. feb. 2025 · **7,1 mio** Visninger

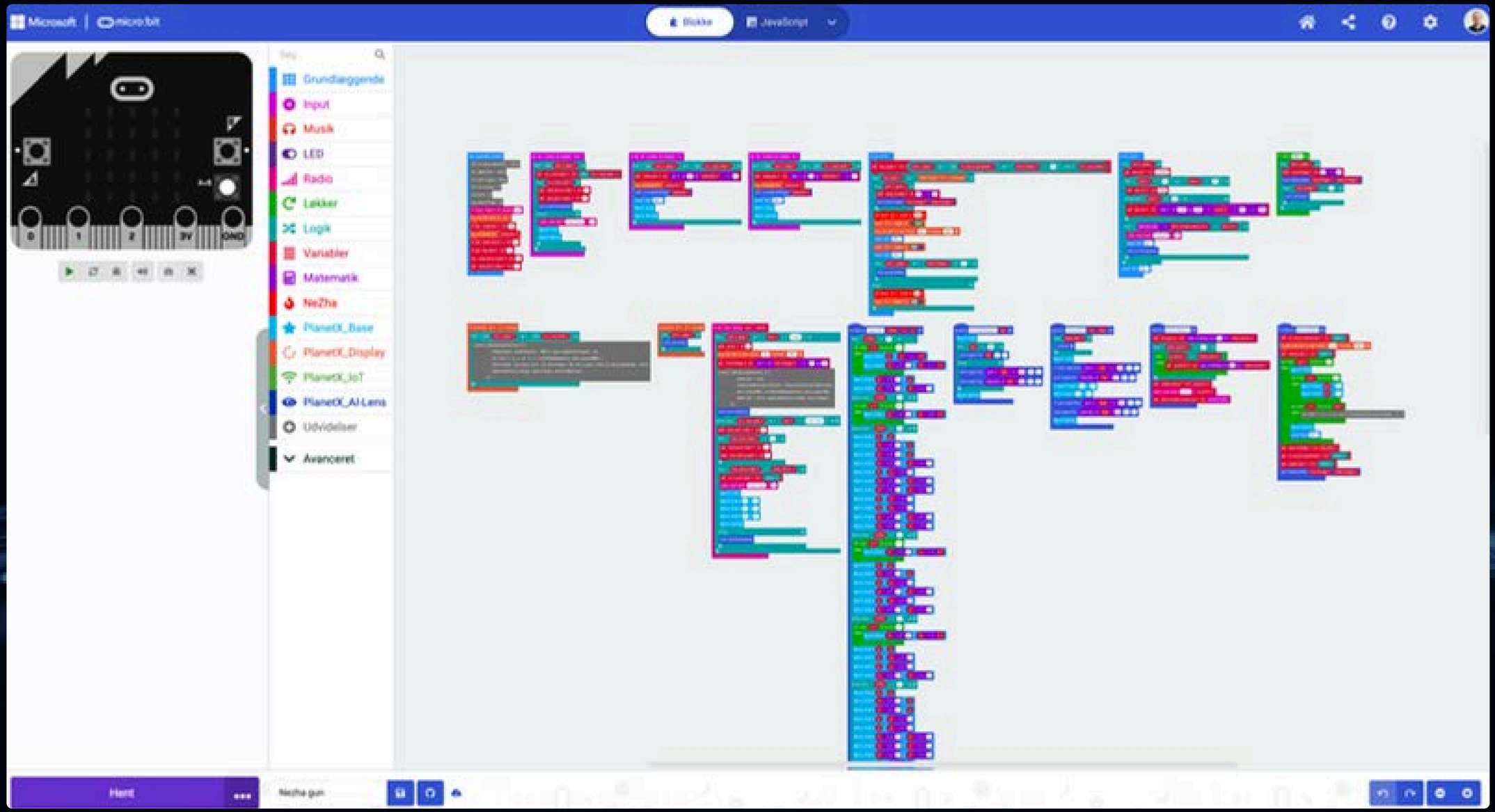
https://karpathy.ai

ET LILLE PROJEKT

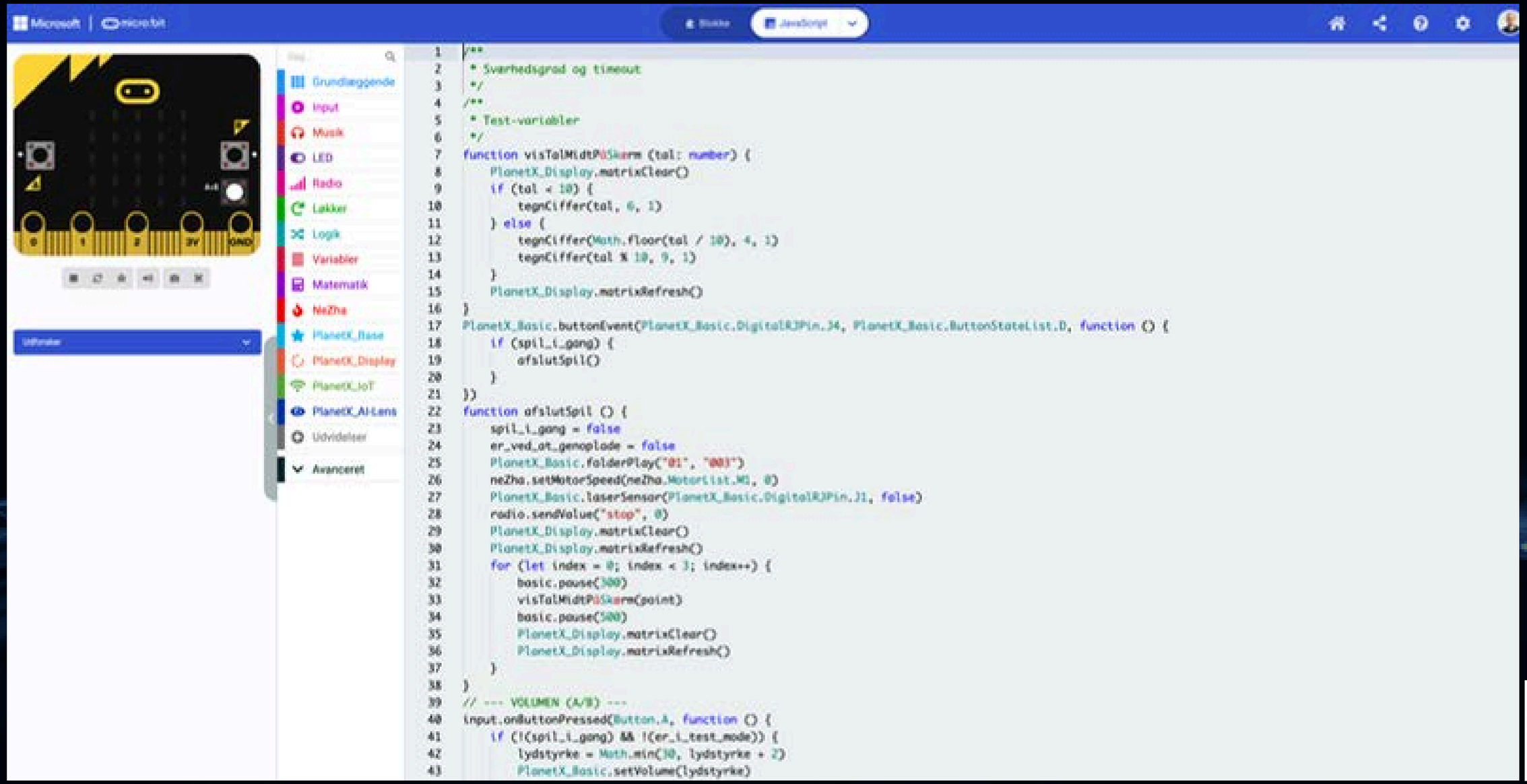
MICRO:BIT LASER-GATLING-GUN



ET LILLE PROJEKT



ET LILLE PROJEKT - 346 LINJER KODE



KALDER AI MÅSKE PÅ...

	DET KLASSISKE (CT 1.0)	VIBE CODING / AI-ASSISTERET KODNING (CT 2.0)
Tilgang til maskinen	**Mennesket lærer maskinens sprog.** Man skal tænke som en computer. Vi nedbryder problemer systematisk og oversætter dem til præcise instruktioner (hvad enten det er byggeklodser eller stram tekstsyntaks), som computeren kan forstå og udføre.	**Maskinen forstår menneskets sprog.** Man formulerer sin vision, sit behov og de overordnede mål i naturligt sprog (prompts). AI'en fungerer som tolk og omsætter visionen til den komplekse kode, der får projektet til at virke.
Abstraktion og Kompleksitet	**Mikrostyring.** Man bygger systemet fra bunden, linje for linje eller klods for klods. Man har fuld kontrol over alle detaljer, men kompleksiteten og kravet til overblik stiger enormt i takt med projektets størrelse.	**Makroarkitektur.** AI tager sig af det underliggende "håndværk". Du overvåger de store linjer og sikrer, at systemets overordnede byggeklodser taler logisk sammen.
Fejlfinding (Debugging)	**Manuel og metodisk test.** Man gennemgår sin egen logik trin for trin. Man leder efter manglende semikoloner, forkerte løkker eller logiske brister og udbedrer dem manuelt i koden.	**Guidende og iterativ dialog.** Fejl opstår i en uigennemskuelig "sort boks". Man løser dem ved at teste programmet, isolere problemet og ændre/præcisere sine instruktioner, så AI'en kan rette koden.
Etik og Kritisk Sans	**Byg-selv-ansvar.** Sikkerhed og logik bygges manuelt ind af programmøren. Fokus i undervisningen ligger ofte på at fremme skabertrang og basal teknologisk forståelse ved at lære at bygge selv.	**Digital dømmekraft.** Det absolutte kerneelement. Da man ikke selv har skrevet hver linje, skal man aktivt fange "hallucinationer", algoritmiske bias, sikkerhedsbrister og dårlig arkitektur valgt af AI'en.
Nødvendige Kernekompetencer	**Algoritmisk håndværk.** Logisk problemløsningsevne og evnen til at mestre et specifikt programmeringsværktøj eller sprog til perfektion for at kunne skabe et produkt.	**Domæneviden og orkestrering.** At kende selve *problemet* og brugerens behov for at bygge den rette løsning. Dertil evnen til at kommunikere skarpt (prompting) og kvalitetssikre AI'ens løsninger (teknisk læseforståelse).

MÅSKE PAPERTS DRØM?

TEKNOCENTRISME

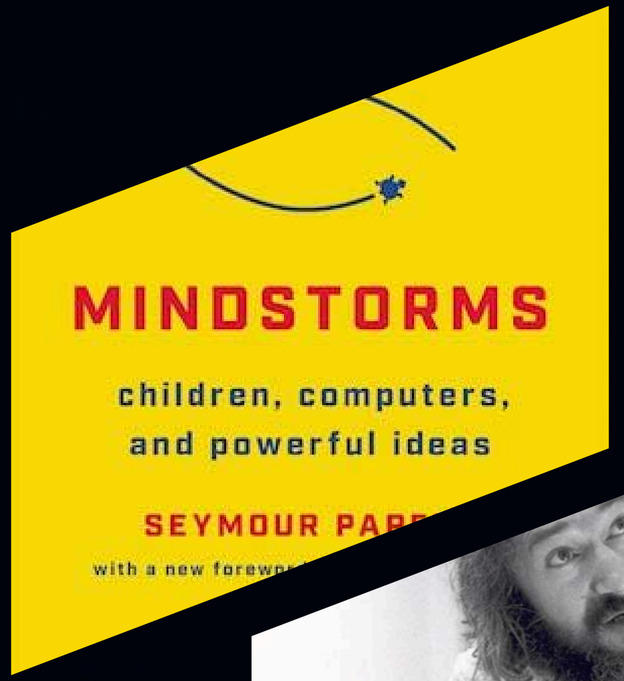
Papert advarede kraftigt mod det, han kaldte teknocentrisme, en fejlagtig tendens til at sætte computeren og teknologien i centrum frem for den kultur og skabertrang, der omgiver den

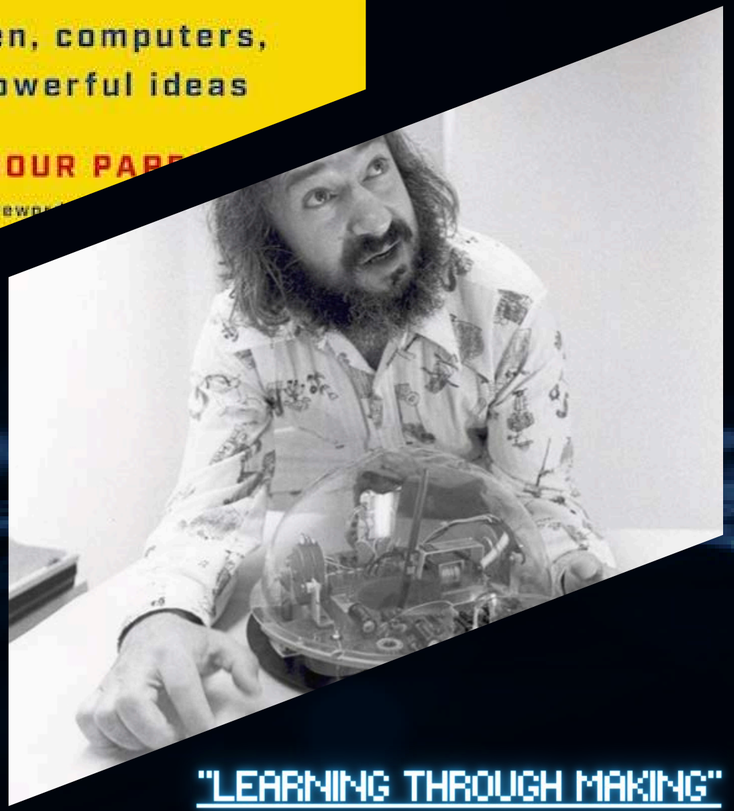
Papert, S. (1987). Computer Criticism vs. Technocentric Thinking. Educational Researcher, 16(1), 22–30. https://doi.org/10.2307/1174251

"BRICOLAGE" OG DEN MALENDE PROGRAMMØR

Papert argumenterer for, at bricolage er en dybt valid og kraftfuld måde at tænke på, som skolesystemet alt for ofte marginaliserer til fordel for abstrakt og formel tænkning

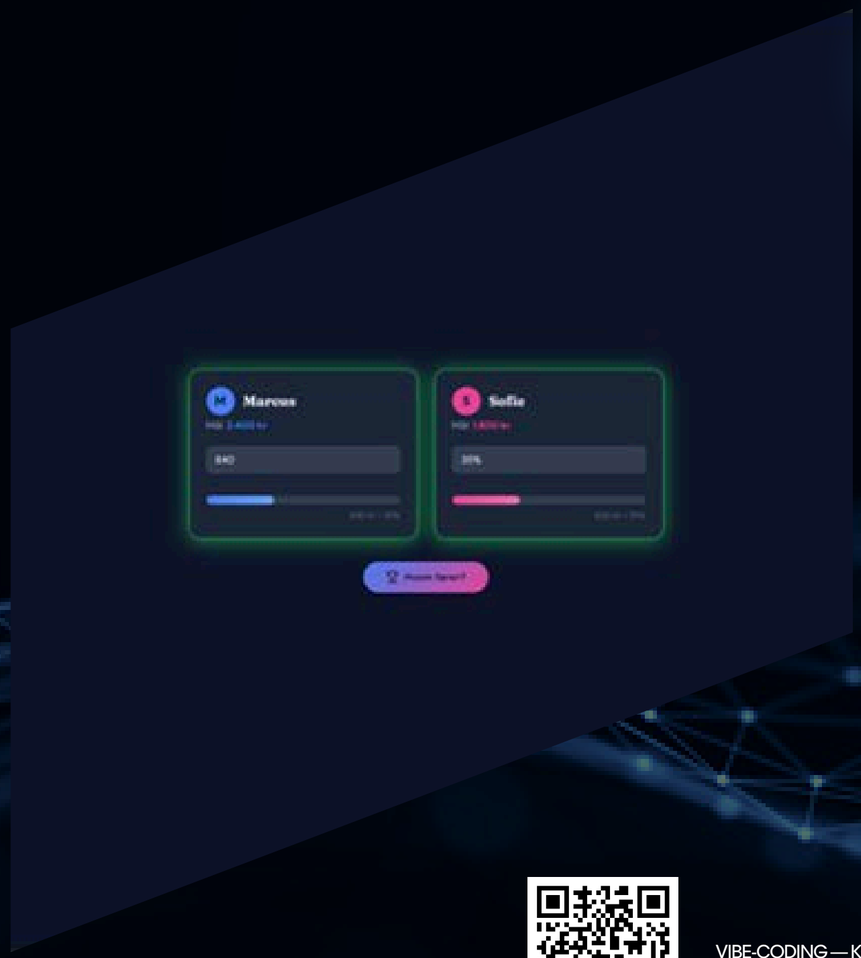
Papert, S. (1993). The children's machine. Technology Review (1899), 96(5), 28–28.





"LEARNING THROUGH MAKING"

VIBE-CODING AFPRØVET I 8. KLASSE

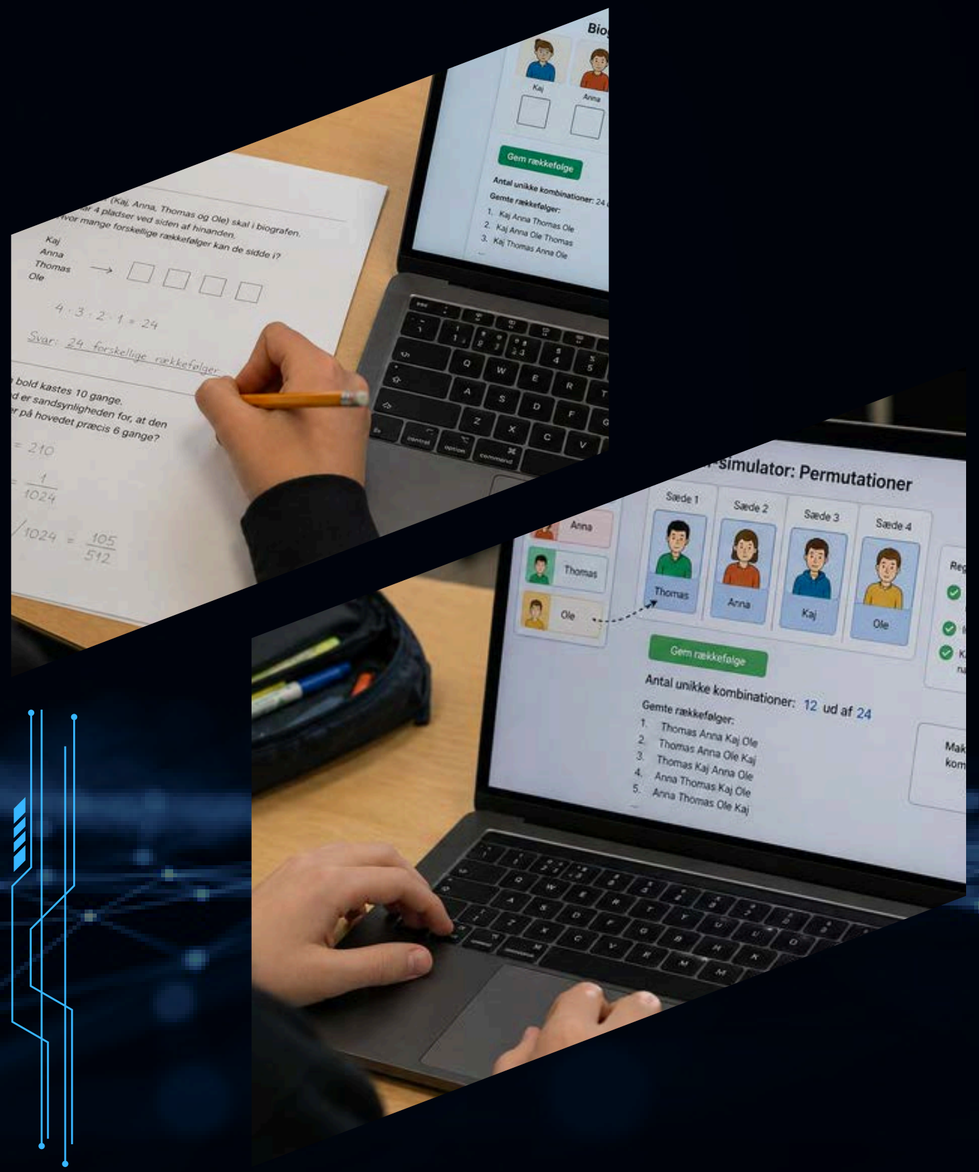


- En fagdag i matematik
- Tre opgaver
  - Ca. 1,5 time pr. opgave
- Eleverne byggede små matematiske systemer med AI
- Mit fokus:
  - Hvad blev synligt i elevernes måde at arbejde på?

VIBE-CODING — KOMPLET FORLØBSPAKKE

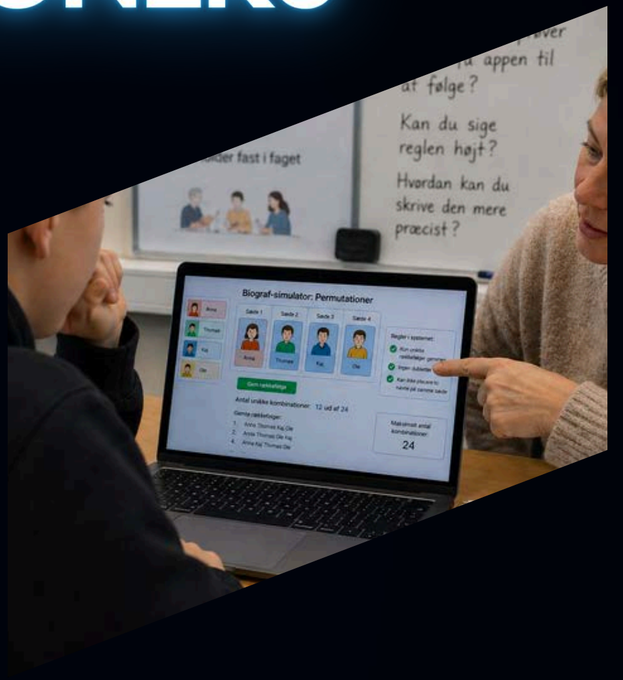
TRE OPGAVER, TRE FORMER FOR FAGLIGHED

1. Biograf-simulatoren
   - Rækkefølger, dubletter og systematik
2. Kort-mysteriet
   - Betingelser, summer og mønstre
3. Procentkoden
   - Sammenhæng mellem kroner, procent og helhed



JEG SÅ TRE TYDELIGE ELEVREAKTIONERS

- NOGLE FLØJ
  - DE GIK HURTIGT I GANG OG EKSPERIMENTEREDE
- NOGLE KÆMPEDE OG BLEV I DET
  - DE SYNTES, DET VAR SVÆRT, MEN FORTSATTE
- NOGLE BLEV RAMT AF FRUSTRATION
  - DET BLEV FOR UVANT ELLER UOVERSKUELIGT



DEN OBSERVATION, DER SATTE SIG FAST

De “dygtige” var ikke altid dem, der klarede sig bedst

Nogle stærke matematik-elever blev frustrerede

Nogle normalt udfordrede elever gik mere undersøgende til værks

- mange muligheder
- mønstre
- betingelser
- forandring
- noget der kan testes

HVAD GJORDE VIBE-CODING SYNLIGT?

ELEVERNES EVNE TIL AT:

- undersøge
- formulere regler
- afprøve idéer
- håndtere fejl
- se mønstre
- forklare deres valg

TRE ERFARINGER FRA AFPRØVNINGEN

- Vælg opgaver, hvor systemet giver mening
  - Mange muligheder, mønstre, betingelser eller forandring

- Kig på processen, ikke kun produktet
  - Hvordan prøver eleven sig frem?

- Brug fejlene fagligt
  - Fejl viser, hvor tænkningen kan undersøges

Så....

SKAL BØRN VIRKELIG LÆRE AT KODE
- ELLER VAR DET BARE EN FASE?

HVORFOR ER DET EGENTLIGT AT ELEVERNE SKAL LÆRE AT PROGRAMMERE?

HÅB

VIDEN

HANDLEKRAFT

PROFESSOR OLE SEJER IVERSEN
AARHUS UNIVERSITET, DANMARK

FREMTIDENS JOBS.
Andreas Steno er teknologiinvestor

“Altså jeg tror, at de fleste, som delte råd ud omkring, hvilken uddannelse man skulle tage for 5-6 år siden, de ville have fortalt dig, at du skulle blive softwareprogrammør. Det tror jeg er det sidste jeg vil forslå dig at blive i dag!