# Les 2

## OpenCode

AI als pair-programmer in de terminal

Intro

# Een nieuwe manier van werken

## Wat we niet meer doen

- Browser-AI tools (v0.dev, chat-UIs)
- Copy/paste tussen browser + editor
- AI buiten je echte project
- Eén chat, geen zicht op je code

## Wat we vandaag doen

v Lokaal werken in echt project

v AI direct in je editor

v Lokale dev-server toont meteen

v Plan → Build → Accept per file

**Browser-AI blijft nuttig voor snelle UI-experimenten.**

Voor echt bouwen: lokaal + AI-pair in je terminal.

Tool: OpenCode — open-source, multi-model, gratis.

Planning

# Vandaag

Welkom + nieuwe manier van werken

AI-CLI landschap: Claude Code, Codex, OpenCode

Introductie OpenCode

**LIVE DEMO — scroll-storytelling landing page**

Online zetten: git init → GitHub → Vercel

**Pauze ☕**

**Hands-on opdracht**

Resultaten + afsluiting

Vandaag

# Drie dingen

1

## AI-coding-CLI landschap

Claude Code vs Codex vs OpenCode — verschillen + wanneer kies je welke

2

## LIVE bouwen: scroll-storytelling landingpage

Vanuit een starter-zip — Lenis, GSAP, parallax, micro-animaties

3

## Online zetten — from scratch

git init → nieuwe GitHub repo → nieuw Vercel project → live URL

**Aan het einde: jouw eigen live URL, gepubliceerd via je eigen repo + Vercel project.**

Theorie 2

# Drie spelers — Claude Code, Codex, OpenCode

| Claude Code | Codex | OpenCode |
| --- | --- | --- |
| Anthropic | OpenAI | Open-source |
| Safety-first, vraagt approval, beste raw capability | Autonoom, token-efficient, minder safety-rails | BYO-model, mooiste TUI, MIT-licensed |

**Allemaal terminal-based agents. Geen IDE nodig. Schrijven direct in je bestanden.**

**Verschillen: filosofie, prijsmodel, model-keuze, autonomie.**

Bron: NeuralNine — Claude Code VS Codex VS OpenCode (2026)

Theorie 2

# Vergelijking op 7 assen

| As | Claude Code | Codex | OpenCode |
|---|---|---|---|
| Safety / approval | WINNAAR | autonoom | config |
| TUI / aesthetics | ok | ok | WINNAAR |
| Model-flexibiliteit | Anthropic | OpenAI | WINNAAR |
| Undo / rewind | WINNAAR | geen undo | rewind |
| Token-efficiency | veel | WINNAAR | varieert |
| Open-source | nee | nee | WINNAAR |

**Praktijk-benchmark uit de video: bouw een Sudoku-game**

Claude Code, Codex en OpenCode-met-GPT — vergelijkbare kwaliteit.

OpenCode op gratis open-source modellen (Kimi K2.5, DeepSeek) — duidelijk minder.

Theorie 2

# Waarom OpenCode in deze cursus?

1

**Gratis te beginnen**

Kimi K2.5 Free of lokaal Ollama. Geen creditcard.

2

**Open-source**

Volledig MIT. Je kunt zien hoe deze tools werken.

3

**BYO-model**

Later switchen naar Claude/GPT? 30 seconden werk.

4

**Beste TUI**

Onderdeel van een goede flow is een interface die niet stoort.

**Eindconclusie NeuralNine:**

"OpenCode is filosofisch mijn favoriet. Voor zwaar engineering-werk: Claude Code met Opus."

**Voor jullie: start met OpenCode + Kimi. Upgrade wanneer een werkgever betaalt.**

Theorie 3

# Wat is OpenCode?

Open-source CLI coding agent. Geinstalleerd via npm, draait in je project-folder.

## Kernfeatures:

| | | |
|---|---|---|
| 1 | **Multi-model** | OpenAI, Anthropic, Google, Ollama (local) |
| 2 | **Tool calling** | Files lezen/schrijven, terminal, tests |
| 3 | **AGENTS.md** | Markdown file = project-context |
| 4 | **Plan mode** | AI plant eerst, jij keurt af/goed |
| 5 | **Open-source** | Gratis, audit-baar, geen lock-in |

Theorie 3

# OpenCode vs Cursor vs Claude Code

| OpenCode | Cursor | Claude Code |
| --- | --- | --- |
| **Type:** | **Type:** | **Type:** |
| CLI | Full IDE | CLI |
| **Cost:** | **Cost:** | **Cost:** |
| Gratis* | $20/mnd | $20/mnd |
| **Model:** | **Model:** | **Model:** |
| Elk model | Cursor + own | Alleen Claude |
| **Best:** | **Best:** | **Best:** |
| Backend, scripts | Frontend, IDE-flow | Diepe code-tasks |

* OpenCode: gratis tool, je betaalt alleen voor API-credits (~$5 OpenRouter is ruim).

Setup

# Installeren + eerste keer opstarten

### Installeer (eenmalig):

```
npm install -g opencode-ai
```

### API key (aanrader: OpenRouter - alle modellen, een key):

```
# 1. Maak account op openrouter.ai + $5 credit
# 2. Login in OpenCode:
opencode auth login
```

### Starten in je project:

```
cd ~/projects/mijn-app
opencode
```

Terminal opent een chat-interface. Type je vraag - AI bewerkt files direct.

LIVE DEMO

# Scroll-storytelling landing page

Project: Polderwave — AI infrastructure for builders

## Wat we bouwen:

- Smooth scroll over de hele pagina (Lenis)
- Hero met grote naam die op scroll subtiel parallax-beweegt
- Features sectie met 4 cards — staggered fade-in (GSAP ScrollTrigger)
- Sticky horizontal scroll-sectie met 4 storytelling panels
- Parallax-blobs — drie kleurrijke achtergrond-cirkels op verschillende snelheid
- Micro-interacties: hover-scale op cards, sticky CTA met color transition

**Vijf prompts. Vijf upgrades. Geen handmatige code geschreven.**

Lenis voor smooth scroll. GSAP voor animaties. Tailwind voor styling.

LIVE DEMO

# Flow — slide blijft op scherm

| | | |
|---|---|---|
| STAP 1 | **Starter uitpakken** | `unzip + npm install + npm run dev` |
| STAP 2 | **OpenCode openen** | `Desktop App → Open Folder → Kimi K2.5 Free` |
| STAP 3 | **PROMPT 1** | `Smooth scroll + Hero` |
| STAP 4 | **PROMPT 2** | `Features sectie + staggered reveals` |
| STAP 5 | **PROMPT 3** | `Sticky horizontal storytelling` |
| STAP 6 | **PROMPT 4** | `Parallax-blobs` |
| STAP 7 | **PROMPT 5** | `Micro-interactions (hover, CTA)` |
| STAP 8 | **Deploy** | `git push → Vercel auto-deploy` |

LIVE DEMO

# De 5 prompts — copy-paste vriendelijk

1

### Smooth scroll + Hero

lenis + gsap installeren, POLDERWAVE hero in extreem groot serif font, LenisProvider client component

2

### Features sectie

4 cards in grid, staggered fade-in op scroll via GSAP ScrollTrigger

3

### Sticky horizontal scroll

4 panels horizontaal op scroll, ScrollTrigger pin + horizontal translate

4

### Parallax-blobs

3 gekleurde blobs op fixed achtergrond, mix-blend-screen, verschillende y-translate-snelheden

5

### Micro-interactions

sticky CTA met hover-scale + color transition, hover-pop op cards, subtle parallax op hero-titel

LIVE DEMO

# Tijdens de demo — wat te benoemen

| | |
|---|---|
| **Plan eerst → Tab → Build** | Veilige flow: zien wat er gaat gebeuren voor accepteren |
| **Refresh na elke prompt** | Het magische moment is wanneer studenten de pagina zien groeien |
| **Eén prompt = één commit** | Goede git-hygiëne én duidelijke history |
| **Bij build-error** | Plak de error in OpenCode, vraag om fix |
| **Bij schokkerige animatie** | "Maak smoother met ease-out 1.2s" |
| **Bij mobile-layout breekt** | "Fix de mobile layout voor de features sectie" |

Workflow

# Online zetten — from scratch

**1**

**GIT INIT**

lokaal repo + 1e commit

**2**

**GITHUB**

nieuwe lege repository

**3**

**REMOTE + PUSH**

remote add + push origin

**4**

**VERCEL**

nieuw project + import + deploy

```
# Stap 1 — lokaal
git init && git add . && git commit -m "Initial Polderwave"
git branch -M main

# Stap 3 — na GitHub repo aanmaken
git remote add origin https://github.com/JOUW/polderwave.git
git push -u origin main

# Stap 4 — vercel.com → Add New → Import → Deploy
# Daarna: git push  =  automatische re-deploy
```

Workflow

# Vercel preview deployments per branch

## Elke branch = eigen URL:

| | | |
|---|---|---|
| `main` | `polderwave.vercel.app` | productie |
| `feature/extra-blob` | `polderwave-git-feature-extra-blob.vercel.app` | preview |
| `chore/typo` | `polderwave-git-chore-typo.vercel.app` | preview |

### Waarom handig:

- Test wijzigingen zonder productie te raken
- Deel preview-link met klant/team voor feedback
- Automatisch — geen configuratie nodig
- Stack: één PR per feature, één preview-URL per PR

HANDS-ON

# Bouw je eigen scroll-landing page

## Stappen:

1. Unzip opencode-ui-starter.zip uit lesbestanden → npm install → npm run dev
2. Open OpenCode Desktop App → Open Folder → Kimi K2.5 Free
3. Bouw minimaal 3 van de 5 demo-prompts na (of eigen variant)
4. Eén prompt per taak — refresh browser tussen elke prompt
5. Commit + push naar GitHub na elke feature
6. Check Vercel: live deploy bekijken
7. Maak optioneel een feature-branch voor preview-URL

**Vuistregel — één prompt per taak.**

Niet: 'Maak een complete landing page met alles'.

Wel: 'Voeg hero toe' → check → 'Voeg features grid toe' → check → ...

**Resultaat: live deploy URL + GitHub URL in Teams delen.**

Afsluiting

# Samenvatting + Huiswerk

## Vandaag gezien:

- Het CLI-landschap: Claude Code, Codex, OpenCode
- Plan vs Build mode + @explore
- Live demo: scroll-storytelling met Lenis + GSAP (5 prompts)
- git push → Vercel auto-deploy + preview per branch

## Huiswerk:

1. Bouw eigen scroll-landing page (eigen merknaam, eigen kleurpalet)
2. Minimaal 3 van de 5 demo-features in je eigen versie
3. Bonus: voeg één eigen animatie toe (dark mode, cursor-trail, etc.)
4. Inleveren via Teams: Vercel URL + GitHub URL + korte beschrijving

**Tip: deploy meermaals tijdens het bouwen.**

Elke feature = git commit + push = Vercel auto-deploy. Zie progressie live.

## Volgende les

# Les 3 - Cursor Basics

### Van CLI naar volledige IDE:

- Cursor - VS Code fork met AI overal verweven
- Chat (Cmd+L) / Inline Edit (Cmd+K) / Composer (Cmd+I) / Tab
- @-mentions voor context
- .cursorrules - project-conventies
- Debug-flow met AI

**Voorbereiden:**

Download Cursor van cursor.com - student plan = gratis Pro.

## Vragen?

Tot volgende week!