# **AngularJs vs. Angular 2 vs. Angular 4 vs. Angular 5 vs. Angular 6 vs. Angular 7 vs Angular 8**

|  |  |
| --- | --- |
| **Angular JS** | * It is controller based where a $scope variable is used to communicate between the HTML files. * Does not support mobile devices |
| **Angular 2** | * Released in 2016 * Complete rewrite of Angular 1 * Written entirely in typescript * Component-based instead of Controller * ES6 and typescript supported * More testable as component-based * Support for Mobile/Low-end devices * Up to typescript 1.8 is supported |
| **Angular 4** | * Released in 2017 * Changes in core library * Angular 4 is simply the next version of angular 2, the underlying concept is the same & is an inheritance from Angular 2 * Lot of performance improvement is made to reduce size of AOT compiler generated code * Typescript 2.1 & 2.2 compatible — all feature of ts 2.1 & 2.2 are supported in Angular 4 application * Animation features are separated from [@angular/core](http://twitter.com/angular/core) to [@angular/animation](http://twitter.com/angular/animation) * Else block in \*ngIf introduced |
| **Angular 5** | * Released 1st November 2017 * Build optimizer: It helps to removed unnecessary code from your application * Angular Universal State Transfer API and DOM Support — By using this feature, we can now share the state of the application between the server side and client side very easily. * Compiler Improvements: This is one of the very nice features of Angular 5, which improved the support of incremental compilation of an application. * Preserve White space: To remove unnecessary new lines, tabs and white spaces we can add below code(decrease bundle size) * Increased the standardization across all browsers * exportAs: In Angular 5, multiple names support for both directives and components * HttpClientModule is introduced * Few new Router Life-cycle Events being added in Angular 5 * Angular 5 supports TypeScript 2.3 version. * Improved in faster Compiler support |
| **Angular 6** | * Released on April 2018 * This release is focused less on the underlying framework, and more on tool-chain and on making it easier to move quickly with angular in the future * No major breaking changes * Dependency on RxJS 6 * Remove support for *<template>* tag and “*<ng-template>*” should be used. * Registering provider * The way ngModelChange event works * CLI Changes * CLI + Material starter templates * It uses angular.json instead of .angular-cli.json * Support for multiple projects. * Angular Elements |
| **Angular 7** | * Released on October 2018 * **CLI Prompts** * Added a new interface — UrlSegment[] to CanLoad interface * Added a new interface — DoBootstrap interface * Angular 7 added a new compiler — Compatibility Compiler (ngcc) * Introduce a new Pipe called — KeyValuePipe * Angular 7 now supporting to TypeScript 2.9. * Added a new elements features — enable Shadow DOM v1 and slots * Added a new router features — warn if navigation triggered outside Angular zone * Added a new mapping for ngfactory and ngsummary files to their module names in AOT summary resolver. * Added a new “original” placeholder value on extracted XMB * Added a new ability to recover from malformed URLs * Added a new compiler support dot (.) in import statements and also avoid a crash in ngc-wrapped * Update compiler to flatten nested template fns |
| **Angular 8** | * Releasing March/April 2019 * Being smaller, faster and easier to use and it will be making Angular developers life easier. * Added Support for TypeScript 3.2 * Added a Navigation Type Available during Navigation in the Router * Added pathParamsOrQueryParamsChange mode for runGuardsAndResolvers in the Router * Allow passing state to routerLink Directives in the Router * Allow passing state to NavigationExtras in the Router * Restore the whole object when navigating back to a page managed by Angular Router * Added support for SASS * Resolve generated Sass/Less files to .css inputs * Added Predicate function mode for runGuardsAndResolvers:- * Added functionality to mark a control and its descendant controls as touched: — add markAllAsTouched () to AbstractControl * Added an ng-new command that builds the project with Bazel * Use image based cache for windows BuildKite * Export NumberValueAccessor & RangeValueAccessor directives * Use shared DomElementSchemaRegistry instance for improve performance of platform-server([@angular/platform-server](http://twitter.com/angular/platform-server)) * Now the Performance Improvements on the core, more consistent about “typeof checks” * In the Compiler-CLI, expose ngtsc as a TscPlugin * Restore whole object when navigating back to a page managed by Angular Router |