# Getting Started with Angular

# **Objectives and Outcomes**

In this first Angular exercise, you will first install *angular-cli*, the command line tool for scaffolding Angular applications. You will then use the tool to scaffold out a basic Angular application. We will thereafter develop this application into a full-fledged Angular application in the process of doing the exercises in this course. At the end of this exercise you will be able to:

* Install *angular-cli*
* Scaffold out a basic Angular application

Installing *Angular-CLI*

From the Angular-CLI documentation we learn that the Angular CLI makes it easy to create an application that already works, right out of the box. It already follows the best practices suggested by the Angular community!

To install *angular-cli* globally, type the following at the prompt:

npm install -g @angular/cli

This will make the command line tool for creating Angular applications. To learn more about the various commands that this CLI provides, type at the prompt:

ng help

### Generating and Serving an Angular Project using Angular-CLI

* At a convenient location on your computer, create a folder named Angular and move into that folder.
* Then type the following at the prompt to create a new Angular application named conFusion:

ng new conFusion --style=css

* This should create a new folder named *conFusion* within your *Angular* folder and create the Angular application in that folder.
* Move to the conFusion folder and type the following at the prompt:

ng serve --open

* This will compile the project and then open a tab in your default browser at the address [http://localhost:4200](http://localhost:4200/).
* You can initialize your project to be a Git repository by typing the following commands at the prompt:

### Conclusions

In this exercise you installed the Angular CLI tool and created a basic Angular project and served up the compiled project to your browser.

### Angular Resources

* [Angular.io](https://angular.io/)
* [Angular CLI](https://cli.angular.io/)
* [Angular Material 2](https://material.angular.io/)
* [Angular Material Toolbar](https://material.angular.io/components/component/toolbar)
* [Angular Flex Layout](https://github.com/angular/flex-layout)
* [Angular Flex Layout Documentation](https://github.com/angular/flex-layout/wiki/API-Documentation)

### Typescript

* [Typescript](http://www.typescriptlang.org/index.html)
* [Typescript Tutorial](http://www.typescriptlang.org/docs/tutorial.html)
* [Typescript: Migrating from JavaScript](http://www.typescriptlang.org/docs/handbook/migrating-from-javascript.html)

### Definitions

* [Framework](https://en.wikipedia.org/wiki/Software_framework)
* [Hollywood Principle](https://en.wikipedia.org/wiki/Hollywood_principle)
* [Inversion of Control](https://en.wikipedia.org/wiki/Inversion_of_control)
* [Imperative vs Declarative Programming](https://netguru.co/blog/imperative-vs-declarative)
* [Imperative vs Declarative](http://latentflip.com/imperative-vs-declarative/)

### Blog Articles

* [5 Best JavaScript Frameworks in 2017](https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282#.tt1k09l1d)
* [Top JavaScript Frameworks & Topics to Learn in 2017](https://medium.com/javascript-scene/top-javascript-frameworks-topics-to-learn-in-2017-700a397b711#.pe809bf0u)
* [Declarative vs. Imperative Programming for the Web](http://codenugget.co/2015/03/05/declarative-vs-imperative-programming-web.html)
* [Intro to Material (2) Components, Custom Styles, Buttons & Indicators — Using Angular Material (2) Components in your Angular (2) Project](https://medium.com/@ladyleet/using-angular-material2-components-in-your-angular-2-project-intro-custom-styles-buttons-d2178e4b45c5#.y8cz8yk0r)

### Angular Resources

* [Angular Components](https://angular.io/docs/ts/latest/guide/architecture.html#!#components)
* [Angular Component Styles](https://angular.io/docs/ts/latest/guide/component-styles.html)
* [Angular Templates](https://angular.io/docs/ts/latest/guide/architecture.html#!#templates)
* [Angular Metadata](https://angular.io/docs/ts/latest/guide/architecture.html#!#metadata)
* [Angular Directives](https://angular.io/docs/ts/latest/guide/architecture.html#!#directives)
* [Structural Directives](https://angular.io/docs/ts/latest/guide/structural-directives.html)
* [Angular Pipes](https://angular.io/docs/ts/latest/guide/pipes.html)
* [Angular Material List](https://material.angular.io/components/component/list)
* [Angular Material Card](https://material.angular.io/components/component/card)
* [Angular Data Binding](https://angular.io/docs/ts/latest/guide/architecture.html#!#data-binding)
* [Angular Template Syntax](https://angular.io/docs/ts/latest/guide/template-syntax.html)
* [Angular Services](https://angular.io/docs/ts/latest/guide/architecture.html#!#services)
* [Angular Dependency Injection](https://angular.io/docs/ts/latest/guide/dependency-injection.html)

### Other Resources

* [Software design pattern](https://en.wikipedia.org/wiki/Software_design_pattern)
* [Model–view–controller](https://en.wikipedia.org/wiki/Model-view-controller)
* [Model-View-ViewModel](https://en.wikipedia.org/wiki/Model_View_ViewModel)
* [Model-View-Whatever](http://www.beyondjava.net/blog/model-view-whatever/)
* [Design Patterns: Elements of Reusable Object-Oriented Software](http://c2.com/cgi/wiki?DesignPatternsBook)
* [Web Application](https://en.wikipedia.org/wiki/Web_application)
* [Dependency Injection (Wikipedia)](https://en.wikipedia.org/wiki/Dependency_injection)