

Les 3 — Docenttekst (normaal)

Fysieke les. Voor uitgebreide voorbereiding + context per blok. Voor letterlijk voorleesbaar script: zie `Les03-Docenttekst-Autocue.md`.

Lesvorm en flow

Onderdeel	Indeling
Theorie + demo	±1,5 uur klassikaal
Pauze	15 minuten
Hands-on	±1,25 uur zelf bouwen met begeleiding

Vorbereiding — een week vooraf

Studenten checklist (stuur via Slack):

- Laptop met admin-rechten (Cursor installatie vereist admin op Windows)
- NOVI e-mailadres werkend
- `node -v` werkt (versie 18 of hoger)
- `git --version` werkt
- Browser-bookmark voor `cursor.com/students` ergens klaarzetten

Jouw eigen voorbereiding:

- Test je eigen Cursor Pro Student status (om geen "ik vergat hem te activeren" te krijgen tijdens demo)
- Test `npx create-next-app@latest` (cache kan blokkeren — `npm cache clean --force` als nodig)
- Backup-project klaar met Hero al gebouwd voor het geval live demo vastloopt
- `.cursorrules` snippetet copy-paste klaar

Vorbereiding — dag van de les

30 minuten vooraf:

- Cursor open in een lege folder voor de demo
- Twee terminals klaar (één in `~/web`, één leeg)
- Browser tabs: `cursor.com/students`, `cursor.com/download`, `github.com` (logged in)
- Slides in fullscreen op presentatiescherm

Backup scenario:

- SheerID verificatie geeft soms 5–10 minuten vertraging → vertel studenten dit vooraf
- `npx create-next-app` kan hangen → check cache of gebruik `bun create-next-app` als alternatief
- Internet traag in zaal → backup-project op standby

Per blok — wat je extra wilt zeggen

Blok 1 — Welkom + Wat is Cursor (slides 1–3)

Doel: verwachtingen zetten, energie inschakelen.

Vraag vooraf: wie heeft GitHub Copilot wel eens gebruikt? Wie ChatGPT voor code? Dat is je startpunt — Cursor is dat, maar nauwer geïntegreerd in de editor zelf.

Blok 2 — Student Plan + Installatie (slides 4–6)

Doel: iedereen heeft Pro Student actief aan het eind van dit blok.

Vooraf melden: SheerID-verificatie kan 1–10 minuten duren. Geen reden om paniek te krijgen.

Common issue: studenten met persoonlijk Gmail-adres in plaats van NOVI. Verificatie faalt → "log in met je NOVI-mail" — dat is non-negotiable.

Voor wie technisch slim is: Cursor heeft een `settings.json` zoals VS Code. Importeer de jouwe als je hem hebt. Maakt de transitie soepel.

Blok 3 — Skills/Docs + Eerste project (slides 7–9)

Doel: student heeft `mijn-eerste-cursor-app` draaiend op localhost en open in Cursor.

Verwacht problemen:

- "I get a permission error" → npm prefix issue, gebruik `nvm` of `corepack`
- "create-next-app duurt eeuwig" → cache + slecht internet, even geduld
- "ik krijg geen Tailwind" → student kiest "no" bij Tailwind-prompt, opnieuw beginnen

Loop rond in dit blok — dit is waar studenten het meest vastlopen.

Blok 4 — Drie sneltoetsen (slides 10–14)

Doel: student kan Tab, Cmd+K en Cmd+L actief gebruiken.

Voor elke sneltoets: doe een mini-live-demo. Niet alleen praten — laat letterlijk zien wat er gebeurt. Studenten leren dit door te zien.

Bij Tab: typ een commentaar en wacht 2 seconden — Cursor stelt de hele functie voor. Dat is een wow-moment.

Bij Cmd+K: kies bewust iets simpels (zoals een variabele hernoemen) — niet iets complex dat 10 seconden duurt.

Bij Cmd+L: stel een vraag, niet meteen een grote taak. "Hoe werkt `useEffect`?" — laat Cursor antwoorden zien.

Blok 5 — `.cursorrules` (slide 15)

Doel: student weet wat het is + heeft één in zijn project.

Vertel ook: in Les 14 (veel later) gaan we naar `.cursor/rules/` (folder met meerdere bestanden). Voor nu is het oude format prima en simpeler om uit te leggen.

Demo: ik vraag Cursor in `Cmd+L` "*Maak een `.cursorrules` voor dit Next.js project*". Laat de output zien, plak in `.cursorrules`-bestand, save.

Blok 6 — Hands-on (slides 16–19)

Doel: elke student heeft een werkende Hero gebouwd en één commit gedaan.

Voor de pauze: demonstreer taak 1 + 2. Niet meer.

Na de pauze: loop rond. Voor elke vraag, beantwoord met "wat zou je in Cursor zelf vragen?" — leer ze om de IDE te benutten.

Verwacht vragen:

- "Mijn Hero ziet er saai uit" → vraag Cursor om een tweede iteratie via `Cmd+K`
- "Tailwind classes werken niet" → check of `globals.css @tailwind base/components/utilities` heeft
- "Git werkt niet" → check `gh auth login` of `git config user.email`

Blok 7 — Afronding (slide 20)

Doel: koppel vandaag aan de rest van de opleiding.

Maak expliciet: vandaag was de basis. `Tab + Cmd+K + Cmd+L` gaan jullie elke les terugzien. Les 4 gaat over TypeScript. Les 14 (veel later) gaat dieper op Cursor 3 features.

Veelvoorkomende vragen + antwoorden

"Werkt Cursor op Linux?" Ja — AppImage download via cursor.com.

"Wat als ik geen Pro Student krijg?" Stuur student naar support@cursor.com met NOVI mail + bewijs van inschrijving. Werkt meestal binnen 24 uur.

"Mag ik mijn eigen OpenAI/Anthropic key gebruiken?" Ja — Settings → Models → eigen key plakken. Soms goedkoper bij weinig gebruik, niet nodig met Pro Student.

"Wat is het verschil tussen `Cmd+K` en `Cmd+L`?" `Cmd+K` = inline edit op selectie (één bestand, één gericht ding). `Cmd+L` = chat (uitleg vragen of grotere taak met meerdere bestanden).

"Wordt mijn code naar OpenAI/Anthropic gestuurd?" Standaard: ja, voor inference. Cursor heeft een Privacy Mode in Settings — als die aan staat, wordt code niet bewaard. Geen training op je code.

"Wat als ik door 500 requests heen ben?" Wacht op nieuwe maand (reset op de 1e), of upgrade naar betaalde tier (\$20/maand), of plug je eigen API-key in.

"Kan ik blijven coden zonder internet?" `Tab` + lokale modellen via Settings, ja. `Cmd+K` en `Cmd+L` vereisen internet.

Foutmodi tijdens de les

Probleem	Snelle fix
SheerID hangt	Volgende student helpen, terugkomen na 10 min
npx create-next-app hangt	npm cache clean --force of bun create next-app
Cursor niet ingelogd na herstart	Settings → Account → opnieuw inloggen
Tab-completion verschijnt niet	Settings → Features → Cursor Tab → check enabled
Cmd+L opent niet	Mogelijk Notion of andere app vangt sneltoets — check System Preferences
.cursorrules lijkt genegeerd	File moet in project-root staan, niet in src/ — herstart Cursor

Vooruitblik Les 4

Volgende les: TypeScript Fundamentals. Strict types, generics, utility types. Studenten gaan ze toepassen in hun Next.js project. Geen extra installatie nodig — Next.js scaffold zit al met TS.

Stuur de avond ervoor een Slack-bericht: "voor les 4: zorg dat je Cursor Pro Student actief hebt, anders sta je achter".