

# Les 1 — Lesstof

Introductie tot AI en Large Language Models

**Vak:** AI Fundamentals  
**Opleiding:** NOVI Hogeschool Utrecht  
**Cursus:** AI Developer  
**Volgende les:** Les 2 — OpenCode + slim werken

**Vak:** AI Fundamentals **Opleiding:** NOVI Hogeschool Utrecht **Cursus:** AI Developer **Volgende les:** Les 2 — OpenCode + slim werken

## Inhoud

1. [Wat is AI](#1-wat-is-ai)
2. [Large Language Models (LLMs)](#2-large-language-models-llms)
3. [Hoe LLMs tekst genereren](#3-hoe-llms-tekst-genereren)
4. [Hallucinaties — waarom AI fouten maakt](#4-hallucinaties--waarom-ai-fouten-maakt)
5. [Het AI-landschap — providers en modellen](#5-het-ai-landschap--providers-en-modellen)
6. [ChatGPT effectief gebruiken](#6-chatgpt-effectief-gebruiken)
7. [Prompt engineering basics](#7-prompt-engineering-basics)
8. [v0.dev — van prompt naar UI](#8-v0dev--van-prompt-naar-ui)
9. [De slimme workflow — schets naar website](#9-de-slimme-workflow--schets-naar-website)
10. [Ethiek en valkuilen](#10-ethiek-en-valkuilen)

## 1. Wat is AI

**Artificial Intelligence** is een verzamelnaam voor systemen die taken uitvoeren die normaal menselijke intelligentie vereisen — taal begrijpen, beelden herkennen, beslissingen nemen, leren van data.

### Drie generaties AI

- **Symbolische AI** (jaren '60-'80) — regels en logica. "Als X en Y, dan Z."
- **Machine Learning** (jaren '90-2010) — systemen leren patronen uit data. Statistiek, geen regels.

- **Deep Learning + LLMs** (2012-nu) — neurale netwerken met miljarden parameters. Modellen "snappen" taal in context.

Wat je vandaag als AI ervaart — ChatGPT, Claude, Midjourney — zit in die derde generatie.

## Generatieve AI

Het specifieke type AI waar wij in deze leerlijn op focussen: **generatieve AI**. Modellen die nieuwe output maken (tekst, code, beelden, audio) op basis van een prompt.

Subcategorieën:

- **Tekst** — ChatGPT, Claude, Gemini
- **Beeld** — DALL·E, Midjourney, Stable Diffusion, Flux
- **Code** — GitHub Copilot, Cursor, v0.dev
- **Audio/video** — ElevenLabs, Sora, Veo

## 2. Large Language Models (LLMs)

Een **Large Language Model** is een neuraal netwerk getraind op enorme hoeveelheden tekst (terabytes van het hele internet) om de volgende waarschijnlijke woorden te voorspellen.

### Grootte

- **GPT-3 (2020)**: 175 miljard parameters
- **GPT-4 (2023)**: geschat 1.7 biljoen parameters
- **Claude 4, Gemini 2.5, GPT-5 (2025)**: schattingen lopen uiteen, miljarden tot biljoenen

Meer parameters = meer "kennis" maar ook duurder om te trainen + draaien.

### Training in drie stappen

1. **Pre-training** — model leest miljarden teksten en leert patronen voorspellen (volgend woord)
2. **Fine-tuning** — extra training op specifieke taken (vraag-antwoord, codering)
3. **RLHF** (Reinforcement Learning from Human Feedback) — mensen beoordelen antwoorden, model leert wat "goed" is

Het resultaat: een model dat menselijke instructies begrijpt en plausibele antwoorden produceert.

## 3. Hoe LLMs tekst genereren

### Tokens

LLMs werken niet met woorden maar met **tokens** — stukjes tekst van 1-4 karakters. "Programmeren" wordt bijvoorbeeld pro | gramme | ren (3 tokens).

Engelse tekst: ~1 token per 4 karakters  
Nederlandse tekst: ~1 token per 3 karakters (efficiënter dan Engels)  
Code: ~1 token per 3-5 karakters

Test zelf via <https://platform.openai.com/tokenizer>.

## Next-token prediction

Bij elke generatie kijkt het model naar alle vorige tokens en berekent een **waarschijnlijkheidsverdeling** over alle mogelijke volgende tokens. Het kiest één — meestal de meest waarschijnlijke, met een vleugje randomness (temperature) voor variatie.

```
Input: "De kat zit op de"
Mogelijke volgende tokens:
- "mat"      (85% waarschijnlijk)
- "bank"     (8%)
- "stoel"    (4%)
- "tafel"    (3%)
```

Dit gebeurt token voor token, totdat het model "klaar" denkt te zijn of de maximum lengte bereikt.

## Context window

Modellen kunnen alleen X aantal tokens onthouden in één gesprek. Dit heet de **context window**.

- GPT-4o: 128.000 tokens (~250 pagina's tekst)
- Claude Sonnet 4: 200.000 tokens (~400 pagina's)
- Gemini 2.5 Pro: 2 miljoen tokens (~4.000 pagina's)

Verder dan dat = "vergeet" het oudste deel.

## 4. Hallucinaties — waarom AI fouten maakt

LLMs zijn **niet** databases die feiten ophalen. Ze zijn statistische voorspellers van de meest waarschijnlijke woordvolgorde. Soms produceert dat onzin — een **hallucinatie**.

### Wanneer hallucineert AI vaak?

- **Specifieke feiten** — namen, datums, URLs, getallen
- **Onbekende onderwerpen** — over jouw startup, jouw studie, jouw stad
- **Recente gebeurtenissen** — model kent geen data na zijn training cutoff
- **Citaten en bronnen** — verzint vrolijk referenties die niet bestaan

### Wat is een hallucinatie precies?

Niet een "fout" in technische zin. Het model voorspelt wat plausibel klinkt op basis van patronen. Als het waarschijnlijke antwoord toevallig waar is, lijkt het slim. Als het niet waar is, lijkt het slim maar is het fout.

### Voorbeeld

"Wie heeft de eerste Tony's Chocolonely reep gemaakt?"

Een model kan met groot zelfvertrouwen een naam noemen die volledig verzonnen is. Het patroon "wie maakte X bedrijf" + "Tony's" voorspelt iets dat klinkt als een naam — maar dat hoeft de waarheid niet te zijn.

Wat doe je ertegen?

- **Verifieer feiten** — vooral bij namen, cijfers, datums
- **Vraag om bronnen** — en check of die echt bestaan
- **Gebruik AI met search** — Perplexity, ChatGPT met web search, Claude met web search
- **Wees alert** — als iets té perfect klinkt, dubbelcheck

5. Het AI-landschap — providers en modellen

Grote spelers (2025)

| Provider  | Modellen (2025)                        | Sterke punten                          |
|-----------|----------------------------------------|----------------------------------------|
| OpenAI    | GPT-5, GPT-5-mini, o3 (reasoning)      | All-round, populairste API             |
| Anthropic | Claude Opus 4.6, Sonnet 4.6, Haiku 4.5 | Beste voor code + lange context        |
| Google    | Gemini 2.5 Pro, Flash                  | Multimodaal, lange context (2M tokens) |
| Meta      | Llama 4                                | Open-source, gratis te draaien         |
| Mistral   | Mistral Large 3                        | Europese provider, snel                |

Gratis vs betaalde tier

| Tier                | Wat krijg je                                | Geschikt voor             |
|---------------------|---------------------------------------------|---------------------------|
| Gratis ChatGPT      | GPT-5 (rate-limited), beeld/voice           | Dagelijks gebruik, leren  |
| ChatGPT Plus (\$20) | GPT-5 onbeperkt, o3 reasoning, voice modals | Sleu agent, professionals |
| Claude.ai gratis    | Sonnet, beperkte usage                      | Code-vragen, schrijven    |
| Claude Pro (\$20)   | Opus + Sonnet onbeperkt, projects, MCP      | Developers                |
| Gemini gratis       | Flash, 2.5 Pro (lim.)                       | Quick lookups, multimodal |

Voor deze leerlijn raad ik aan: **ChatGPT Plus** (voor algemene workflows) + **Claude Pro** (voor code-werk). Ofwel gewoon één Pro-abonnement van je voorkeur.

6. ChatGPT effectief gebruiken

Voorbij "type vraag, krijg antwoord". Een paar features die echt verschil maken.

---

## Model kiezen

ChatGPT toont verschillende modellen in het dropdown menu:

- **GPT-5** — default, snel, voor 90% van de taken
- **GPT-5 mini** — sneller maar minder slim
- **o3** — "reasoning" model — denkt eerst stap-voor-stap. Trager maar veel beter voor wiskunde, complexe code, planning
- **GPT-4o** — multimodaal (beeld + voice)

Vuistregel: complex probleem → o3 of GPT-5. Quick vraag → GPT-5 mini.

## Tijdelijke chat

Voor gevoelige vragen of testen: **"Tijdelijke chat"** modus. Geen geschiedenis, geen training-data. Verdwijnt na sluiten.

## Custom Instructions

Settings → Custom Instructions. Hier vertel je ChatGPT:

- Wie je bent (developer student in NL, leert AI Developer leerlijn)
- Hoe je antwoorden wilt (in NL, code in TypeScript, niet té lang)

Geldt voor alle nieuwe chats. Scheelt elke keer herhalen.

## Projects

ChatGPT Projects = persistent context voor lange projecten. Maak een Project, upload files, geef context-instructies. Alle chats in dat project hebben die context.

## Voice mode

Vooraf op telefoon: praat tegen ChatGPT. Heel snel voor brainstormen tijdens lopen, koken, autorijden.

## Bestanden uploaden

PDF, Word, Excel, code-files, screenshots. AI leest mee en kan vragen beantwoorden over de inhoud. Voor school-handleidingen, code-bestanden, screenshots van foutmeldingen — werkt geweldig.

# 7. Prompt engineering basics

Een prompt is wat je tegen de AI zegt. Hoe je 't zegt bepaalt voor 80% wat je terugkrijgt.

## Slechte prompt vs goede prompt

Slecht:

"Maak een website voor mij"

Goed:

"Maak een landingspagina voor een fictief koffiemark genaamd 'Polderbean'. De pagina moet bevatten: hero met tagline en CTA-button, drie product-cards (espresso, latte, cold brew), testimonials carousel met 3 reviews, FAQ accordion met 4 items. Stijl: warm, minimalistisch, kleuren bruin/cream/donker oranje. Gebruik Tailwind CSS en Shadcn-componenten. Geef alles in één React component."

Het verschil: specificiteit. Tech stack, kleuren, secties, stijl, output-format.

## Het prompt-framework

Robuust template voor consistente output:

1. **RoI** — "Je bent een Senior Frontend Developer met expertise in Next.js"
2. **Context** — "Ik bouw een coffee-merk landing page voor een studieproject"
3. **Taak** — "Genereer een Tailwind component voor de hero-sectie"
4. **Constraints** — "Gebruik Shadcn Button + Card. Geen externe libs. Max 80 regels."
5. **Output-formaat** — "Geef in TypeScript + Tailwind. Code-only, geen uitleg."
6. **Voorbeelden** (optioneel) — "Zoals deze: ..."

Niet alles is altijd nodig — maar hoe complexer je vraag, hoe meer onderdelen.

## Iteratie

Eerste antwoord is zelden de eindstaat. Iteratie patterns:

- "Maak de hero groter en voeg een achtergrondafbeelding toe"
- "Vervang de testimonials carousel met een statische grid"
- "Refactor naar een component-bestand per sectie"

Behandel AI als een collega — geef feedback, vraag verbeteringen.

## Tips uit de praktijk

- **Geef constraints aan** — geen library X, max Y regels, in TypeScript
- **Vraag om uitleg** als je iets niet snapt
- **Vraag om alternatieven** — "geef 3 versies van deze sectie"
- **Geef voorbeelden** van wat je wilt (positief) en wat je niet wilt (negatief)

## 8. v0.dev — van prompt naar UI

**v0.dev** (van Vercel) is een AI-tool die UI-componenten genereert in React + Tailwind + Shadcn.

### Hoe werkt het

1. Ga naar v0.dev (gratis tier beschikbaar)
2. Type een prompt (of plak vanuit ChatGPT)

3. v0 genereert een werkende component, live preview
4. Itereer ("maak het oranje", "voeg buttons toe", "fix de mobile view")
5. Copy de code, of deploy direct naar Vercel

## Wat maakt v0 sterk

- **Stack-aware** — output is direct werkende code in moderne stack
- **Visueel + code samen** — zie live wat je krijgt, niet alleen tekst
- **Component library** — Shadcn is industry-standard
- **Deploy in 1 klik** — vanuit v0 direct live op Vercel-subdomain

## Wanneer NIET v0

- Logica-zware features (forms met validatie, complex state) → liever Cursor of zelf
- Backend / API routes → niet v0's terrein
- Custom design systems → v0 leunt sterk op Shadcn

Voor landing pages, marketing sites, snelle prototypes: v0 is goud waard.

## 9. De slimme workflow — schets naar website

Voor de lesopdracht gebruiken we deze workflow. Het is een blauwdruk die je in elke creatieve flow kunt toepassen.

```
1. TEKEN je idee op papier
  ↓
2. Maak een FOTO van je tekening
  ↓
3. Upload naar ChatGPT + beschrijf je wensen
  ↓
4. ChatGPT maakt een gedetailleerde PROMPT
  ↓
5. Plak die prompt in v0.dev → Website!
  ↓
6. DEPLOY naar Vercel
```

## Waarom dit werkt

- **Tekenen** dwingt je tot duidelijkheid — wat staat waar, wat is groot, wat klein
- **Foto + ChatGPT** = AI snapt je intentie inclusief layout
- **ChatGPT genereert prompt** = beter dan zelf typen (ChatGPT kent v0's voorkeuren)
- **v0.dev** = de visuele uitvoering
- **Vercel** = live op het web in 1 minuut

## Tips voor de tekening

- Werk groot — A4 met dikke marker
- Geef labels (HERO, FEATURES, CTA)
- Schets minstens 3 secties
- Hoeft niet mooi! Hoeft niet to scale!

## Tips voor de ChatGPT-stap

Schrijf erbij:

- Wat is het product / merk
- Doelgroep
- Sfeer (modern, speels, premium, donker, ...)
- Tech: "Genereer een v0.dev-prompt"

ChatGPT zet jouw tekening + beschrijving om in een gedetailleerde v0.dev-prompt. Plak die in v0, voilà.

## 10. Ethiek en valkuilen

### Auteursrecht

AI-modellen zijn getraind op het hele internet — inclusief auteursrechtelijk beschermd materiaal. Wat betekent dat voor wat je maakt?

- **Output is doorgaans van jou** — je hebt de prompt geschreven, je beslist wat te gebruiken
- **Geen plagiaat-claim 100% veilig** — als output sterk lijkt op bestaand werk, kan dat een probleem zijn
- **Vermijd** — namaak van bestaande logo's, specifieke artist-stijl waar de artist nog leeft, bekende karakters

In Nederland: rechtspraak is nog in ontwikkeling. Wees voorzichtig met commerciële uitingen.

### Bias

Trainingsdata weerspiegelt de wereld zoals 'ie is — inclusief bias. AI-output kan:

- Stereotypen versterken
- Onevenredig veel mannen / blanke mensen tonen in beelden
- Minder accuraat zijn voor underrepresented dialecten

Je kunt actief tegen-prompten. Maar wees bewust dat het er zit.

### Transparantie

Vermeld dat je AI hebt gebruikt:

- In school-opdrachten (waar relevant per opdracht-eis)
- In professionele contexten waar dit verwacht wordt
- In creatieve output als de stijl AI-typisch is



---

Heimelijk AI-werk presenteren als 100% eigen werk = academisch + ethisch problematisch.

## Privacy

Wat je in ChatGPT typt kan gebruikt worden voor verbetering van het model (default in gratis tier). Voor gevoelige info:

- Gebruik Tijdelijke Chat
- Of: privacy-vriendelijke alternatieven (Claude — opt-out by default, of lokale modellen)
- Nooit klant-data, medische gegevens, paspoortnummers, wachtwoorden

## Bronnen

- **OpenAI Tokenizer:** <https://platform.openai.com/tokenizer>
- **ChatGPT:** <https://chat.openai.com>
- **Claude:** <https://claude.ai>
- **v0.dev:** <https://v0.dev>
- **Vercel:** <https://vercel.com>
- **Lex Fridman podcast — Sam Altman:** uitleg over hoe LLMs werken
- **3Blue1Brown — Neural Networks playlist** (YouTube): visuele uitleg van neural nets
- **Karpathy — "Let's build GPT" (YouTube):** programmeer een mini-GPT from scratch