Angular 2 application anatomy

1. Appln=component +component
2. Compo= template+class+metadata

Language that can be used to develop AJS

1. JS lang specification standard officially called ECMAScript (ES)
2. Typescript= superset of JS
3. Dart=non JS based lang for developing Angular JS

Type Script Editors

* VS Code(https://code.visualstudio.com/docs/?dv=win), Atom, Eclipse

Enviornment:

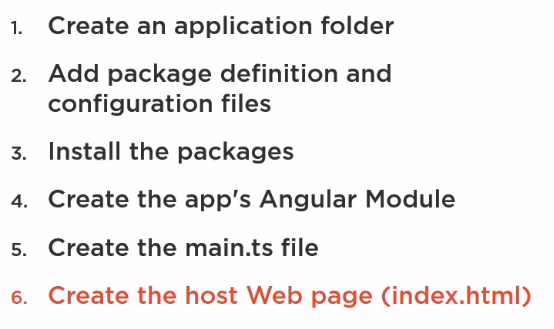
Required:

npm – is a CLI utility that interacts with the repository of open source projects

* A package manager for java script
* To install all lib of angular
* Download from npmjs.com

Setting up the Angular 2 application

To manually setup the A2 application



Alternate steps:

* + - Tooling like Angular CLI:

CLI for generating the setup files and boilerplate code for A2 application (incl Scaffolding unit and end to end test)

* + - Starter files for building our application from git hub

<https://github.com/DeborahK/Angular2-GettingStarted>

Not req: to separately/manually install angular

7) Component:

    A component includes:

    a) A **View** defined in a template

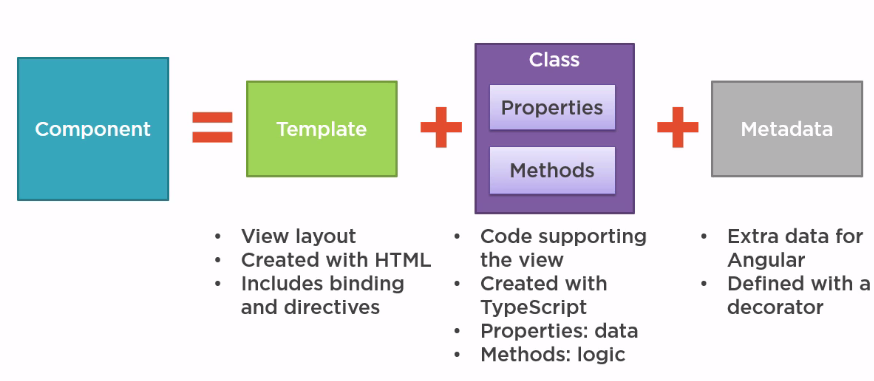
    b) Its associated code defined within a **Class** (Properties, Methods)

         A Class becomes a function when we give a metadata.

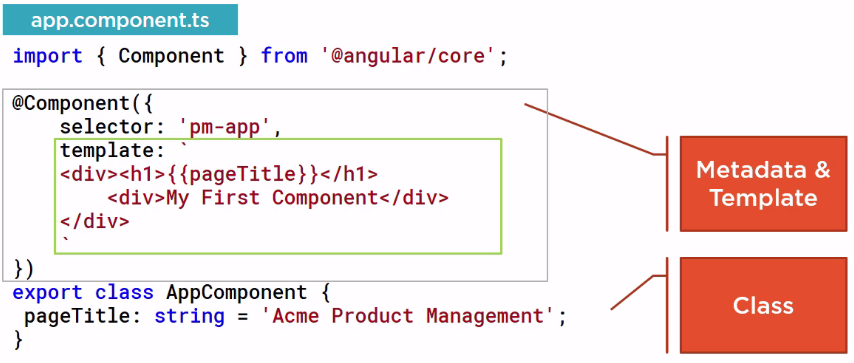
    c) **Metadata** defined with a  **Decorator** (an angular component function)

             Decorator is a function that adds Metadata to a class, its members.

             Always prefixed with @



Example of a component in a TypeScript:



Explanation of the above code from bottom:

**Class definition:**

Angular convention of naming a class:

\* classname followed by suffix '*Component*'.

\* This classname is the *Component name.*

\* The root component of an application is '*AppComponent'*

*\** Since the above class 'exports', it is called as an **ES Module**.

\* The class 'AppComponent' has only 1 property '**pageTitle**' of type string and no methods.

*class <classnameComponent>  {*

*//*properties

*propertyName: type = 'some value';*

*//methodNames...*

*}*

Next, we need to define the **template** for the above class.? How...? Using the **Metadata***.*

**Defining a Metadata with Decorator:**

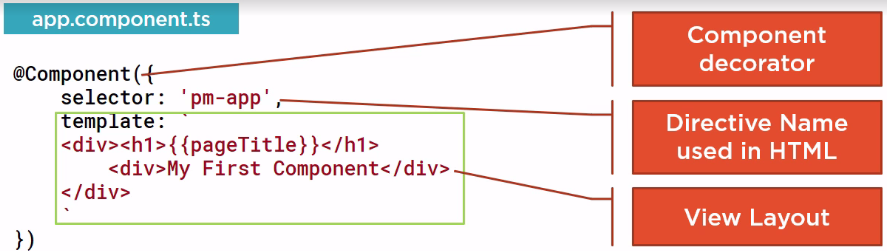
We define the Angular Metadata, with the Angular **Component function,** by attaching to the class **as a Decorator.**

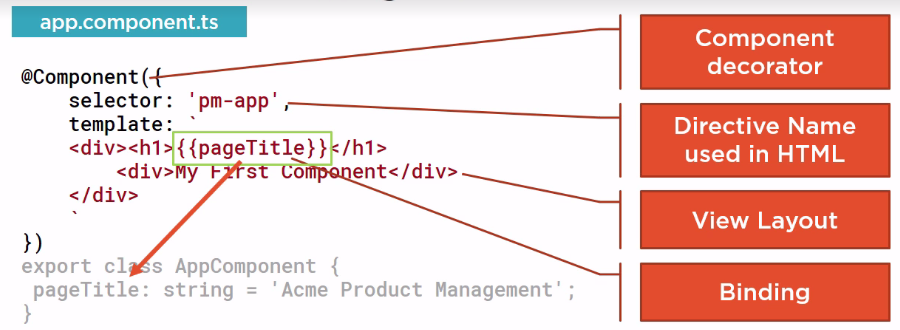
**Format:**

@Component( {objects

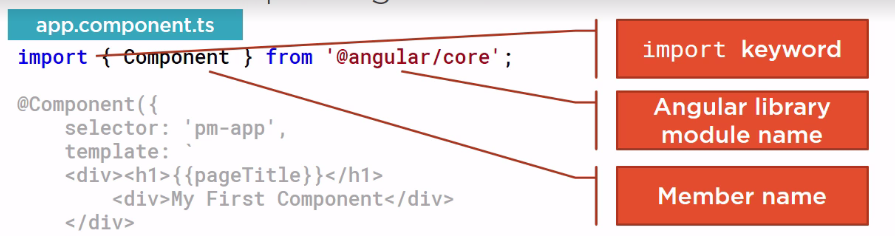
     Properties...}

)



**

**Importing**



Some Angular library modules:

**@angular/core**

**@angular/animate**

**@angular/htpp**

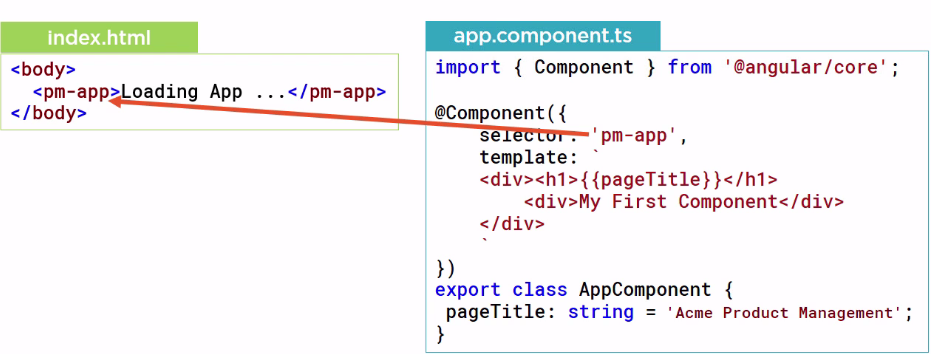
**@angular/router**

**...**

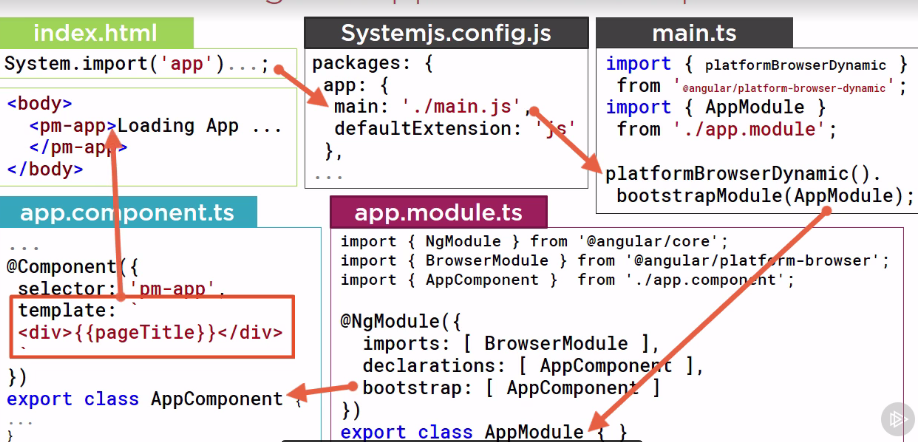
Other list of Angular library modules can be found at:

*http://www.npmjs.com/~angular*

8) Bootstrapping the application:



Bootstrapping our **app.component.ts**:



Where the 'bootstrap' array defines the **root component** of the application ie **AppComponent** .