

Cheat Sheet

Introduction & Setup

How this Course Works

This course will start at the very basics and gradually dive deeper into Angular 2. To support your learning path, the following materials are available (besides the video lectures):

- **Overarching project:** During the course a complete project will be built, besides the examples given in each lecture. This project will gradually incorporate the things learned. **Source code** of the project can be found at the **first and last lecture** of each module
- **Cheat Sheets** attached to the **last lecture** of each module
- **Code:** The **source code of each module's examples** can be found attached to the **last lecture** of that module

How to Ask Questions

I'm very happy to help all students to get the most out of this course. Responding to questions, fixing errors and discussing code is part of that service.

However, I'm only able to help timely if questions are posed in a certain format. Keep these points in mind when asking questions:

- If you got an error, provide the error message(s) from the JS console
- Copy & paste the code resulting in an error or to which you have a question
- Don't just copy & paste errors and code, also give a verbal description

Workspace Setup – Angular CLI

The Angular CLI makes the creation of projects very easy. Have a look at the corresponding course module for a deep dive on its capabilities and commands.

Here are the important commands you need to create a new project and work with it:

- Create new project with new folder: `ng new PROJECT_NAME`
- Create new project in existing folder: `ng init`
- Build project: `ng build`
- Serve project (will auto-reload upon changes to code): `ng serve`
- Create a new component: `ng generate component`

Workspace Setup – From Scratch (Custom)

Have a look at the corresponding module for a detailed guide on how to set up a project from scratch.

IDEs and Editors

Angular 2 code can be edited with any editor or IDE of your choice. Here are some ideas and links:

IDEs:

- Webstorm by JetBrains: <https://www.jetbrains.com/webstorm/>
It's a great IDE, very fast, lots of features and great TypeScript and Angular 2 support built-in
- Visual Studio Code: <https://code.visualstudio.com/>
Also ships with very good Angular 2 / TypeScript support (as TypeScript is maintained by Microsoft)

Editors:

- Atom: <https://atom.io/>
Great, free editor, created by the Github guys. Allows you to add plugins and change appearance with third-party packages.
- Sublime: <https://www.sublimetext.com/>
Great editor, very modular. Allows you to add plugins and change appearance with third-party packages.

TypeScript

TypeScript is the language of choice for Angular 2 development. It's a superset to JavaScript and therefore compiled into plain JavaScript. The main feature it adds are the Types, which make development easier (autocompletion etc.) and less error-prone (compiler errors).

Whilst you can create Angular 2 apps with plain JavaScript, there is way less documentation available and it is unnecessarily hard.

Check out the dedicated TypeScript module for more information about it.

You may also have a look at the official documentation for a deep-dive:

<https://www.typescriptlang.org/>

Angular 2 Styleguide

Angular 2 code should follow the official style guide. Of course it will work fine even if you're not following that guide, but it is a good practice to follow official style guides. This makes your code more readable to fellow Angular 2

developers and makes sure that you have an easy time reading the official docs.

Here's a link to the official styleguide: <https://angular.io/styleguide>