

# Les 2 — OpenCode Pro

Rules · Worktrees · Een scroll-animatie site bouwen

<b>Vak:</b>	AI-Assisted Development
<b>Opleiding:</b>	NOVI Hogeschool Utrecht
<b>Klas:</b>	Klas B (online, 2 uur)
<b>Onderwerp:</b>	OpenCode Desktop · AGENTS.md · worktrees · GSAP/Lenis

## Leerdoelen

- Begrijpen waarom we OpenCode kiezen i.p.v. Cursor
- Plan/Build mode en @explore subagent inzetten
- AGENTS.md schrijven (officieel format)
- opencode.json config met permissions
- opencode-worktree plugin installeren via ocx
- Parallel agents via Sessions sidebar
- Next.js 16 + GSAP + Lenis scroll-site opzetten

## Inhoud

1. OpenCode in 2026 — wat moet je weten?
2. TUI vs Desktop — welke kies je?
3. Plan/Build/@explore + /init
4. AGENTS.md — het officiële regelbestand
5. opencode.json — config & permissies
6. Plugins & opencode-worktree
7. De scroll-animatie stack (Next.js 16)
8. Project setup van A tot Z
9. Workflow patterns
10. Bronnen

## 1

# OpenCode in 2026

OpenCode is een open-source AI coding-IDE. 60.000+ GitHub stars. Gemaakt door SST.

Provider-agnostisch — werkt met OpenAI, Anthropic, Google, Groq, lokale modellen via Ollama, en meer.

Vergelijkbaar concept als Cursor of Claude Code, maar:

- **Open source** — geen vendor lock-in
- **Provider-keuze** — niet vast aan één AI-bedrijf
- **Script-vriendelijk** — perfect voor automation, CI/CD, plugins
- **Twee smaken:** Desktop (UI) en TUI (terminal)

## Desktop App (vandaag onze keuze)

Native app voor macOS, Windows, Linux. Download via [opencode.ai/download](https://opencode.ai/download).

Layout (globaal):

- **File tree** links — bestanden + folders, right-click voor acties
- **Diff viewer** midden — visueel groen/rood met klik-accept/reject
- **Chat panel** onder — prompts naar de agent
- **Sessions sidebar** links — meerdere parallele sessies (voor worktrees!)
- **Ingebouwde terminal** — onderaan, optioneel

## TUI (terminal)

```
npm i -g opencode-ai@latest  
opencode
```

Zelfde engine, zelfde commando's, zelfde plugins. Goed voor SSH, scripts, CI/CD.

## 2

## TUI vs Desktop

Aspect	TUI	Desktop
Leercurve	Steiler — alles via keyboard	Vlakker — visueel
Diff review	Plain text inline	Visuele diff-viewer
Files browsen	Via @-references	File tree zichtbaar
Sessies parallel	Tmux / aparte terminals	Sessions sidebar
SSH / headless	Ja	Nee
Worktree auto-spawn	Ja	Nee (handmatige tab)
Scripted workflows	Ja	Beperkt

### Wanneer wel TUI?

- **SSH-only servers** — geen X11 nodig
- **CI/CD pipelines** — scripted invocaties
- **Tmux fans** — past in je tmux setup
- **Worktree power-users** — auto-terminal-spawn van de plugin

### Onze keuze voor deze les en eindopdracht

Desktop. Sneller te leren, visueller, beter voor parallel-werken via Sessions sidebar.

### 3

## Plan/Build/@explore + /init

### Plan vs Build mode

OpenCode heeft twee modes — Tab om te wisselen.

**Plan mode** is read-only. Tools: read, grep, glob, webfetch. Geen edits, geen shell. Goed voor brainstorm / verkenning.

**Build mode** is default. Alle tools aan. Schrijft code, voert commands uit.

#### Best practice

Begin altijd in plan-mode. Laat het uitleggen wat het van plan is. Geef feedback. Pas dan Tab → build.

### Sub-agents

Naam	Wat	Wanneer
@explore	Read-only verkenning	'Waar staat X?' 'Wat is dit?'
@general	Brede taak-agent	Default voor de meeste taken
@scout	Gerichte zoekopdracht	'Vind alle plaatsen waar...'

### /init commando

In de chat: `/init`. OpenCode scant je repo, stelt vragen, en genereert/update je AGENTS.md.

Doe dit als eerste actie in elk nieuw project.

## 4 AGENTS.md — het officiële regelbestand

AGENTS.md is markdown in je project-root die OpenCode automatisch in elke context laadt. Officieel format. Expliciet vergeleken in de docs met Cursor's .cursorrules.

### Twee locaties

Pad	Wanneer
./AGENTS.md	Per project (in repo, geversioneerd)
~/.config/opencode/AGENTS.md	Globaal voor alle projecten

### Externe regels referenzen

```
# Stack rules
@rules/styling.md
@rules/security.md
```

OpenCode laadt deze lazy via z'n Read-tool — pas wanneer relevant.

### Voorbeeld voor scroll-animatie project

```
# Project Rules

## Why this stack
High-end scroll storytelling (Awwwards/FWA niveau).
GSAP + Lenis = pro standaard – Apple, Stripe, OpenAI,
Active Theory, Locomotive, Darkroom Engineering.

## Stack
- Next.js 16 (App Router, TypeScript, Turbopack)
- TailwindCSS · GSAP 3.15+ · Lenis 1.3+

## Hard rules
- Geen Framer Motion, react-spring, AOS
- useGSAP(() => {}, { scope: ref }) – nooit useEffect
- Lenis sync: gsap.ticker.add met autoRaf: false
- Next 16: await params, await cookies()

## Done =
- Geen hydration warnings
- ScrollTrigger.refresh() na dynamische content
```

#### De Why-sectie is cruciaal

Hiermee snapt de agent waarom we deze keuzes maken — niet alleen wat. Resultaat: hij stelt op eigen initiatief geen Framer Motion meer voor.

## 5 opencode.json — config & permissies

### Locaties

Pad	Scope
./opencode.json	Per project
~/.config/opencode/opencode.json	Globaal

### Basis voorbeeld

```
{
  "$schema": "https://opencode.ai/config.json",
  "model": "openai/gpt-4o-mini"
}
```

Model formaat: `provider/model-id`. Voorbeelden:

- `openai/gpt-4o-mini`
- `openai/gpt-4o`
- `anthropic/claude-sonnet-4`
- `groq/llama-3.3-70b-versatile`

### Permissies

```
{
  "permission": {
    "bash": {
      "*": "ask",
      "git *": "allow",
      "rm *": "deny"
    }
  }
}
```

Drie levels: **allow**, **ask**, **deny**. Glob patterns voor bash én files.

## 6

## Plugins & opencode-worktree

Plugin van **kdcokenny** (community maker, ook achter ocx en opencode-workspace).

### Wat doet het?

- **worktree\_create** — maakt nieuwe branch + worktree-folder
- **worktree\_delete** — commit changes + cleanup

Worktree-locatie: `~/.local/share/opencode/worktree/<project>/<branch>/`

### Installatie via ocx

```
# Eenmalig: ocx installeren
curl -fsSL https://kdco.dev/ocx/install.sh | sh

# Plugin toevoegen
ocx add kdco/worktree --from https://registry.kdco.dev
```

Restart OpenCode na install.

### Verschil Desktop vs TUI

Actie	TUI	Desktop
worktree_create aanroepen	Ja	Ja
Auto nieuw terminal venster	Ja	Nee
Nieuwe context openen	Auto	Handmatig (Sessions sidebar)

### Workflow in Desktop

1. In hoofd-Session: **'Maak een worktree voor feature-hero.'**
2. Plugin maakt worktree-folder
3. Klik **+ New Session** in sidebar
4. **Open Folder** → navigate naar de worktree-folder
5. Nu heb je 2 sessies parallel — main + feature-hero
6. Klik tussen sessies om te wisselen

## 7

## De scroll-animatie stack

### Next.js 16 (mei 2026)

Uitgekomen oktober 2025. Breaking changes ten opzichte van 15:

- **Async API's verplicht:** `cookies()`, `headers()`, `params`, `searchParams` zijn nu async — altijd `await`
- **Turbopack default** voor dev en build
- `middleware.ts` → `proxy.ts`
- **Node.js 20.9+** vereist

### Waarom GSAP + Lenis?

Dit is dé stack voor scroll storytelling op pro-niveau. Niet 'een' optie — de standaard.

Wie gebruikt dit?

- **Apple** — product pages met scroll-revealed content
- **Stripe** — homepage en product pages
- **OpenAI** marketing pages
- **Awwwards / FWA winnaars** — vrijwel altijd GSAP
- Pro studios: **Active Theory**, **Locomotive**, **Resn**, **Darkroom Engineering**

#### Waarom niet Framer Motion?

Framer Motion is uitstekend voor app UI — modals, transitions.  
Maar voor scroll storytelling wil je timing-precisie en GPU-perf.  
Daarom: Framer Motion voor app-werk, GSAP voor scroll-werk.

### Packages

```
npm install gsap @gsap/react lenis
```

**Belangrijk:** pakket heet `lenis` (niet meer `@studio-freight/lenis`).



## 8 Project setup van A tot Z

### Step 1 — Next.js 16 project

```
npx create-next-app@latest scroll-site \
  --typescript --tailwind --app --eslint --no-src-dir --turbo
cd scroll-site
```

### Step 2 — Dependencies

```
npm install gsap @gsap/react lenis
```

### Step 3 — Git init

```
git init
git add . && git commit -m "init: next 16 + gsap + lenis"
```

### Step 4 — AGENTS.md aanmaken

Plak het voorbeeld uit hoofdstuk 4 in AGENTS.md in root.

### Step 5 — opencode.json

```
{
  "$schema": "https://opencode.ai/config.json",
  "model": "openai/gpt-4o-mini",
  "permission": {
    "bash": {
      "": "ask", "git ": "allow",
      "npm ": "allow", "rm ": "deny"
    }
  }
}
```

### Step 6 — Open in Desktop, run /init

File → Open Folder → scroll-site. In chat: /init.

## Workflow patterns

### Pattern 1 — Plan eerst, dan Build

Begin altijd in plan-mode. Laat de agent uitleggen wat het van plan is. Geef feedback. Pas dan Tab → build.

### Pattern 2 — Eén worktree per feature

Werk niet alles in main. Een feature = een worktree = een branch. Makkelijk parallel + reviewen via PR.

### Pattern 3 — AGENTS.md evolueert

Loop je tegen iets aan dat steeds verkeerd gaat? Voeg toe aan AGENTS.md. Het is een levend document.

### Pattern 4 — @explore voor verkenning

Nieuwe codebase of vergeten waar iets staat? @explore eerst, dan pas hoofd-agent inzetten. Spaart tokens.

### Pattern 5 — Permissions matchen je risico

Beginnersproject: bash.\* op ask. Productie-werk: alleen specifieke folders editable.

### Pattern 6 — Sessions sidebar = parallel werk

Eén Session = orchestrator (in main). Worktrees = workers. Orchestrator stuurt, workers bouwen.

## 10 Bronnen

### OpenCode (officieel)

- Docs: [opencode.ai/docs/](https://opencode.ai/docs/)
- Download: [opencode.ai/download](https://opencode.ai/download)
- Rules (AGENTS.md): [opencode.ai/docs/rules/](https://opencode.ai/docs/rules/)
- Config: [opencode.ai/docs/config/](https://opencode.ai/docs/config/)
- Plugins: [opencode.ai/docs/plugins/](https://opencode.ai/docs/plugins/)
- GitHub: [github.com/sst/opencode](https://github.com/sst/opencode)

### Plugin ecosystem

- opencode-worktree: [github.com/kdcokenny/opencode-worktree](https://github.com/kdcokenny/opencode-worktree)
- ocx extension manager: [github.com/kdcokenny/ocx](https://github.com/kdcokenny/ocx)
- opencode-workspace: [github.com/kdcokenny/opencode-workspace](https://github.com/kdcokenny/opencode-workspace)

### Next.js 16

- Blog: [nextjs.org/blog/next-16](https://nextjs.org/blog/next-16)
- Upgrade guide: [nextjs.org/docs/app/guides/upgrading/version-16](https://nextjs.org/docs/app/guides/upgrading/version-16)

### GSAP & Lenis

- GSAP docs: [gsap.com/docs/v3/](https://gsap.com/docs/v3/)
- GSAP React (useGSAP): [gsap.com/resources/React/](https://gsap.com/resources/React/)
- GSAP License (gratis sinds 2025): [gsap.com/standard-license](https://gsap.com/standard-license)
- Lenis: [github.com/darkroomengineering/lenis](https://github.com/darkroomengineering/lenis)

#### Volgende les preview

Les 3: Introductie Cursor — de commerciële tegenhanger. Cursor rules vs AGENTS.md, Composer, @-references, wanneer welke tool.