

---

# LLMer som verktøy i læringsprosessen

Anne Bosch  
INF-1400  
20.02.2026

---

---

---

# Hvordan lærer vi?

- Kognitiv konstruktivistisk læringsteori fra Piaget
  - Assimilasjon og akkomodasjon av skjema
  - Ekvilibrasjon
-

---

---

# Teknologideterminisme

- Uunngåelige konsekvenser
  - Negative og positive ytterpunkter
  - Nyansert
  - Nysgjerrig og kritisk
-

---

---

"AI skriver 100% av all kode i 2026, så det er ikke vits å lære seg å programmere. Det kommer ikke til å være behov for nye utviklere, for AI-verktøyene er mye bedre enn mennesker om kort tid."

"LLMer spyttter bare ut det den tror du vil høre. Det som kommer tilbake er et speilbilde av søppel på internett. Du blir lat, du får kode full av feil, og lærer ingen ting."

---

---

# Hvilke type verktøy finnes?

- Chat-interface i nettleseren
  - IDE extensions
  - CLI
    - Claude Code
    - Codex CLI
  - Lokal kjøring uten internett
    - Ollama
  - Dedikerte desktop-apper
    - Claude Cowork
-

---

# Hvilke verktøy er tilgjengelig for dere?

Vi anbefaler:

- ChatUiT
- Microsoft copilot
- GitHub copilot

Andre muligheter:

- Studentlisenser: GitHub copilot pro
  - Produkter fra:
    - Anthropic, OpenAI, Google, Cursor, etc.
  - Bruke 10 minute emails
-

---

---

# Modeller

- Raske modeller
  - Reasoning models
  - Agentic models
  - Åpne/lukkede modeller
  - Ulike Leverandører
-

---

---

# Valg av modeller

- Ikke bruk den dyreste modellen til alt!
  - Bruk billige modeller til å undersøke hva du egentlig har lyst til å spørre om
-



---

---

# Systemprompt

- Custom instructions
  - `.github/custom_instructions.md`
-

---

# Konkret arbeidsprosess

- Små endringer, aldri store endringer
  - Teste/verifisere hele tiden det som kommer ut
    - Gjør koden det den skal
    - Forstår jeg denne koden?
    - Er dette den beste måten å gjøre dette på?
    - Har modellen gjort noen antakelser jeg ikke er enig i?
  - Cognitive load
  - Be koden om å gjøre code review.
-

---

# Måter å bruke LLMer på som ikke er å generere kode

- Kan du forklare meg hva Konsept X er?
    - Hvordan kan dette se ut i Python?
  - Kan du gi meg en code-review av denne pull requesten?
  - Kan du forklare hva som skjer i denne delen av koden?
    - X antall linjer, eller en hel fil
  - Kan du skrive en test for denne funksjonen?
  - Kan du stille meg 10 spørsmål som tester forståelsen min av denne koden?
  - Kan du evaluere designet i programmet med tanke på polymorfi?
  - Kan du evaluere lesbarheten av denne funksjonen?
-

---

# Design et godt prompt - RACE

- Rolle (Role)
  - Hva konkret skal utføres (Action)
  - Relevant kontekst (Context)
  - Beskrive output-formatet du ønsker (Execute / Expected output)
-

---

## Lage et godt prompt del 2

- Be om flere varianter
  - Oppklarende spørsmål før utførelse
  - Evaluere om oppgaven er utført
  - Evaluere sitt eget svar er godt nok
-

---

---

# Oppgave - teste RACE selv

Oppgaven ligger ute i forelesnings-repoet på GitHub under uke8

---

---

# Model-benchmarking

- <https://gso-bench.github.io/index.html>
  - <https://arena.ai/nb/leaderboard/code>
  - <https://www.onyx.app/open-llm-leaderboard>
-

---

# Etiske spørsmål og avveininger

- Lik tilgang til verktøy for alle
  - Akademisk integritet
  - Kildekritikk
  - Personvern og datadeling
  - Meningspåvirkning
    - Hvem er avsender, og hva er deres agenda?
  - Hvem får pengene våre
  - Fotavtrykk på miljøet
-