**Angular**

Angular is a JavaScript front-end framework, which is used to develop web and mobile apps. As the name suggests, it is a declarative approach to developing an application. It is, therefore, different from other [**JavaScript frameworks**](https://www.interviewbit.com/blog/javascript-frameworks/) like React, which is an application of higher-order JavaScript. The development process of an Angular application is not different from other traditional web apps. The structure of the code is similar to the following: directives, services, models, pipes, and Rural routes. The only difference is that the source code of the application is written in angular format.

You may hear of too many versions of Angular as a beginner, and as a result, it is likely that you will get confused with so many different versions out there for the same framework. There are versions like AngularJS, Angular 2, Angular 4, Angular 5, Angular 6, Angular 7, Angular 8, and now Angular 9. There are actually two different frameworks - AngularJS and Angular.

In 2010, AngularJS was the initial release and was known as AngularJS. It was a JavaScript-based web development framework that was developed and maintained by Google. The type-setting language JavaScript is super-set to the language of Java. In September 2016, Angular 2 was created, which was a complete rewrite of the framework using TypeScript, a superset of JavaScript.

In this article, we will discuss some of the Angular features. You can follow this angular cheat sheet to build your application. We've tried to cover Angular CLI, Angular Lifecycle Hooks, Angular Routing, and a lot more in this post.

**Angular Tutorial: Basics to Advanced**

**1. Angular CLI**

The Angular CLI or the command line interface is a very powerful and sophisticated tool that enables you to perform a lot of tasks in an Angular project by utilizing simple commands. Everything is handled by the CLI.     In order to scaffold a brand-new Angular project, for example, the CLI generates the application, compiles the application, and ships it to you for testing.   The development server monitors the source code files for changes and when you change any of them, it automatically compiles the source code files and refreshes the app in the browser.

| **Command** | **Meaning** |
| --- | --- |
| npm install -g @angular/cli | To install the Angular CLI into our local machine using npm, run this command. |
| ng version | Displays the information about the currently installed CLI. |
| ng new <application name> | Using the ng new command, a new Angular application will be created. |
| ng new <application name> --prefix best | New project is created, and the project prefix is set to new. |
| ng new --help | All available Angular commands are returned by this command. |
| ng lint my-app | Linting warnings are checked against this command in our entire application. |
| ng lint my-app --fix | This command will correct any form of linting errors. |
| ng lint my-app --format stylish | Our entire codebase is formatted using this command. |
| ng lint my-app --help | The list of linting commands is returned by this command. |
| ng add <package name> | To use this command, you must first enable your package manager. Then, this command will use your package manager to download new dependencies and update your project with configuration changes. |
| ng generate component <name> | A new component of our application will be created as a result of this command. |
| ng g s <service name> | Creates a new class-based service based on Javascript classes. |
| ng g cl <destination> | This command creates a new class in the specified directory. |
| ng build | An application is created and stored in the dist directory using this command. |
| ng serve | The local development server is launched, and the app is served locally in the browser. Port and open are both specified. When you change any of the source files, the app is rebuilt and reloaded, and the page is changed automatically. |
| ng serve -o | This command opens up the application in a browser using any port 4200 or any available port |
| ng serve -ssl | This command enables the application to be accessed using SSL. |
| ng generate | To produce elements, services, components, classes, providers, pipes, and other types of modules. |
| ng g c MyComponent -d | This dry runs the code and helps in cleaning the command line clean. |
| ng g c MyComponent --export | This exports the component |
| ng g c MyComponent -f | This is used to overwrite the existing components. It forces rewriting. |
| ng g c --help | List of options for a given command is displayed using this. |

-+