

React Installatiegids

### Toolchains

Je kan React in een bestaande HTML pagina toevoegen, net zoals je dat bij Bootstrap hebt gedaan. In de praktijk wordt deze optie niet vaak gebruikt - als je wil leren hoe je dat doet, kan je dat [hier](https://reactjs.org/docs/add-react-to-a-website.html) doen.

Wat wel vaak gedaan wordt, is het installeren van React via een ***Toolchain***. Dit zijn een verzameling programma’s die samen het gebruik van React makkelijker maken. Een Toolchain bestaat meestal uit drie onderdelen:

* Een **bundler**, zoals webpack of Parcel. Dit zorgt ervoor dat je je code makkelijk in meerdere files kan opsplitsen. De bundler zet ze wanneer dat nodig is allemaal bij elkaar.
* Een **compiler**, zoals Babel. Een compiler is een programma dat code “vertaalt”. Bij React betekent dit dat je nieuwe JavaScript trucjes kan gebruiken. Als je website dan bekeken wordt door een oudere browser, vertaalt de compiler automatisch je nieuwe JavaScript naar oudere JavaScript.
* Een **package manager**, zoals Yarn of npm. Dit zijn de programma’s die je gebruikt om de andere tools te installeren en ook gemakkelijk te updaten. Wij gaan tijdens de opleiding met npm werken.

## npm en Node.js

npm is een afkorting voor ***Node package manager***. Zoals de naam al weggeeft is het een programma dat andere software installeert. Je kan er redelijk op vertrouwen dat npm niet zomaar spyware installeert - het is een dochterbedrijf van Github/Microsoft. De officiële website van npm is [npmjs.com](https://www.npmjs.com/).

npm draait op ***Node.js***. Dat is een programma waarmee je JavaScript code buiten een browser kan laten draaien. **Als je Node.js installeert, installeert hij automatisch de laatste versie van npm**. Voordat je verder kan met React, zal je dit dus moeten doen. Dit kan via de officiële website: [Nodejs.org](https://nodejs.org/en/).

Lukt het niet om Node te installeren? Probeer de taalinstellingen van Windows naar Engels te wijzigen en kijk of het dan lukt. Als je nog steeds problemen hebt, schakel gelijk de LC in!

## De Terminal

npm werkt vanaf de Terminal - dat wordt ook wel eens een Command Line Interface genoemd. Je kan deze in VSCode openen, maar ook de terminal van je Operating System werkt hier prima voor.

Als je via npm packages installeert, maakt npm een package-lock.json aan waarin wordt bijgehouden welke packages je nou allemaal gebruikt. Als je net begint met React, is **create-react-app** de package die je wil gebruiken.

**npm i “packageNaam”**

Installeert de package “packageNaam”. Hiermee kan je dus zelf later nog packages aan je project toevoegen. De 2 packages die je bij React moet gebruiken zijn react en react-dom. Bij create-react-app zijn deze beiden inbegrepen.

**npx create-react-app projectNaam**

Creëert een react project template met de naam “projectNaam”. Het woordje npx ipv npm houdt in dat de package wordt gedownload, 1x uitgevoerd en daarna weer verwijderd. Werkt sneller dan “npm i create-react-app” + “create-react-app projectNaam” en verwijdert daarna dus ook nog de onnodige installatie files! Super handig.

**npm start**

Draait een test versie. Vergelijkbaar met live server, maar dan voor React. **Let op!** Hij probeert uit de huidige folder te draaien! Als je net “npx create-react-app new-app” hebt gebruikt, zit je nog niet automatisch in de folder “new-app”

**npm run build**

Verzamelt je huidige code en maakt daarmee een productie versie. Deze versie heeft bijvoorbeeld geen warnings in je console die de gebruiker krijgt te zien, zelfs al zit je website er stiekem nog vol mee.

Natuurlijk helpt het als je in de terminal een beetje kan navigeren door verschillende folders. Hier enkele commando’s die **WINDOWS** gebruikt om dit te doen:

**dir**

Toont alle files en folders in de huidige folder

**cd folderNaam**

Navigeert naar de benoemde folder binnenin de huidige folder

**cd ..**

Navigeert uit de huidige folder 1 folder omhoog/terug.

Als je Mac/Linux gebruikt moet je zelf even onderzoeken hoe dat moet!