Angular 2:

What and Why Angular JS:

* Framework for creating Single Page application.
* Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app.
* Example = <https://angular-university.io> and go to course. When you select anyone course page does not fully reloaded.

Install:

* Use Angular CLI (Command Line Interface)
* To use CLI, need node JS (Server-side language) but we mostly needed node JS for NPM (node package management). NPM is famous for managing various dependencies of your project.
* Open window CLI and enter npm install -g angular-cli and download CLI and install globally

Create Project:

* Create project with new folder: ng new project name
* Create project in existing folder: ng init
* Build project: ng build
* Create new component: ng generate component
* Build and start server = ng serve
* To check server is up and running - localhost:4200
* Whenever make you changes, server automatically detect it, rebuild it and changes will visible to browser

Type Script:

* Super set of Javas Script which primarily provide static typing, classes and interface
* Type script always compile to Javas script because not understand Type script
* Why: -
* Strong typing, Next Gen Java Script feature, Interface and Generics,
* You can write any java script code in type script and it will work

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: 'app.component.html',

styleUrls: ['app.component.css']

})

export class AppComponent {

title = 'I changed it!';

}

Import: Import code from another file. You can import @angular package/module or code from your own file.

Never add file extension in the import statement

Decorators (@Components): Decorator are basically function which get attached to other code in this

case to a class.

Decorator execute some piece of code in background which “Does Something”.

Here @Component take java script object as an argument and use this argument

To add some metadata (in background) to this class.

Angular 2 uses Decorators a lot (not only for Components) and you may

simply keep in mind that Decorators "do something in the background

and transform whatever they are attached to, to something else

Angular 2 knows (like a Component)".

We always import decorator like here @Component from @angular/core modul

Class: Blueprint of Java Script Object.

You rarely crate instance of class (Crate JS object based on them) yourself in Angular App2. Angular 2 will

do that for you.

Types: Types to properties, variable etc.

Ex: someproperty: string = ‘Hello’; (This makes sure that only string values may be

assigned to someProperty.)

Important Points:

* Create new component = ng generate component component-name
* First request goes to main.ts (looking bootstrapModule(AppModule) function)
* Next goes to app.module.ts. Here we have another decorator @NgModule and this decorator having set of statements:
* Declarations [AppComponent,…] = Declares which directive use in your application. Component is like directive. Directive means a set of instructions to tell app what to do.
* Imports[] = Imports tell angular which our module do I use. We have some predefined module.
* Providers[] = declare service which you want to use application vise.
* Bootstrap[AppComponent] : Telling to angular app which is root component for application.
* <app-root> = this is selector and it should be unique, and selector is allowing to use component through out application. This selector work like CSS selector. Selector can select ids , css classes like same css selector works.
* Index.html is a file which at last loaded in server.
* CLI dynamically add some script whenever we run our application.
* Main.nundle.js = this main bundle file contains all our code, angular code and third party package and this js start our application.
* Main.ts file executed first.

View Encapsulation:

<ng-content>