Angular

1. Angular Module
   1. Container for
      1. Components
      2. Services
      3. Pipes
      4. Directive
2. Decorator
   1. An attribute applied on Class/Method/Member to define behavior of it
   2. @NgModule
      1. Class is angular module
   3. @Component
      1. Class is component
   4. @Injectable
      1. Class will be used for DI
      2. Applied on class to be used as Angular Service
   5. @Input
      1. Applied on Member (public variable / property) to accept data from parent component
   6. @Output
      1. Applied on EventEmitter to emit event from Child component to parent component
   7. @ViewChild
      1. Applied on Component Instance to make component as DOM Child of Parent component
   8. @HostListener
      1. Applied on method with Event so that when an event occur the method will be auto-invoked
3. @angular/core
   1. Package that contains Object model for Angular Programming
      1. @NgModule, @Component, @Input, @Output, etc
4. @angular/compiler
   1. Ahead-of-Time (AoT) compilation service
      1. Optimized Build of the code
5. @angular/Common
   1. Common Module-loader for external modules
   2. @angular/common/http
      1. HttpClientModule Angular 4.0+
6. @angular/forms
   1. Form Object
      1. FormsModule
         1. Used for ngModel
      2. ReactiveFormsModule
         1. Data-Driven / Model-driven / Reactive Forms
            1. Model validation forms
7. @angular/router
   1. SPA
8. @angular/platform-browser
   1. Provides BrowserModule that loads Angular OM in browser
9. @angular/platform-browser-dynamic
   1. Dynamically executes Angular OM for
      1. Databinding
      2. Event-Binding
      3. Directives

**Angular Support Packages OR external dependent Libs**

1. Rxjs
   1. Reactive Extension for JavaScript, supported by DevLabs that is inspired from Rx.NET
   2. An Observable class, maintain state of An Async Operations
2. Core-js.js
   1. ES 6 compatibility to all browsers those who does not support ES 6 and only uses ES 3
      1. Class
         1. Access Specifier
            1. Public (Default)
            2. Private
            3. Protected
         2. Access Modifiers
            1. Static
            2. Async 🡪 await
      2. Datatypes (only in TypeScript)
      3. Arrow Operation/ Lambda =>
      4. Promises
      5. Template String
         1. Build an Immutable Html parsed string
            1. `${<Variable>}`
      6. Modules
         1. Export
         2. Import
      7. Transpiler
         1. Transform ES 6 to ES 6
         2. ES 6 Engines
            1. TypeScript

Microsoft

Npm install -g typescript

The **tsc** transpiler

* + - * 1. ES2015

High-Level JavaScript

Bable

Used by React

Used Angular

Tracure

* + - * 1. Dart

1. Zone.js
   1. Library used by platform-browser-dynamic for
      1. Browser level Error Logs and StackTrace

Package.json

* Application configuration file that contains
  + Runtime and Development time dependencies
  + ng build –aot
  + ng build –prod
  + ng serve –prod

Programming with Angular

1. Create a Component
   1. Class decorated with @Component() decorator
      1. Component class Name should be
         1. XXXXComponent
            1. E.g. SimpleComponent
      2. Component File Name
         1. app.XXXXX.component.ts
            1. E.g. app.simple.component.ts
   2. @Component properties
      1. The **selector** property
         1. This is a custom HTML tag to load component on HTML Page
         2. Naming Comventions
            1. app-<COMPONENT-Name>-component
      2. The **template** property
         1. Contains inline HTML Template means write HTML inside component’s file
      3. The **templateUrl** property
         1. Contains an Absolute path for external Html file
      4. The **style** and **styleUrls** properties
         1. Contains an inline and external CSS respectively
      5. The **provider** property
         1. Contains DI for Angular Service specific to component
2. Create a Angular Module
   1. Class decorated with @NgModule
   2. Properties for NgModule
      1. The **imports** an array type property, used to import all standard Angular Module those are needed for the current application
      2. The **exports** an array type property, used to export components, services, directives, ect from current application to other angular application
      3. The **declarations** an array type property, used to declare components and directives and piped of current application in module for **lazy loading.**
      4. The **providers** an array type property, used to declare Services in DI for current application.
      5. The **bootstrap** an array type property, used to define component(s) to be executed when the current application is loaded in browser. Note these components must be already declared in **declarations.**
3. General Folder Structure with Angular CLI
   1. Src🡪app🡪components🡪Folder-for-each-component
      * 1. 🡪services🡪app.XXXX.service.ts
        2. 🡪directives🡪app.XXXX.directive.ts
        3. 🡪pipes🡪 app.XXXX.pipe.ts
        4. 🡪models🡪 app.XXXX.model.ts

Databinding

* + - 1. Angular Interpolation
         1. Bind public member of Component class to UI using {{}}.
         2. Read-only binding
         3. Value can only be changed from component
      2. In Two-Way Binding, we use ngModel, and we need to import FormsModule from @angular/forms in NgModule.

CSS

TextBox 🡪 form-control

Top level Div 🡪 container

Inner DIV 🡪 form-group