**1) Using Angular-CLI**

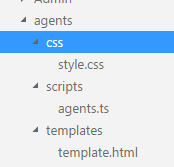
Every production ready code needs bundling. With the latest version, angular shipped its own CLI named Angular CLI. It helps us to create the basic app with basic architecture, setup typescript, enable live reload, build, code analysis, run unit tests, run end to end tests and deploy our code. We need to have nodejs installed to have angular-cli. The installation happens from npm.

**2) Using Visual Studio Code and installing ‘angular2-useful-dev-extensions’, ‘Angular v6 Snippets’**

Visual studio code has lots of extensions that enable auto complete and code completion. angular-useful-dev-extensions install a set of plug-ins that help us stay productive with angular. Generating angular snippets are automated using ‘Angular v6 Snippets’ by John Papa plug-in.

## 3) Some best practices from the Angular team

* Break everything into components. App module should be at the root of your application.  
  Modularize your use case to make widgets. A widget contains an angular component, its template and style.



* Each component and template should not exceed 400 lines.
* Make functions smaller. The maximum lines on a function should not exceed 10-15 lines (derived from Uncle Bob’s clean code)
* Use consistent name for all assets. Make classes upper camel case. Class name should be noun and method names should be verb.
* Append symbol name with conventional suffix. (for example, the login component can be named as LoginComponent with Component suffixed).

components

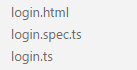
* File name should have conventional suffix. For example, a component can have its name suffixed with component.

layouts

* Use lower camel case for naming selectors and components.
* Separate words with hyphens in selectors.

selectors

* The best way to store bootstraping and platform information is main.ts.
* Configure your components to use logger. Use proper error handling.
* Unit testing is mandatory and the test spec file should have the same name as the component name.



* Do not use capital letters for constants. Use lower camel case instead.
* Do not use data type names while naming properties. ‘strusrename’ is wrong and ‘username’ is correct.
* Stop prefixing interfaces with ‘I’. It is a typescript best practice as some predefined interfaces use I during compilation.
* Use classes instead of interfaces. In ES5 and above only classes exist and even though we write interfaces, it will be converted into classes.
* Import line spacing. Leave an empty line between angular imports and custom imports



* Follow the LIFT principle (Locate your code quickly, Identify the code easily, Flattest structure, Try DRY-Don’t Repeat Yourself. )
* Use lazy loading on subcomponents and where ever possible.