

Scaffold – nothing but the Saaram used while building house.

Angular CLI is a command-line interface tool to scaffold and build Angular applications

**Advantages:**

​​​​​​​**Cross-Browser Compliant**

Internet has evolved significantly from the time Angular 1.x was designed. Creating a web application that is cross-browser compliant was difficult with Angular 1.x framework. Developers had to come up with various workarounds to overcome the issues. Angular helps to create cross-browser compliant applications easily.

**Typescript Support**

Angular is written in Typescript and allows the user to build applications using Typescript. Typescript is a superset of JavaScript and more powerful language. The use of Typescript in application development improves productivity significantly.

**Web Components Support**

Component-based development is pretty much the future of web development. Angular is focused on component-based development. The use of components helps in creating loosely coupled units of application that can be developed, maintained, and tested easily.

**Better support for Mobile App Development**

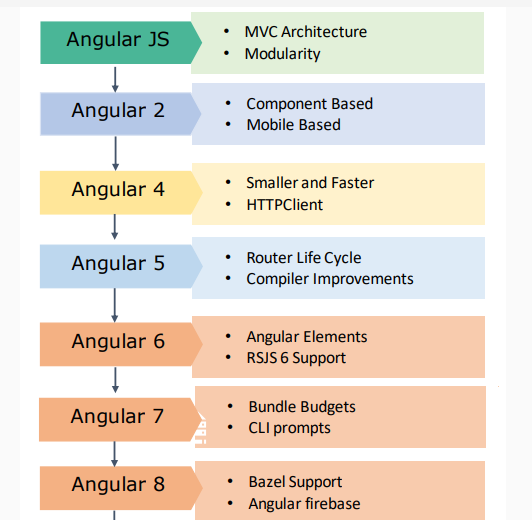
Desktop and mobile applications have separate concerns and addressing these concerns using a single framework becomes a challenge. Angular 1 had to address the concerns of a mobile application using additional plugins. However, the Angular framework, addresses the concerns of both mobile as well as desktop applications.

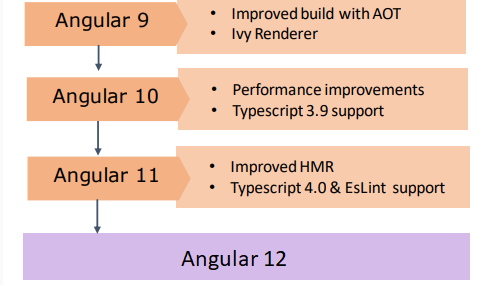
**Better performance**

The Angular framework is better in its performance in terms of browser rendering, animation, and accessibility across all the components. This is due to the modern approach of handling issues compared to earlier Angular version 1.x.

Angular helps to create SPAs that will dynamically load contents in a single HTML file, giving the user an illusion that the application is just a single page

**Evolution of Angular Framework:**





**Let us now understand what is Angular and what kind of applications can be built using Angular:**

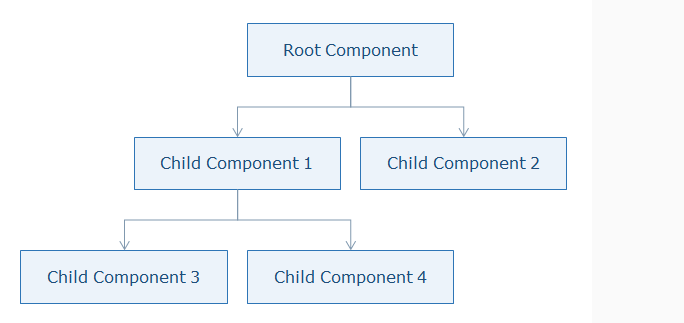
* Angular is an open-source **JavaScript**framework for building both mobile and desktop web applications.
* Angular is exclusively used to build **Single Page Applications (SPA).**
* Angular is completely rewritten and is not an upgrade to Angular 1.
* Developers prefer TypeScript to write Angular code. But other than TypeScript, you can also write code using JavaScript (ES5 or ECMAScript 5).

**Why most developers prefer TypeScript for Angular?**

* TypeScript is Microsoft’s extension for JavaScript which supports object-oriented features and has a strong typing system that enhances productivity.
* TypeScript supports many features like annotations, decorators, generics, etc. A very good number of IDE’s like Sublime Text, Visual Studio Code, Nodeclipse, etc., are available with TypeScript support.
* TypeScript code is compiled to JavaScript code using build tools like npm, bower, gulp, webpack, etc., to make the browser understand the code

Let us look at the features of Angular:

* **Easier to learn**: Angular is more modern and easier for developers to learn. It is a more streamlined framework where developers will be focusing on writing JavaScript classes.
* **Good IDE support**: Angular is written in TypeScript which is a superset of JavaScript and supports all ECMAScript 6 features. Many IDEs like Eclipse, Microsoft Visual Studio, Sublime Text, etc., have good support for TypeScript.
* **Familiar**: Angular has retained many of its core concepts from the earlier version (Angular 1), though it is a complete re-write. This means developers who are already proficient in Angular 1 will find it easy to migrate to Angular.
* **Cross-Platform**: Angular is a single platform that can be used to develop applications for multiple devices.
* **Performance:** Angular performance has been improved a lot in the latest version. This has been done by automatically adding or removing reflect metadata from the polyfills.ts file which makes the application smaller in production.
* **Lean and Fast**: Angular application's production bundle size is reduced by 100s of kilobytes due to which it loads faster during execution.
* **Bundle Budgets:** Angular will take advantage of the bundle budgets feature in CLI which will warn if the application size exceeds 2MB and will give errors if it exceeds 5MB. Developers can change this in angular.json.
* **Simplicity**: Angular 1 had 70+ directives like ng-if, ng-model, etc., whereas Angular has a very less number of directives as you use [ ] and ( ) for bindings in HTML elements.
* **Component-based:**
  + Angular follows component-based programming which is the future of web development. Each component created is isolated from every other part of our application. This kind of programming allows us to use components written using other frameworks.
  + Inside a component, you can write both business logic and view.
  + Every Angular application must have one top-level component referred to as 'Root Component' and several sub-components or child components.



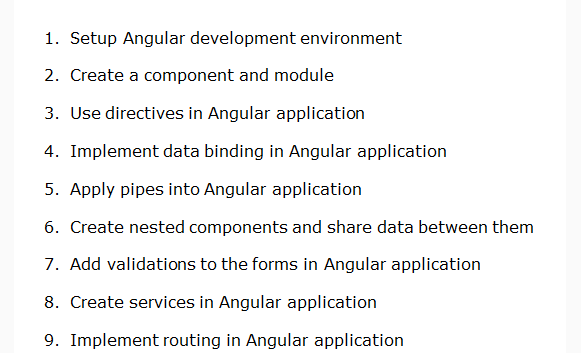
MCart application:

In this course, you will learn the Angular framework by exploring the implementations of the business requirements of an application called **mCart**.

mCart is an **online shopping application** that helps the users to purchase mobiles and tablet devices. This application allows users to log in for purchasing mobiles and tablet devices. They can select the products and add them to the cart. Once the selection is done, they can go to the cart page for payment. They can search for a product, sort the products list based on rating or price, and can filter the products list based on the manufacturer, operating system, and price.

| **User Stories** |
| --- |
| **Login** to the application to buy tablets/mobiles |
| **Search** for a specific product |
| **Filter**products based on manufacturer, price, and operating system |
| **View**the details of a specific product |
| **Sort** the products based on popularity and price |
| **Add products to the cart** which you want to buy |
| **Change the quantity** of the products selected for purchase |
| **Checkout**for closing the purchase |
| **Log out** from the application |

You will learn the Angular course by building the mCart application. Below is the roadmap to achieve it.



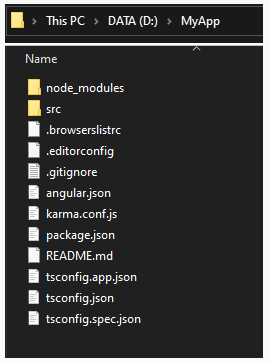
Node JS version check : node –v

Angular version check : ng -version

|  |  |
| --- | --- |
| **Command** | **Purpose** |
| npm install -g @angular/cli | Installs Angular CLI globally |
| ng new <project name> | Creates a new Angular application |
| ng serve --open | Builds and runs the application on lite-server and launches a browser |
| ng generate <name> | Creates a class, component, directive, interface, module, pipe, and service |
| ng build | Builds the application |
| ng update @angular/cli @angular/core | Updates Angular to a newer version |

To create new project : ng new <Project Name>

This will create the following folder structure with the dependencies installed inside the node\_modules folder.



| **File / Folder** | **Purpose** |
| --- | --- |
| **node\_modules/** | Node.js creates this folder and puts all npm modules installed as listed in package.json |
| **src/** | All application-related files will be stored inside it |
| **angular.json** | Configuration file for Angular CLI where we set several defaults and also configure what files to be included during project build |
| **package.json** | This is a node configuration file that contains all dependencies required for Angular |
| **tsconfig.json** | This is the Typescript configuration file where we can configure compiler options |
| **tslint.json** | This file contains linting rules preferred by the Angular style guide |

**Why Components in Angular?**

* A component is the basic building block of an Angular application
* It emphasize the separation of concerns and each part of the Angular application can be written independently of one another
* It is reusable
* Modules in Angular are used to **organize the application**. It sets the execution context of an Angular application.

Modules:

* A module in Angular is a class with the **@NgModule** decorator added to it. @NgModule metadata will contain the declarations of components, pipes, directives, services that are to be used across the application.
* Every Angular application should have one root module which is loaded first to launch the application.
* Submodules should be configured in the root module.