

Les 14 — Docenttekst (normaal)

Online les via Teams. Voor uitgebreide voorbereiding + context per blok. Voor letterlijk voorleesbaar script: zie `Les14-Docenttekst-Autocue.md`.

Lesvorm en flow

Onderdeel	Indeling
Theorie + demo	±1 uur klassikaal (live demo via screenshare)
Pauze	15 minuten
Nu jullie	±1,5 uur zelf doen met begeleiding via chat

Setup tijdens de les:

- Monitor 1 (privé): autocue-document fullscreen
- Monitor 2 (gedeeld in Teams): Cursor + browser + terminal + slides naast elkaar

Tip voor de online-les: zet je Teams-deelvenster zo in dat slides en code allebei zichtbaar zijn. Cursor + terminal samen, slides klein in een hoek.

Voorbereiding — week vooraf

Accounts checken (studenten allemaal):

- GitHub-account met gh CLI ingelogd (`gh auth login`)
- Vercel-account met CLI: `npm i -g vercel && vercel login`
- Cursor geïnstalleerd, Pro of Business plan actief, ingelogd

Communicatie: stuur een Slack-bericht een dag vooraf met deze checklist. Als iemand morgen vastloopt op de installatie ben je een kwartier kwijt.

Voorbereiding — dag van de les

30 minuten vooraf:

- Cursor open in een nieuwe lege folder voor de demo
- Browser tabs: `github.com` (logged in), `vercel.com/dashboard`, `vercel.com/new`
- Twee terminals klaar (één in de demo-folder, één voor ad-hoc commando's)
- Slides in een aparte fullscreen of klein in een hoek
- Autocue-document op tweede monitor

Backup scenario:

- Een eerder voltooide demo-folder klaar voor het geval live demo vastloopt

- Screenshots van Vercel preview-URL's voor het geval Vercel traag deploy't
- Pre-gemaakte `.cursor/rules/general.mdc` en `AGENTS.md` om snel te kunnen plakken

Copy-paste-klaar materiaal:

- Repo-naam (mijn-portfolio of vergelijkbaar)
- Twee feature-prompts (zie autocue Bijlage C)
- `.cursor/rules/general.mdc` inhoud (zie autocue Bijlage A)
- `AGENTS.md` inhoud (zie autocue Bijlage B)

Per blok — wat je extra wilt zeggen

Blok 1 — Welkom + Vandaag (slides 1–2)

Doel: verwachtingen zetten — het is een live demo van begin tot eind.

Maak duidelijk dat het anders is dan eerdere lessen: je bouwt een hele cyclus voor hun ogen op. Vraag tussendoor of het tempo klopt — online is het lastig om non-verbale signalen te lezen. Laat ze hun hand opsteken (Teams reactie) als iets onduidelijk is.

Blok 2 — Scaffold + Git + Vercel (slides 3–7)

Doel: in 15 minuten van leeg naar live productie-URL.

Slide 3 — Scaffold: ga niet door de `create-next-app` prompts heen alsof het triviaal is. Studenten hebben dit vaak nog nooit gedaan. Beschrijf elke keuze kort:

- TypeScript: ja, want strict typing
- ESLint: ja, standaard
- Tailwind: ja, we gebruiken het hele jaar
- `src/` dir: ja, het houdt root schoon
- App Router: ja, dat is de moderne flow
- Turbopack: ja als default, geeft betere DX

Slide 4 — Git: veelgemaakte fout is `git push` zonder eerst `remote add`. Laat zien wat de foutmelding eruitziet en hoe je het oplost. Wijs ook expliciet op `.gitignore` — je toont dat `node_modules` en `.env.local` NIET in de eerste commit zitten.

Slide 5 — Vercel: wacht écht de 45 seconden af. Gebruik die tijd om uit te leggen wat Vercel doet onder de motorkap (`clone`, `install`, `build`, `deploy`). Het wachten levert spanning op — gebruik het didactisch.

Slide 6/7 — URLs en env vars: dit is nog theorie, want we hebben nog geen secrets. Belofte aan studenten: "zo direct ga ik dit ook live gebruiken met Vercel CLI."

Blok 3 — Vercel CLI (slides 8–10)

Doel: student begrijpt het verschil tussen dashboard en CLI, en ziet één concrete workflow (`link + env pull`).

Belangrijk: maak duidelijk dat we GEEN `env vars` echt hebben — we tonen de syntax. Als je het wilt demonstreren met een echte var, voeg er één toe (`MIJN_NAAM=Tim`) en `pull` die naar lokaal, om te laten

zien dat de file gevuld wordt.

`vercel env pull` is een eye-opener voor veel studenten. Het scheelt je manuele gepruts elke keer als productie-vars wijzigen.

Blok 4 — Cursor in detail (slides 11–17)

Doel: student kent de hoofdfeatures en weet wanneer welke te gebruiken.

Deze 7 slides zijn de inhoudelijke kern. Niet door doorheen jakkeren. Geef per modus een micro-demo:

- **Slide 12 (modi):** open de chat met `Cmd+L`, sluit met `Esc`. Druk `Cmd+K` op een regel in code en typ "rename this variable". Toggle Plan-mode in de chat — laat zien hoe de UI verandert.
- **Slide 13 (@-mentions):** typ `@` in chat, laat het autocomplete-menu zien. Probeer `@web:Next.js` `16 release notes` voor een live web-search demo.
- **Slide 14 (Cursor Rules):** maak live een `.cursor/rules/general.mdc` aan. Zet er één duidelijke regel in (bv. "altijd functional components, nooit class components"). Test daarna in een nieuwe chat dat de AI die regel volgt.
- **Slide 15 (AGENTS.md):** plak je voorbereide voorbeeld. Leg uit waarom ook Claude Code en Copilot dit lezen.
- **Slide 16 (Background agents):** demo de feature-2-prompt al hier even kort, om visueel te maken hoe een agent in een aparte tab opent.

Vraag aan het eind van dit blok: "welke modus zou je gebruiken voor X?" — twee of drie scenario's. Kort, voor engagement.

Blok 5 — Live feature-workflow (slides 18–22)

Doel: de hele cyclus laten zien — niet alleen Cursor, niet alleen Vercel, maar de combo.

Slide 19 — Demo 1: plan + apply. Maak het visueel: open de chat, plak de feature-1-prompt, wacht tot het plan verschijnt, lees het door, klik Apply. Dan met `Cmd+K` één tweak. Commit.

Slide 20 — Push + preview: push, switch naar browser tab Vercel, wacht op build, open preview-URL. Gebruik de wachttijd om te zeggen: "kijk, vanaf hier ben je niet meer alleen — je kunt deze URL nu delen met collega's of klanten zonder dat zij iets hoeven te installeren."

Slide 21 — Demo 2 background agent: dit is de WOW-moment van de les. Plak de feature-2-prompt, klik "open in background", en demonstreer LETTERLIJK dat je in main editor blijft werken. Doe een `Cmd+K` in een ander bestand. Dat illustreert het parallel-werken.

Slide 22 — Merge: open beide PR's op github.com. Toon de Vercel-bot comment. Merge één naar `main`. Switch terug naar Vercel dashboard om de productie-deploy te zien starten.

Blok 6 — Recap + Nu jullie (slides 23–25)

Doel: overgang naar zelf-doen.

Voor de pauze: vraag of er nog iets onduidelijk was uit de demo. In een online setting moet je actief uitlokken — studenten typen vaak niet uit zichzelf.

Na de pauze: laat een gedeelde chat open zodat studenten kunnen tonen waar ze vastlopen. Wees actief in de chat. Loop om de 15 minuten een check-in: "hoe ver iedereen?"

Veelvoorkomende vragen + antwoorden

"Waarom Cursor en niet gewoon VS Code met GitHub Copilot?" Beide werken. Cursor's voordeel: tighter integratie van plan/build/agent flows en background agents. VS Code + Copilot is solid voor inline-completion. Voor agent-workflows is Cursor (of Claude Code) doorgaans soepeler. Test allebei en kies.

"Wat kost Cursor?" Hobby tier is gratis met limits. Pro is rond €20/maand. Voor de cursus genoeg.

"Kan ik mijn API-key in Cursor gebruiken in plaats van hun subscription?" Ja — Settings → Models. Voeg je OpenAI/Anthropic key toe. Goedkoper bij weinig gebruik, duurder bij veel.

"Wat als ik een typo maak in .cursor/rules/" Cursor parseert de YAML front-matter, gooit een waarschuwing bij syntax errors. Open de file en kijk linksonder voor de error.

"Zal Cursor mijn code naar OpenAI sturen?" Hangt af van je settings. Cursor heeft "Privacy mode" — code wordt niet bewaard door OpenAI of Anthropic. Check Settings → Privacy.

"Hoe kies ik tussen Plan en Build?" Plan voor onbekende terrein of grote features. Build als je precies weet wat je wilt en het kleiner is. Bij twijfel: Plan, dan klik je Apply als het bevalt.

"Werkt Vercel CLI ook voor andere frameworks?" Ja — Astro, SvelteKit, Vue, Remix, alles wat Vercel ondersteunt. De link- en env-flow is identiek.

"Wat als ik géén Vercel wil gebruiken?" Alternatieven: Netlify (vergelijkbare workflow), Cloudflare Pages (goedkoper bij schaal), zelf hosten op een Hetzner box met Coolify (zoals Tim doet). De Cursor-workflow blijft hetzelfde — alleen de deploy-step verandert.

Foutmodi tijdens de les

Probleem	Snelle fix
gh repo create faalt — niet ingelogd	gh auth login opnieuw
Vercel detecteert geen Next.js	Build directory leeg gelaten, niet out/ typen
Vercel build crasht op env var	Tijdelijk een dummy zetten in dashboard
Cursor verliest context	Esc, nieuwe chat starten, opnieuw @-mention
Background agent doet niets	Eerst zien of het Pro/Business-plan vereist
vercel link koppelt aan verkeerd project	rm -rf .vercel en opnieuw

Vooruitblik Les 15

Volgende les: RAG met embeddings. We bouwen een PDF Q&A-app.; Studenten moeten daarvoor hun Next.js-app van vandaag live hebben staan, met minstens twee PR's gemerged. Anders missen ze het fundament voor de RAG-stack.

Stuur de avond ervoor een Slack-bericht: "checklist voor Les 15 — productie-URL + 2 merged PR's — laat in chat zien als je vastzit."