**What is Angular**

Angular is an open-source framework mainly used for creating web apps, single-page web apps, and hybrid apps. Angular is built on TypeScript, developed and maintained by Google.

Angular is a full-fledged toolkit that has everything developers need to build large-scale apps. No wonder it gains popularity: leading businesses, like **Xbox**, **Forbes**, **BMW**, and many more, choose Angular over React for front-end development.

Angular allows websites to render quickly and efficiently. It’s an all-in-one framework, with built-in tools for routing, state-management solutions, and form validation. Angular is built on TypeScript, a superset of JavaScript.

Angular is the part of a MEAN stack and is compatible with most used code editors and is considered for creating dynamic websites and web applications.

Angular was released in 2013, whereas Angular 2, also known as Angular, was released in 2016. Here, we are talking about Angular and not AngularJS.

Google supports Angular.

## ****Benefits of Angular Framework****

### **Cleaner Code**

Angular uses TypeScript language, which is the superset of JavaScript. It compiles to JavaScript, but also ease the process of finding and eliminating the common issues while typing the code.

### **Material Design-like Interface**

Angular offers pre-built material design components such as navigation elements, form controls, pop-ups, layouts, and data tables.

This helps the mobile developers to overcome Google Material Design’s impact on Mobile app design and design the digital product that users need in the long run.

### **Better Error Handling**

Angular offers feature like the upgraded error handling process for @Output in the scenarios where the property is not initialized.

### **Seamless Updates using Angular CLI**

Angular CLI is easy to install and use, newcomer-friendly, offers innovative testing tools, and more straightforward commands and is supported by different engineers and developers, which makes it possible to update even those components having a third-party dependency.

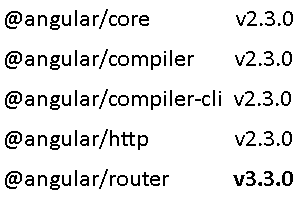
### **ANGULAR 1.X (ANGULARJS)**

* Angular1.x usually referred to as “Angular.js” or “AngularJS”
* It is a JavaScript-based open-source front-end web application framework.
* Maintained by Google and by a community of individuals and corporations to solve challenges that occurred during single-page application development.
* It aims to simplify both the development and the testing phase of applications by providing a framework for client-side model–view–controller (MVC) and model–view–ViewModel (MVVM) architectures, along with components commonly used in rich Internet applications.
* It was initially released in Oct 2010.
* AngularJS is written in JavaScript.

### **ANGULAR 2**

* Angular 2 is a complete rewrite from the same team that built AngularJS.
* It is entirely written in TypeScript.
* Angular 1.x doesn’t support mobile, where Angular 2 is mobile-oriented.
* It was released in September 2016.
* Commonly referred to as “Angular 2+”
* It provides more choices for languages. You can use any of the languages from ES5, ES6, TypeScript or Dart to write Angular 2 code.

### **ANGULAR 3**

* Angular 3 was skipped because the Angular team wanted to sync the angular version with the angular router version.
* The reason behind this is that mismatch in the version between **@angular/core**, **@angular/compiler** and **@angular/router** libraries.
* In Angular 2, version for core and router is like this
* Now the problem creates with the **@angular/router**, which is already in a 3.X version. and that’s because of some active and huge developments in the router section, like route-preload.
* Now releasing Angular as version 3, with its route on version 4 will create confusion.
* To avoid this confusion they decided to skip version 3 and release with version 4.0.0 so that every major dependency is on the right track.

### **ANGULAR 4**

* Angular 4 was released in March 2017.
* It is backward compatible with Angular 2 for most applications.
* There is no major change in Angular 4 from Angular 2 and it is not a complete rewrite of Angular 2.
* New changes reduce the size of the generated code for your components, faster compilation, and better bug fixes alert.
* Introduced HttpClient, a smaller, easier to use, and more powerful library for making HTTP Requests.
* **Typescript:** We can use typescript 2.1 or earlier only up to typescript 1.8 was supported.
* Easily add animation by importing {BrowserAnimationsModule} from **@angular/platform-browser/animations** into **NgModule**.
* No need to write a pattern for email validation in Angular 4.

### **ANGULAR 5**

* Angular 5 was announced to the world in November 2017.
* This release focused on making Angular smaller and faster to use.
* **HttpClient:**Before version 4.3, the **@angular/http** module was used for making HTTP requests in Angular applications. The Angular team has now deprecated **@angular/http** in Angular 5. It is replaced with **@angular/common/http** library.  
  **e.g.** import { HttpClientModule } from ‘@angular/common/http’;
* In Angular 5, you can now give multiple names to your components and directives while exporting. Exporting a component with multiple names can help your users migrate without breaking changes.
* It shipped with a new number, date, and currency pipes that increase standardization across browsers and eliminate the need for i18n polyfills.

### **ANGULAR 6**

* Angular 6 is released with Angular CLI 6 and Material 6.
* The 6th version is released in May 2018 proves to be a major breakthrough and is the latest release from the Angular team after Angular 5.
* **Updated Angular CLI, Command Line interface:** New commands added like **ng-update** to migrate from the previous version to current version and **ng-add**to quickly add application features to make the application progressive web apps.
* Updated Component Development Kit and Angular Material
* **Multiple Validators:** Allows multiple validators to be applicable on a form builder.
* **RxJS:**Reactive Extensions for JavaScript (RxJS) offers a number of functions that can be used to create new observables. These functions can simplify the process of creating observables from things such as events, timers, promises, and so on.

### **ANGULAR 7**

* Angular 7 was released in October 2018.
* **CLI prompts:**Angular CLI has updated to v7.0.2 with some features like now it will prompt users while typing commands like **ng-add, ng-new** and **@angular/material** to help you discover inbuilt SCSS support and routing.
* **Angular Material & Component Development Kit (CDK):**Angular 7 introduced minor visual updates and improvements in Material Design.
* Also, refresh, virtual scrolling, large lists of data, dynamic loading and unloading of parts of the DOM were the part of improvements in CDK and Angular Material.
* **Drag & Drop:**The **@angular/cdk/drag-drop** module provides a better way to easily create drag and drop interfaces.
* **Virtual Scrolling:**Virtual scrolling enables loading and unloading elements from the DOM based on the visible parts.

### **ANGULAR 8**

* Angular 8 was released in May 2019.
* Angular v8 arrives with the much-anticipated Ivy compiler as an opt-in feature.
* The most important features of Angular 8 are differential loading of modern JavaScript, dynamic imports for lazy routes, support for web workers, TypeScript 3.4 support, and Angular Ivy as an opt-in preview.

### **ANGULAR 9**

* Angular 9 was released in October/November 2019.
* This is a major update to Angular, which adds the new and improved Ivy compiler and runtime, faster testing, better debugging, and many more improvements.
* The most notable change is Ivy. With the release of Angular 9, Ivy is now the default compiler and runtime. Ivy improves bundle size, allows for better debugging, adds improved type checking, faster testing, enables the AOT compiler on by default, and improves CSS class and style binding.
* The Angular 9 Language Service for Visual Studio Code and WebStorm will benefit from some improvements. For example, the URL definition will become more consistent. Style URLs will be checked in the same way as template URLs.
* Angular 8 has entirely based on component and consist of some tree structures with parent and child component. Angular 8 classes are created in such a way that the web page can fit in any screen size so that they are fully compatible with mobiles, tablets, laptops, and large systems.

### Angular Versions

|  |  |
| --- | --- |
| Version | Released |
| Angular JS | **October 2010** |
| Angular2.0 | **September 2016** |
| Angular4.0 | **March 2017** |
| Angular5.0 | **November 2017** |
| Angular6.0 | **May 2018** |
| Angular 8.0 | **October 2018** |

* **The release dates of major two upcoming versions of Angular are following:**

|  |  |
| --- | --- |
| Version | Released |
| Angular 8.0 | **March/April 2019** |
| Angular9.0 | **September/October 2019** |

* Google plans to release the significant Angular version every six months.

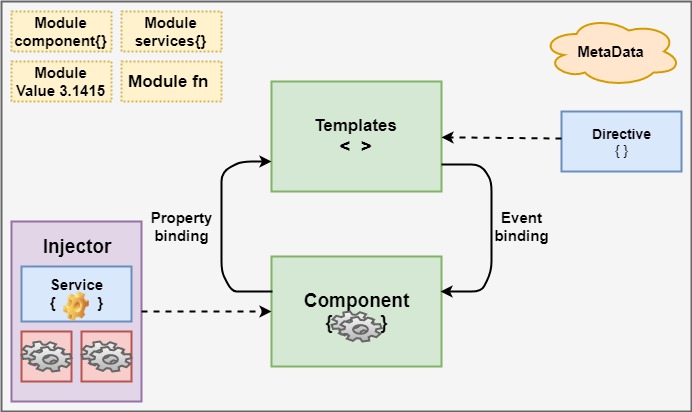
**Architecture of Angular 8**

Angular 8 is a platform and framework which is used to create clients applications in HTML and Typescript. *Angular 8 is written in Typescript*. [Typescript](https://www.tutorialandexample.com/typescript-tutorial/) is a super-set of JavaScript.

[Angular 8](https://www.tutorialandexample.com/angular-8-tutorial/) implements core and optional functionality as a set of Typescript libraries which we can import in our application.

**NgModules** are the *basic building blocks of Angular 8 application*. They provide a compilation context for components.

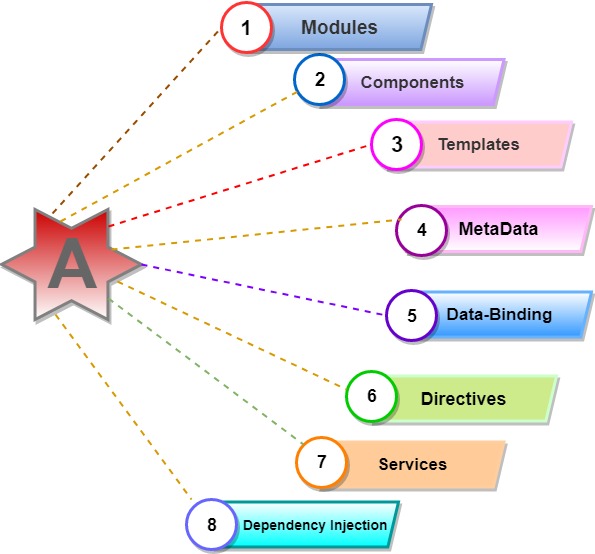
Various units are combined to *build an angular application*, which is as follows.



Angular 8 always has at least a root module which enables bootstrapping.

Components define views, which are set of screen element that is modified according to your program logic and data in Angular 8.

These are the essential modules which are as follows:



**1. Modules:**

Angular 8 NgModules are different from other JavaScript modules. Every Angular 8 app has provided the [bootstrap](https://www.tutorialandexample.com/bootstrap-tutorial/) mechanism that launches the application.

Generally, every angular 8 modules contain many functional modules.

Some interactive **features of Angular 8 modules**:

* NgModules import the functionality from another NgModules just like other JavaScript modules.
* [NgModules](https://angular.io/guide/ngmodules) allow their functionality to be imported and used by other modules, e.g., if we want to use route service in our app, we can import the Routing Ng module.

**2. Components:**

Every angular project has at least 1 component, the root component, and the root components connect the component with a page Document Object Module (DOM). Each component defines a class which contains data, application, logic, and it is binding with the HTML template.

@Component({

selector: ‘app-root’,

TemplateUrl: .’ /app.component.html’,

StyleUrls: [‘. /app.component.css’]

})

**3. Templates:**

The angular template integrates the HTML with Angular mark-up that can modify HTML elements before they are displayed. It provides program logic, and binding mark-up connects to your application data and the DOM.

<div style="text-align: center">

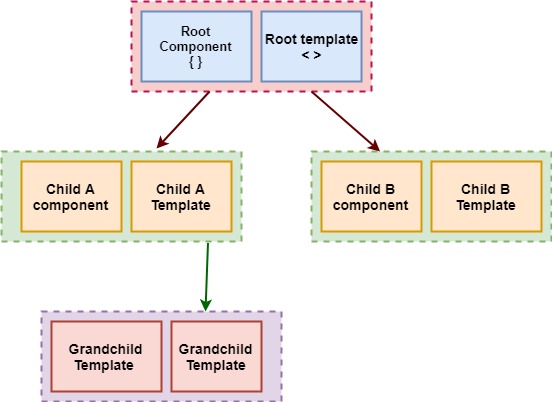
<h1>

{{2| power: 5}}

</h1>

</div>

In the above HTML file, we have been using a template. We have used the pipe inside the template to convert the values to its desired output.



**4. Metadata**

Decorators are the metadata in Angular. It is used to enhance the class so it can configure the expected behavior of a class. Developers are the core concept of when developing with Angular. User can use metadata in a class to tell Angular app that app component is a component. Metadata can attach to the Typescript through the decorator.

@Component ({

selector: ‘app-root’,

templateUrl: ‘./app.component.html’,

styleUrls: [‘./app.component.css’]

}]

**5. Data-binding:**

Angular allows communication between a component and a DOM. Making it very easy to define interactive application without pulling and pushing the data.

**There are two types of data binding.**

**Event Binding:**Our app responds to user input in the target environment by updating our application data.

**Property Binding:**Itinterpolates values that are computed from application data into the HTML.

Interpolation: {{value}}: Interpolation sums the value of the property to the component.

Name: {{student.name}}

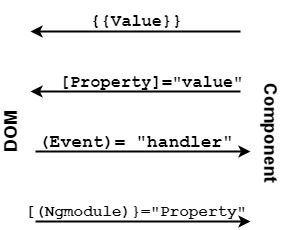
<p>College: {{student. college}} </p>

### Property binding: [property]=”value”

 A value has been passed from a component to a special property, with property binding which can be a simple html attribute.

<input type="”text”" [value]="”student.name”/">

<input type="”text”" [value]="”student." college”="">



**6. Directives:**

Directives used for expanding the functionality of HTML elements. Three types of directives in Angular are Structural Directive, Attribute directive and component directives. We can build in Angular directives like: ngClass which is a better example of excising angular attribute directive.

<p [ngclass="“{‘coffee’" =true,="" ‘red’="false}”">

Angular8  Directives Example

</p>

<style>

.coffee{color: coffee}

.red{color: red}

</style>

**7. Services:**

Services used to reuse the code. This service creates for that code which code is standard for more than one component. The decorator provides the meta-data that allows our services to be injected into the client component as a dependency. Angular distinguishes an element from service to increase modularity and re-usability. By separating component functionality from other kinds of processing, we can make our component more lean and efficient.

**8. Dependency Injection:**

It is a design pattern for efficiency and modularity. **DI** is wired into Angular into an Angular framework and used everywhere to provide new component with new services. Dependency injection does not fetch data from a server, validate the user input, or log directly to console; instead, they delegate such tasks to the service.

**Pipes:**

A class with the @Pipe decorator defines a function which transforms an input value to output value for display in an amount.

Angular defines various pipes like data pipe and currency pipe; for a complete list and we can also define new pipes. To specify a value transformation in a HTML template, use the **pipe operator (|).**

{{interpolated\_value | pipe\_name}}

A pipe can also take dispute that control how it performs it performs its transformation so that we can pass the desired output in the pipe.

<!--Default format: output ‘Jun 15, 2015’-->

<p>Today is {{today| Date}} </p>

<!—Full Date format: output ‘Monday, June15, 2015’-->

<p> the date is {{today | date: 'fullDate'}} </p>

<!—short Time format: output ‘9:43 AM’-->

<p> the time is {{today | date: ‘short Time ‘ }} </p>

### Angular Performance & Up gradation from Angular 7 to Angular 8

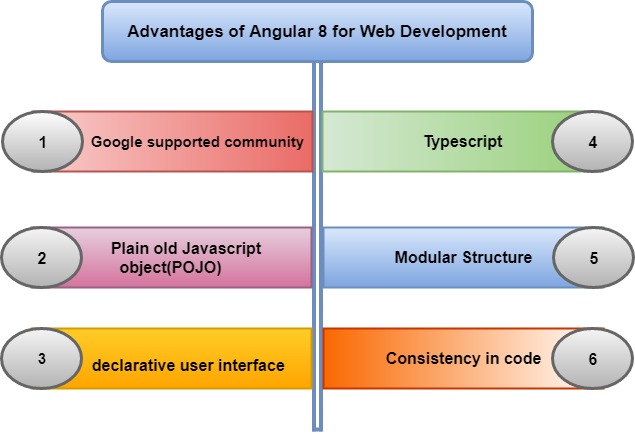
Angular 8 new features are great, but the main reason for many of us to upgrade to new versions of Angular 8 is to get a performance boost. If we worked with previous angular versions, then upgrading an app from Angular 7 over to Angular 8 is simple.

Command to update the version of Angular 7 to Angular 8.

Ng update @angular/cli @angular/core<name>

### Benefits of Angular 8 in Web Development

Many benefits of Angular 8 that invokes user to use it as a framework for web development.



1. **Google Supported Community:**Angular comes with Google’s long-term support (LTS). The Google team is very confident about Angular’s stability; also, Google apps use an angular framework.
2. **Plain Old JavaScript Object (POJO**): It does not require any getter and setter function. It used every object as an everyday old JavaScript object. It provides JavaScript functionality to enable manipulation of an object such as adding properties or removing properties from the purpose.
3. **Declarative User Interface:**Angular uses HTML to define the view part of an Application, which is a complex language. Html is a declarative language too. We don’t worry about the flow of the program when it loads define what we want as per application requirement, and angular will take care of rest things.
4. **Typescript:** It is written in Typescript, which is a superset of JavaScript. It promotes high security. If we have created proper map files during build time, then you can easily debug typescript code in an editor or a browser.
5. **Modular Structure:**Angular organizes code into modules whether it is components, directives, pipes, or services. It makes the organizing of functionality easy and straight forward by separating the code. It also offers lazy loading.
6. **Consistency in code:**It maximizes the readability of the code. For any new developer, it is an easy task to go through the project because of its code consistency features.