

Les 3 — Lesstof

Cursor Basics

Vak: AI-Assisted Development **Opleiding:** NOVI Hogeschool Utrecht **Vorige les:** Les 2 — OpenCode

Volgende les: Les 4 — TypeScript Fundamentals

Doel van deze les

Aan het einde van deze les heb je:

- Cursor geïnstalleerd en geactiveerd met je Student Plan
- Skills/Docs geconfigureerd voor Next.js, React, Tailwind
- Een werkend Next.js project opgezet via `npx create-next-app`
- De drie hoofdsneltoetsen (Tab / Cmd+K / Cmd+L) actief gebruikt
- Een `.cursorrules`-bestand in je project
- Een Hero-component, een styling-tweak en één extra feature gebouwd
- Je eerste commit naar Git gedaan

1. Wat is Cursor

Cursor is een **fork van Visual Studio Code** met AI als kernfeature ingebouwd. Dat betekent: alles wat je in VS Code kunt — file explorer, terminal, debugger, extensies — werkt hier ook. Het verschil: AI is geen plug-in maar een onderdeel van de editor zelf.

Werkt met meerdere modellen

Cursor laat je per chat kiezen welk AI-model je gebruikt:

- **Claude** (Opus, Sonnet, Haiku) — sterk in code-redenering
- **GPT** (4, 4o, 5) — solide allround
- **Gemini** (Pro, Flash) — snel en goedkoop
- **Lokaal** (via Ollama) — privacy en kosten-vriendelijk

Waarom relevant

- Industry-standard in AI-assisted development
- Veel vacatures noemen het expliciet
- Gratis Pro Student voor NOVI-studenten

2. Student Plan activeren

Wat krijg je

Tier	Fast requests/maand	Premium models	Tab-completion
Free	50	beperkt	onbeperkt
Pro	500	volledig	onbeperkt
Pro Student	500	volledig	onbeperkt
Business	1000+	volledig	onbeperkt

Pro Student = gewone Pro, maar gratis voor verified studenten.

Hoe activeer je

1. Ga naar `cursor.com/students`
2. Log in met je **NOVI e-mailadres** (niet je persoonlijke Gmail)
3. SheerID verifieert je via NOVI's inschrijfsysteem (1–10 min)
4. Cursor upgrade je account automatisch naar **Pro Student**

Verifiëren

Open Cursor → Settings (Cmd+,) → Tab **General** → check **Account**. Status moet "**Pro Student**" zijn.
Als het "Free" blijft: log opnieuw in via `cursor.com/students` en herstart Cursor.

3. Installeren

```
# 1. Download van cursor.com (kies je OS)
# 2. Installeer als gewone app (sleep naar Applications op Mac,
#    next-next-finish op Windows)
# 3. Open Cursor → Sign in
# 4. Optioneel: importeer VS Code instellingen (klik Ja bij de prompt)
```

Eerste keer Cursor opent vraagt hij:

- VS Code settings importeren? **Ja** (theme, keybindings, extensies blijven)
- Welk model kiezen? Voor nu: laat default staan

4. Skills / Docs configureren

Cursor kan officiële framework-documentatie indexeren. Vervolgens kun je in chat schrijven `@docs:Next.js` en de AI antwoordt met actuele info uit die docs.

Setup

1. Settings → **Features** → **Docs**
2. Klik + **Add new doc**
3. Voeg toe (één voor één):
 - Next.js (URL: <https://nextjs.org/docs>)
 - React (URL: <https://react.dev>)
 - Tailwind CSS (URL: <https://tailwindcss.com/docs>)
1. Wacht tot indexering klaar is (1–2 min per doc)

Gebruiken

In Cmd+L (chat):

```
@docs:Tailwind hoe centreer ik verticaal en horizontaal met flex?
```

Cursor antwoordt met de juiste Tailwind-classes uit de officiële docs.

5. Eerste Next.js project

Scaffolden

```
cd ~/web # of waar jij je projecten houdt
npx create-next-app@latest mijn-eerste-cursor-app
```

Prompts:

Vraag	Aanbevolen	Waarom
TypeScript?	Ja	strict typing voorkomt bugs
ESLint?	Ja	code-style consistency
Tailwind CSS?	Ja	utility-first styling
src/ directory?	Ja	houdt project-root schoon
App Router?	Ja	moderne Next.js flow
Turbopack?	Ja	snellere dev-builds
Import alias @/*?	Ja	@/components/x ipv .././components/x

Starten

```
cd mijn-eerste-cursor-app
npm run dev # → http://localhost:3000
```

Openen in Cursor

```
cursor . # opent huidige folder in Cursor
```

Of: File → Open Folder → kies de map.

6. De drie hoofdsneltoetsen

Sneltoets	Functie	Kost
Tab	Accept AI auto-completion	Gratis
Cmd+K	Inline edit op selectie	1 request
Cmd+L	Chat-paneel openen	1 request

Tab — autocomplete

Cursor analyseert je hele bestand én recente edits. Hij suggereert hele blokken code, niet alleen één woord.

Voorbeeld: typ een commentaar `// render team list` op een lege regel. Cursor genereert de hele component eronder. Druk **Tab** om te accepteren, **Esc** om te negeren.

Wanneer: boilerplate, herhalende patronen, voorspelbare code (CRUD, list renders).

Cmd+K — inline edit

Voor gerichte één-bestand-wijzigingen.

Stappen:

1. Selecteer code (of zet cursor op een regel)
2. Druk **Cmd+K**
3. Input-balkje opent boven je code
4. Typ je instructie: `""converteer naar TypeScript""`
5. Cursor toont een **diff direct in de editor**
6. **Tab** = Accept, **Esc** = Reject

Wanneer: variabele hernoemen, één functie refactoren, types toevoegen, commentaar uitleggen.

Cmd+L — Chat

Voor vragen en grotere taken.

Stappen:

1. Druk **Cmd+L** — chat-paneel opent rechts
2. Optioneel: gebruik @-mentions voor context (zie volgende sectie)
3. Typ je vraag of taak
4. Cursor antwoordt + kan diffs voorstellen
5. Apply / Reject per file

Wanneer: uitleg vragen, grote multi-file taken, code-review.

7. Context via `@`-mentions

In Cmd+L (chat) type je @ voor een autocomplete-menu:

Tag	Wat het doet
@file:foo.tsx	inclusief één specifiek bestand
@folder:components	hele map
@code:functionName	losse functie of class
@docs:Next.js	officiële Next.js docs (eerst configureren)
@web	live web-search
@git	git history / blame

Voorbeeld:

```
@file:page.tsx Refactor deze component zodat hij @docs:React useReducer gebruikt in plaats van Redux
```

8. Request management

500 requests per maand klinkt veel, maar zonder discipline ben je er in een week doorheen.

Tips

- **Combineer in 1 Cmd+L:** `""maak een Hero met titel, ondertitel, twee CTA-knoppen, gradient achtergrond en responsive""` = één request voor een hele feature.
- **Tab is gratis** — gebruik het voor alles wat eenvoudig is.
- **Cmd+K voor gerichte edits** — niet voor "maak iets nieuws".
- **Cmd+L met @file** — bespaar Cursor het zoeken naar context.

Anti-patterns

- Tien losse Cmd+K's voor één feature
- Cmd+L met te weinig context ("maak iets cools")
- Vergeten dat Tab bestaat

9. `.cursorrules`

Een bestand in de root van je project dat Cursor bij elke prompt meeleest.

Voorbeeld

```
# .cursorrules

- Gebruik TypeScript strict mode
- Tailwind voor styling, geen CSS modules
- Server components by default; "use client" alleen waar nodig
- Imports met @/ alias voor src/
- Functional components, geen classes
- Geen any-types; expliciete return types op exports
```

Genereren

Vraag Cursor het zelf in Cmd+L:

```
Maak een .cursorrules voor dit Next.js project. Focus op TypeScript strict, Tailwind, en
```

Cursor stelt een goede basis voor. Pas aan naar smaak en commit het in je repo.

Effect

Je hoeft niet elke keer "vergeet niet TypeScript te gebruiken" in je chat te typen. Cursor weet het al.

Vooruitblik (Les 14): in nieuwere Cursor-versies is er ook een `.cursor/rules/`-map met meerdere `.mdc`-bestanden + YAML front-matter (`alwaysApply`, `globs`). Voor nu is `.cursorrules` in de root prima en simpeler.

10. Hands-on — wat je gaat bouwen

Drie kleine taken in je net gescaffolde project:

Taak	Wat	Sneltoets
1	Hero-component (titel, subtitel, 2 CTA-knoppen, gradient, responsive)	Openl, grad, resp
2	Styling-tweak: kleuren/spacing van de knoppen	Openl
3	Extra feature naar keuze: FeatureCards, ContactForm, Footer	Openl, Cmd+K

Daarna `git add . && git commit -m "first cursor build"`.

Exacte stappen + voorbeeldprompts: zie `Les03-Lesopdracht.pdf`.

11. Verder lezen

- **Cursor docs** — `cursor.com/docs` — features, sneltoetsen, rules-syntax
- **Cursor changelog** — `cursor.com/changelog` — wat is nieuw
- **Next.js docs** — `nextjs.org/docs`
- **Tailwind docs** — `tailwindcss.com/docs`

Samenvatting

Vandaag heb je geleerd:

- Cursor is een VS Code-fork met AI ingebouwd
- Student Plan = 500 requests/maand gratis (via `cursor.com/students`)
- Drie sneltoetsen: Tab (gratis), Cmd+K (inline edit), Cmd+L (chat)
- Context via @-mentions: file, folder, code, docs, web, git
- `.cursorrules` per project voor consistente AI-output
- Eerste hands-on met Hero / tweak / extra feature

Volgende les: TypeScript Fundamentals. Veel later (Les 14): Cursor 3 deep dive met Editor / Agents Window en deployment.