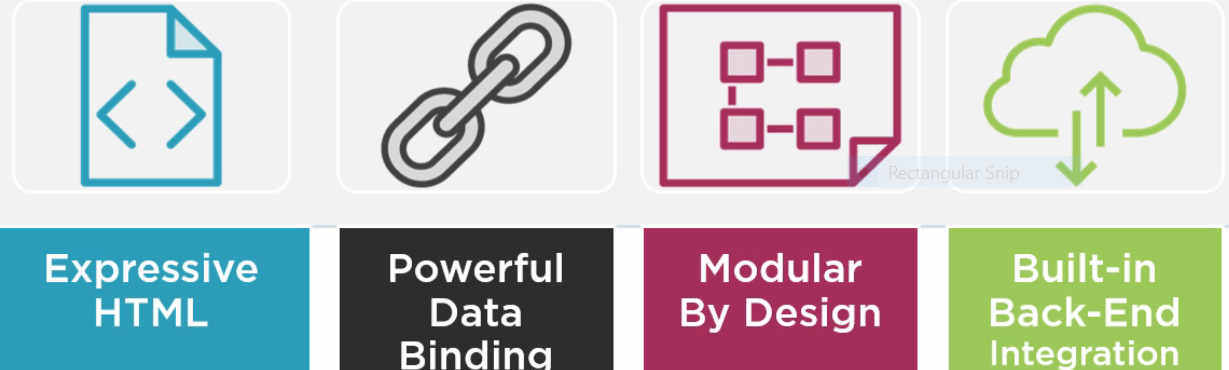
**Angular2**

https://www.youtube.com/watch?v=oa9cnWTpqP8

**Angular:**



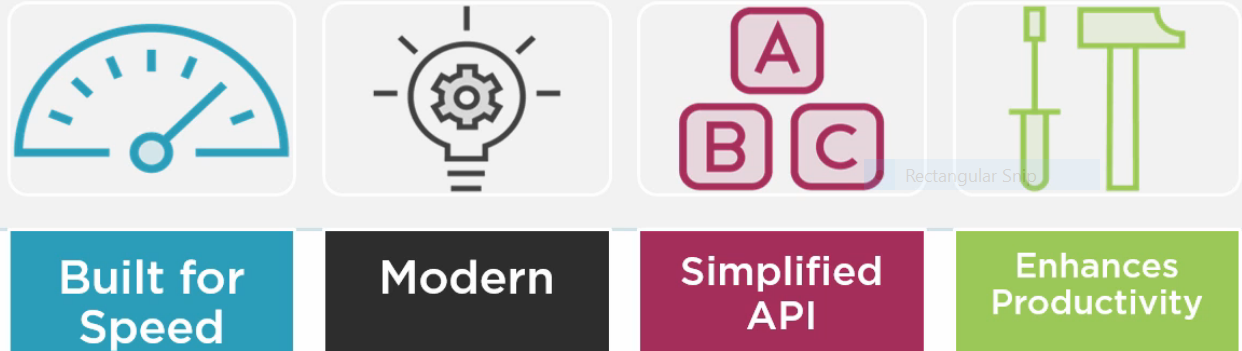
**Why Angular:**



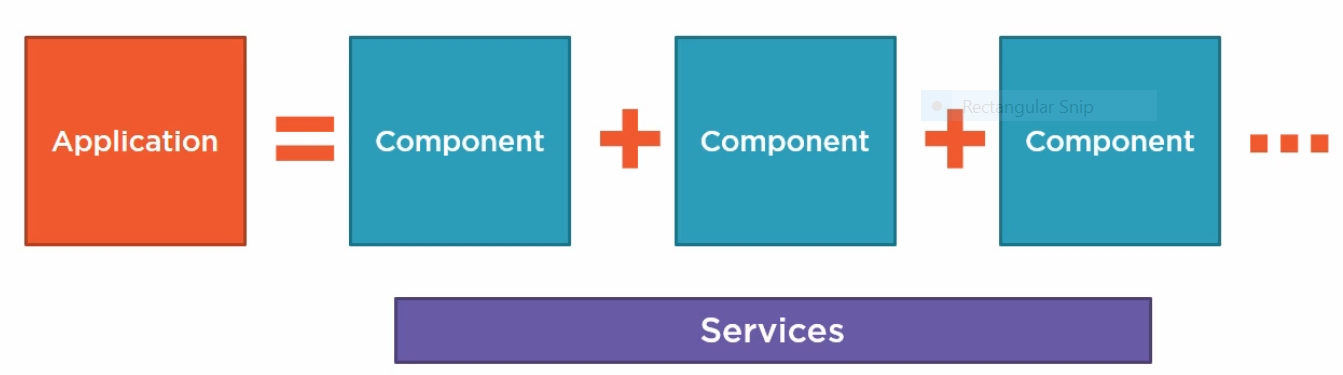
**Why Angular2:**

**Angular2 supports angular features in addition to that it supports below features**

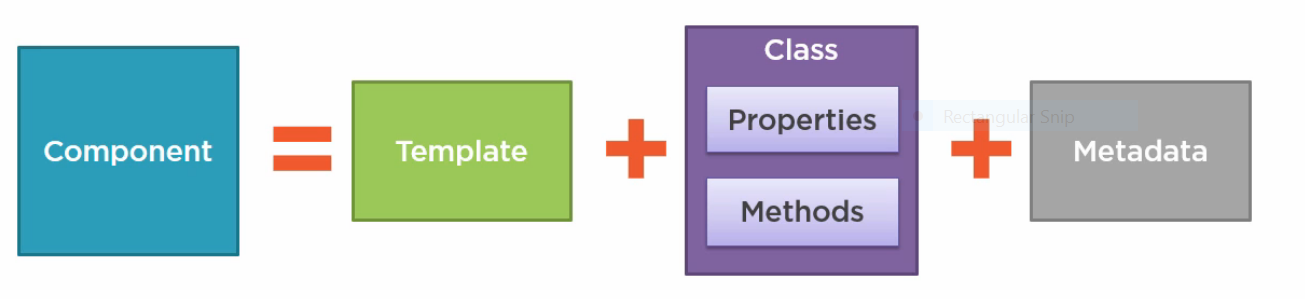
**It's fast, clean code**



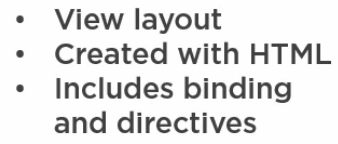
**OverView of Angular2 Application:**



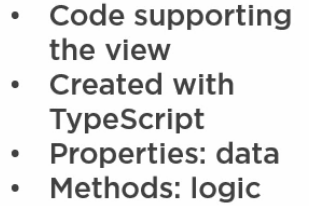
**Component:**



**Template: It contains HTML view.**

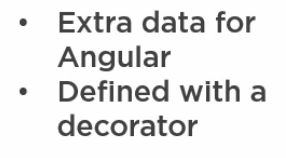


**Class: It contains associated code.**



**Typescript: It supports strongly type data types for script(JavaScript), So we call it as type script.**

**Metedata: additional information.**



**How we will pull using this component with an application?**

**We will use angular modules for combine all these components, every angular application module has at least one angular module. That angular module called as root module.**

**Application can have more then one angular module, these are included in feature angular modules.**



**-----------**

**Coding**

**starting point for angular2 is main.ts, from there will call Appmodule class.**

**AppModule is root module angular2 application**

**need to start coding from app.module.ts**

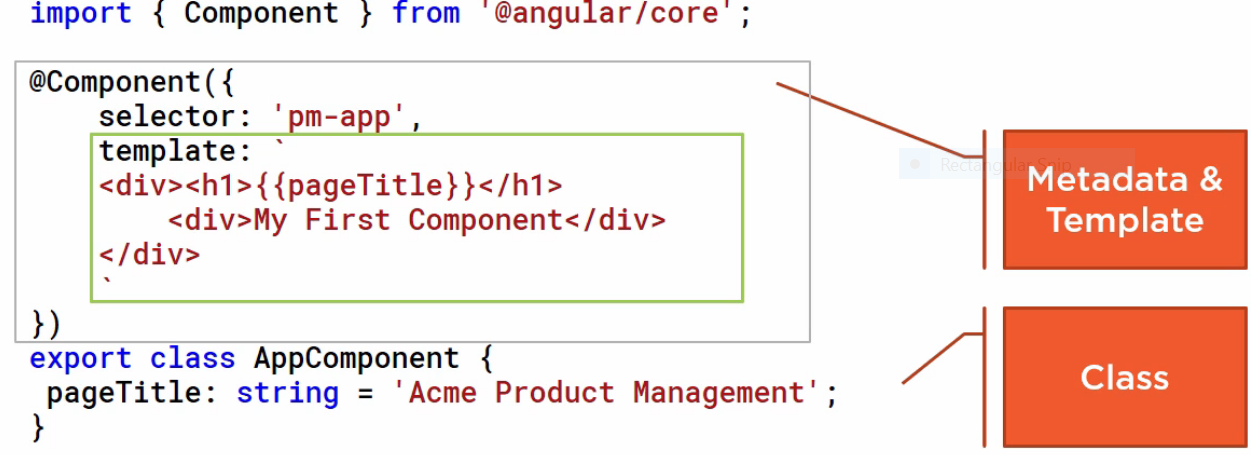


**import :predefined components or third party components**

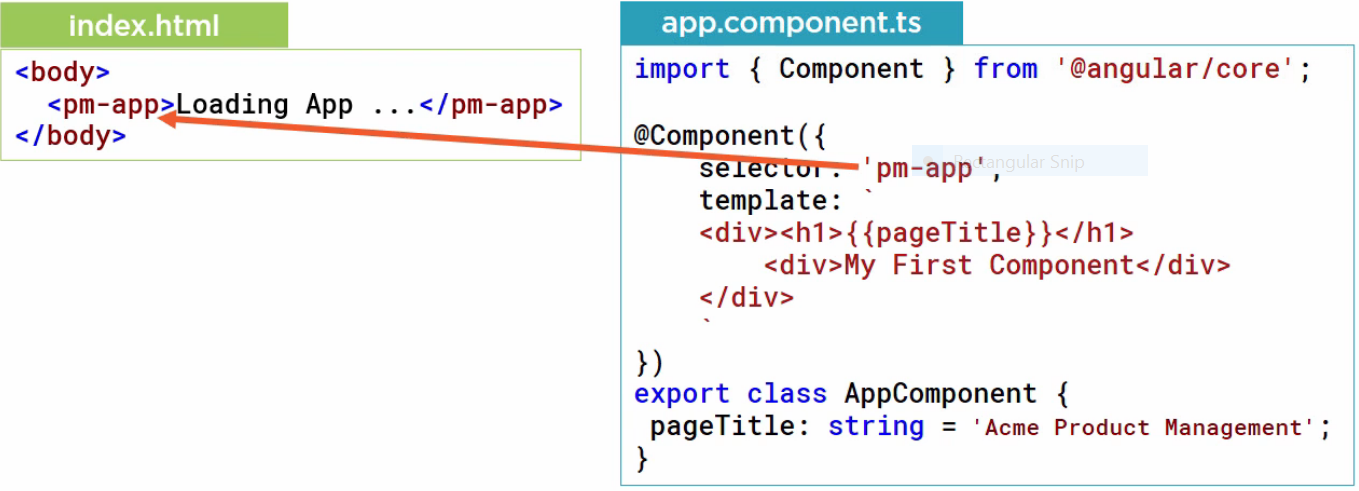
**declarations: It contains list of components which we are using in AppModule.**

**bootstrap: It indicates that which, application start from which component.**

**Components:**



**Selector pointing to index.html body**



**open project folder from vscode**

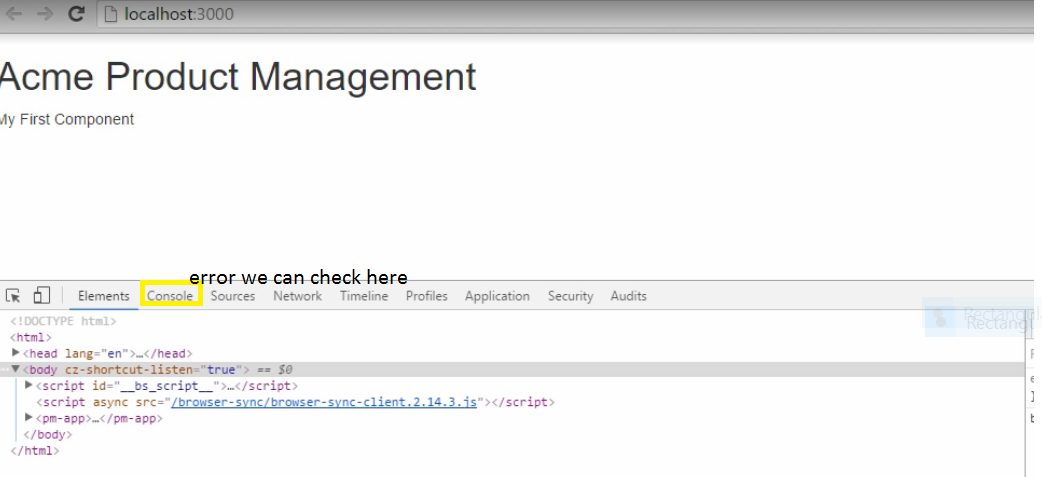
**we run project through command prompt**

**to run project we have use command 'npm start'**

**before that we have to insatll npm using command 'npm install'**

**for configuration application referes package.json file.**

**Index.html is main file;**



**Inside Component decarative,Template property contains html code. here html code we have write inside ``(backtick).**

**--------------------------------------**

**strcutural Directive: \*ngfor,\*ngIf**

**example: <tr \*ngFor="let item of items">**

**<td>{{item.name}}</td>**

**</tr>**

**syntax:**

**example: <div \*ngIf="color == 'blue'">Data is valid.</div>**

**in above example 'color' is property of class.**

**example:**

**<div \*ngFor="let id of ids">**

**Id is {{id}}**

**<div \*ngIf="id%2 == 0">**

**<div [ngClass]="'one'">Even Number</div>**

**</div>**

**<div \*ngIf="id%2 == 1">**

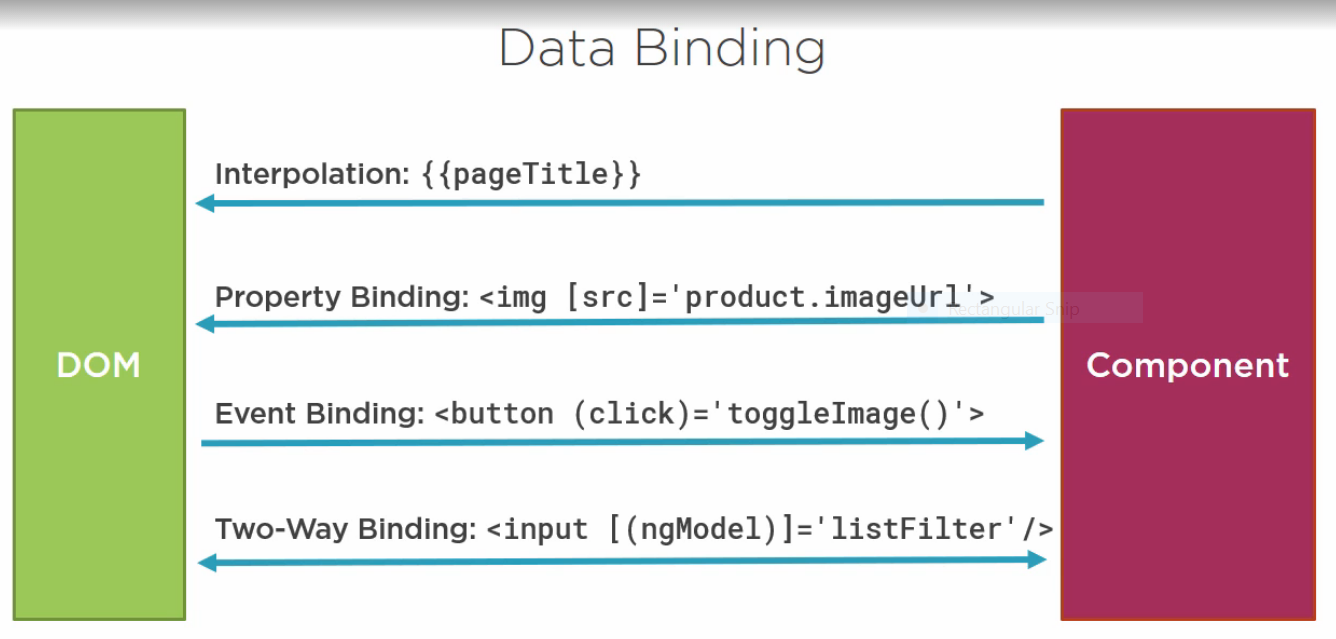
**<div [ngClass]="'two'">Odd Number</div>**

**</div>**

**</div>**

**---------------------**

**DataBinding:**



**DOM(browser)| \_\_\_\_\_DataBinding\_\_\_\_\_\_\_\_|Component**

**Types:**

**Interpolation --------> Component to DoM**

**One way Binding -------> Component to DoM**

**Event Binding -------> DOM to Component**

**Two way Binding -------> both ways**

**One way Binding (or) Interpolation:**

**[] brackets using for one way binding. inside bracket contains HTML property**

**<h3 [innerText]="story.Name"></h3>**

**Event Binding:**

**<button (click)="log('click')">**

**OK**

**</button>**

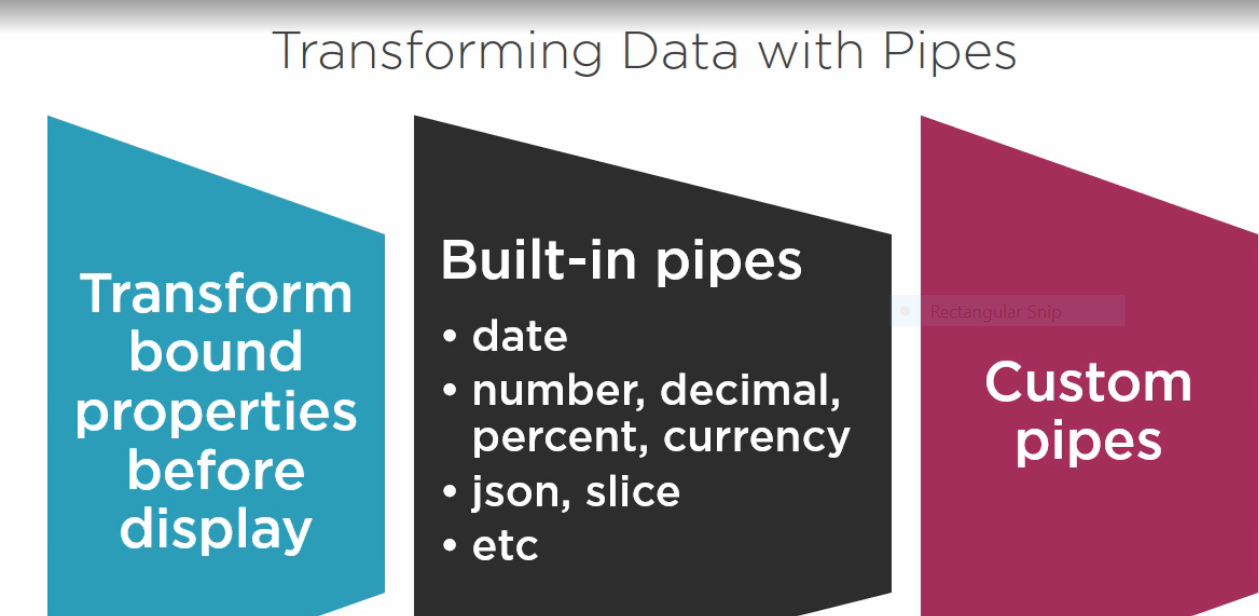
**here when you click button it calls log funtion.**

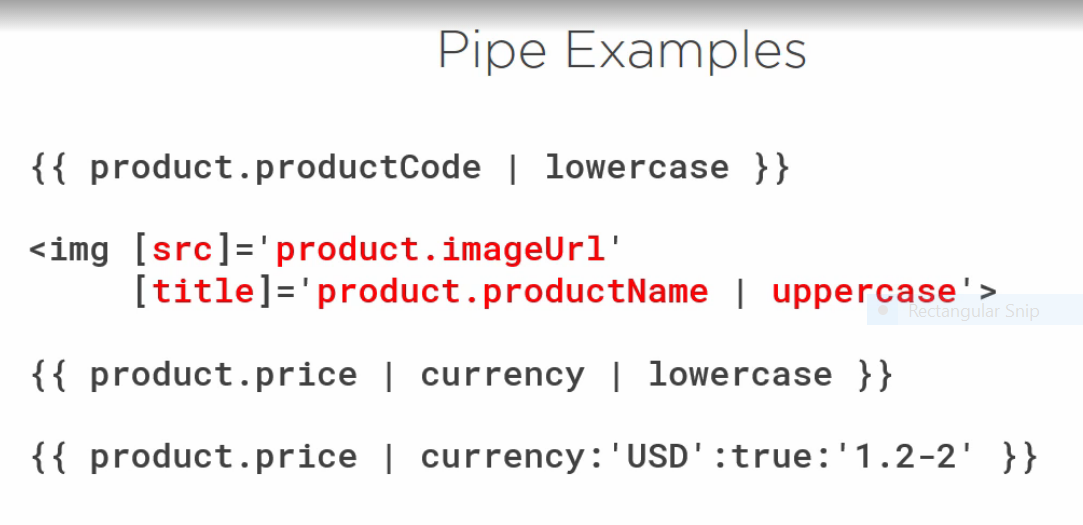
**Two way Binding:**

**[()] used for two way binding , ngModel directive we will use for two way binding**

**<h3 [(ngModel)]="story.Name"></h3>**

**Pipes:**

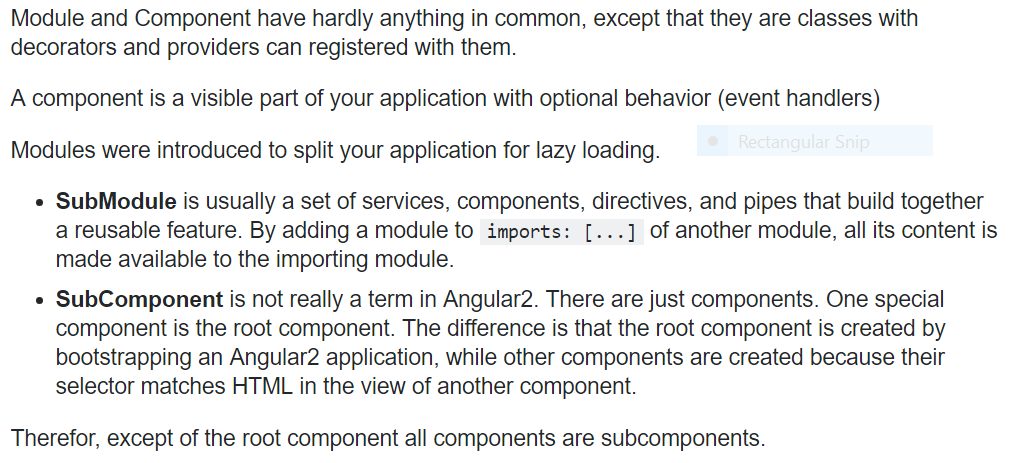




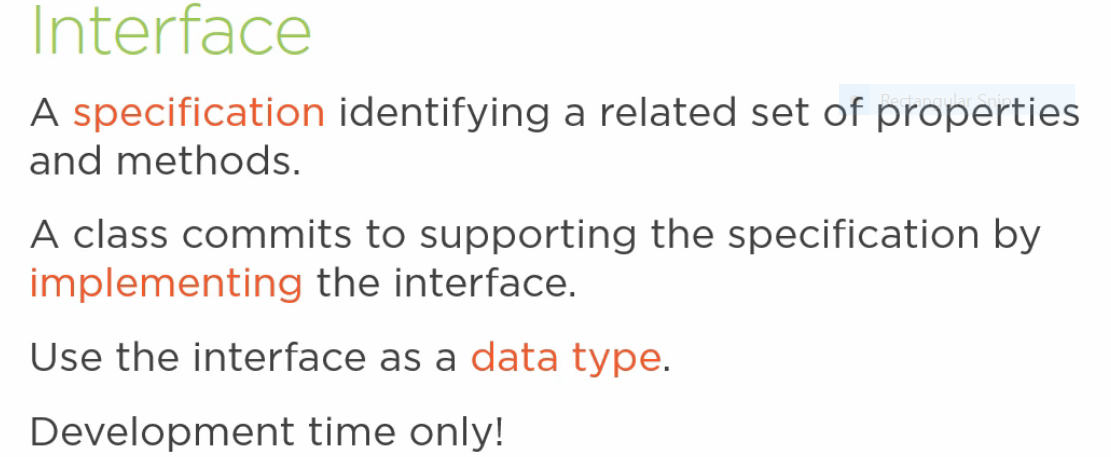
**------------------------------**

**More on Components:**

**Module: difference between module and component**

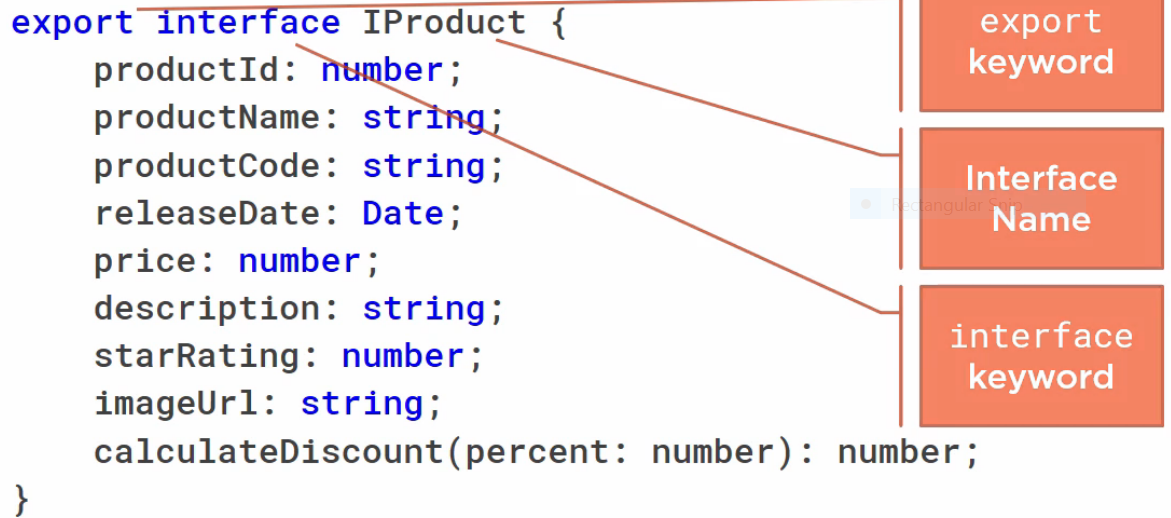


**Interfaces:**



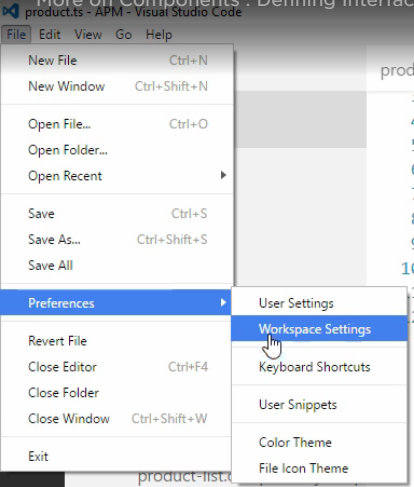
**We will use interface for develop strongly typed types.**

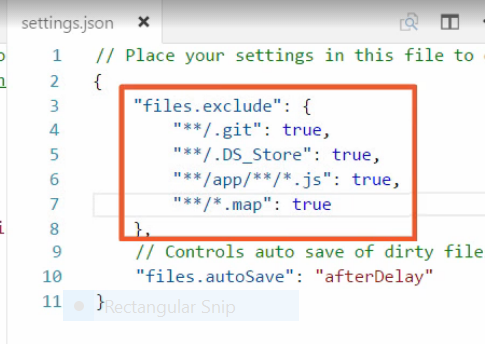
**Syntax:**



**Note: If we have any functionality or calculation changes we will use methods in interface,to implement this interfaces we will create classes. But if want only class structure we can use interface directly no need to implement class.**

**Note: .js,.map files we were not using here for coding purpose we want to hide those files in Visualcode we can set like below.**

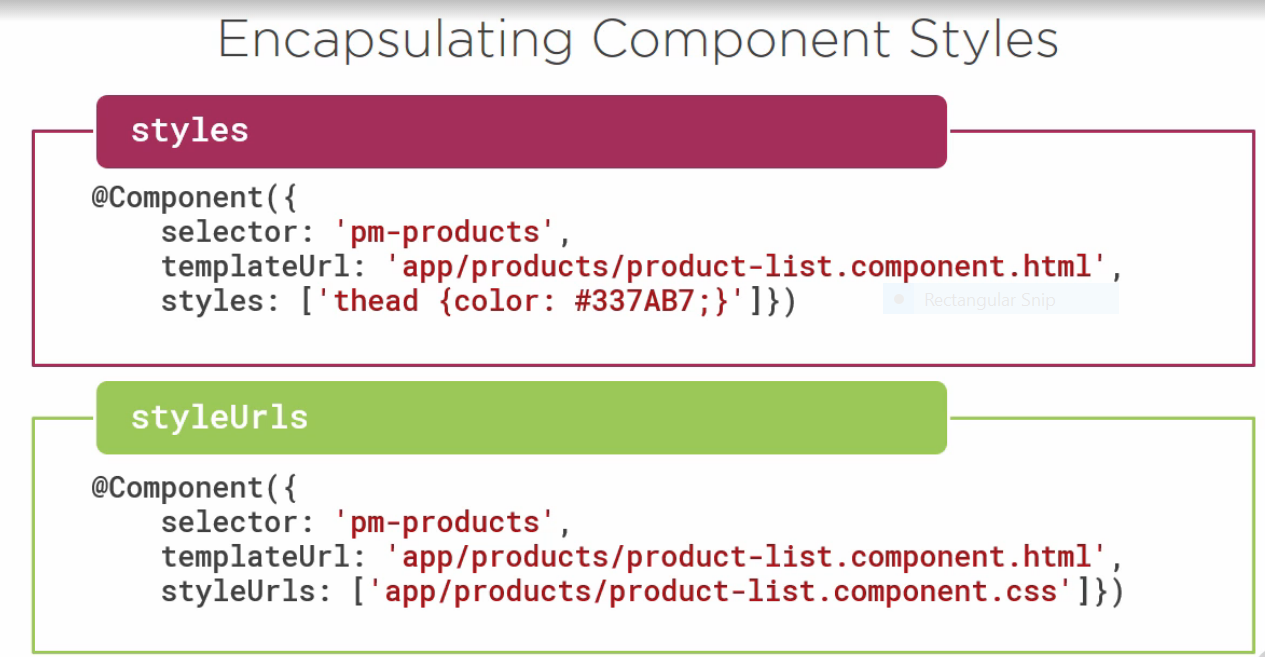




**Styles:**

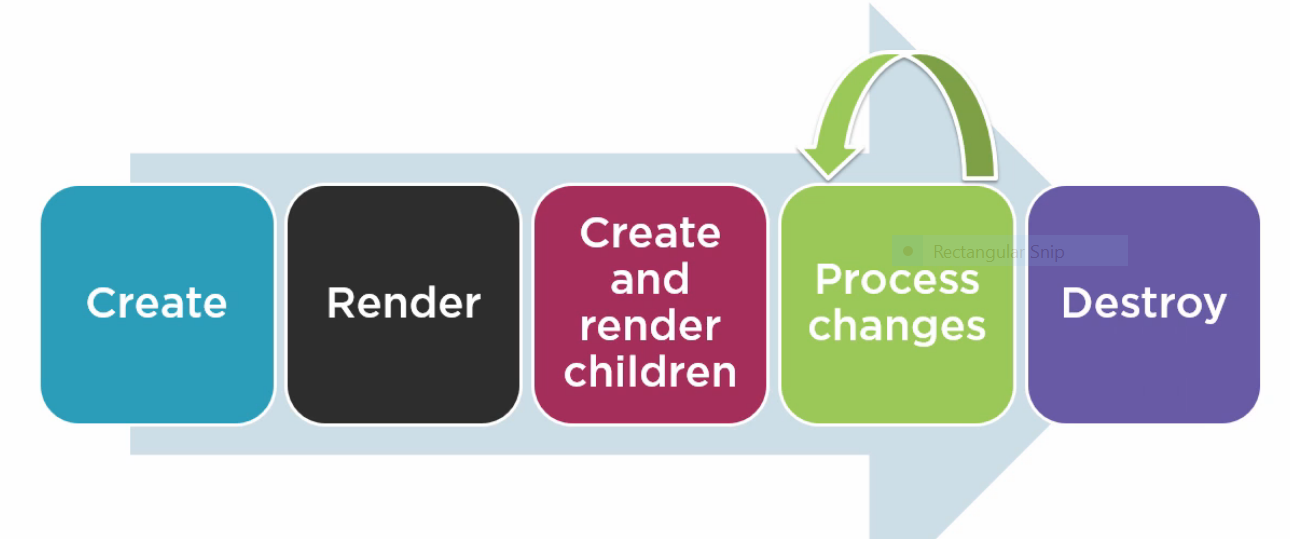
**css class if we want add for template we can use style and styleurl properties in side component decarator.**

**We can include more than one style sheets with comma separator.**

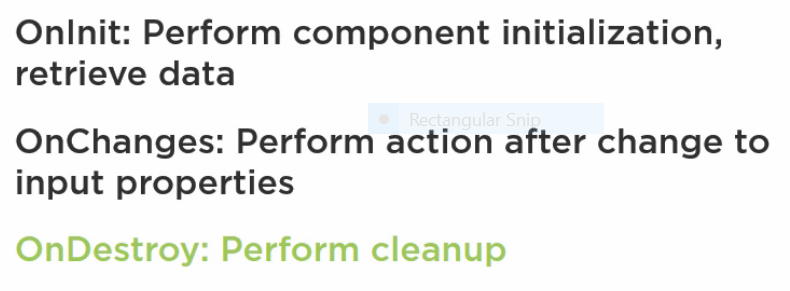


**LifeCycle hooks:**

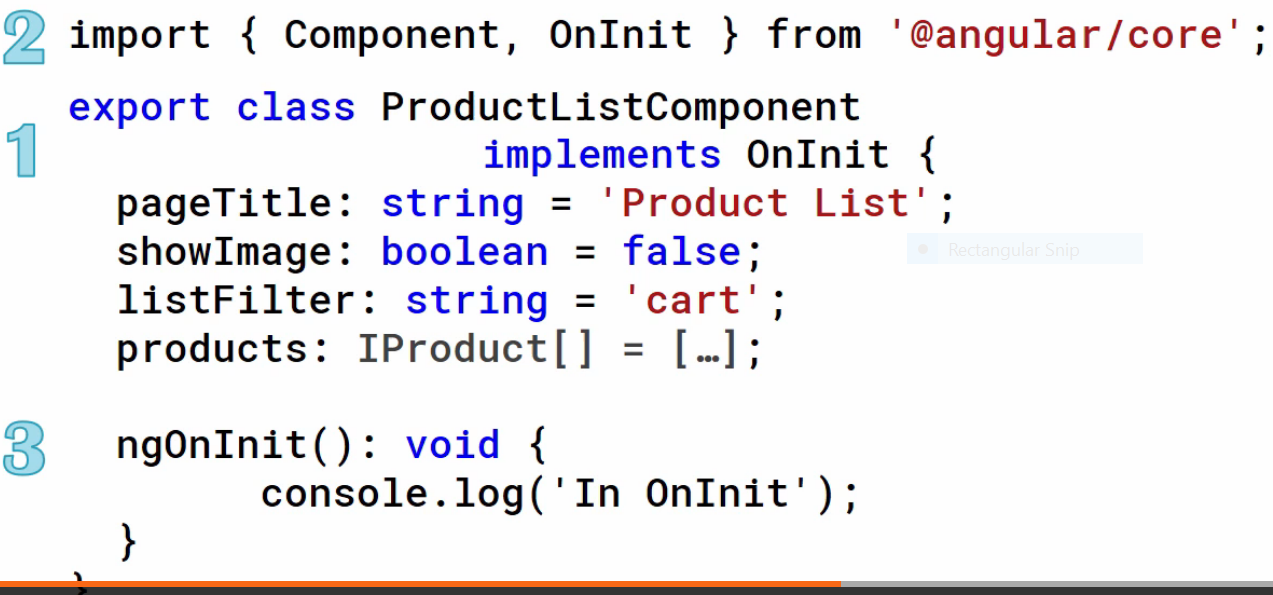
**Component lifecycle:**



**We can hook some functionalities in between above life cycle using below predefined interfaces.**



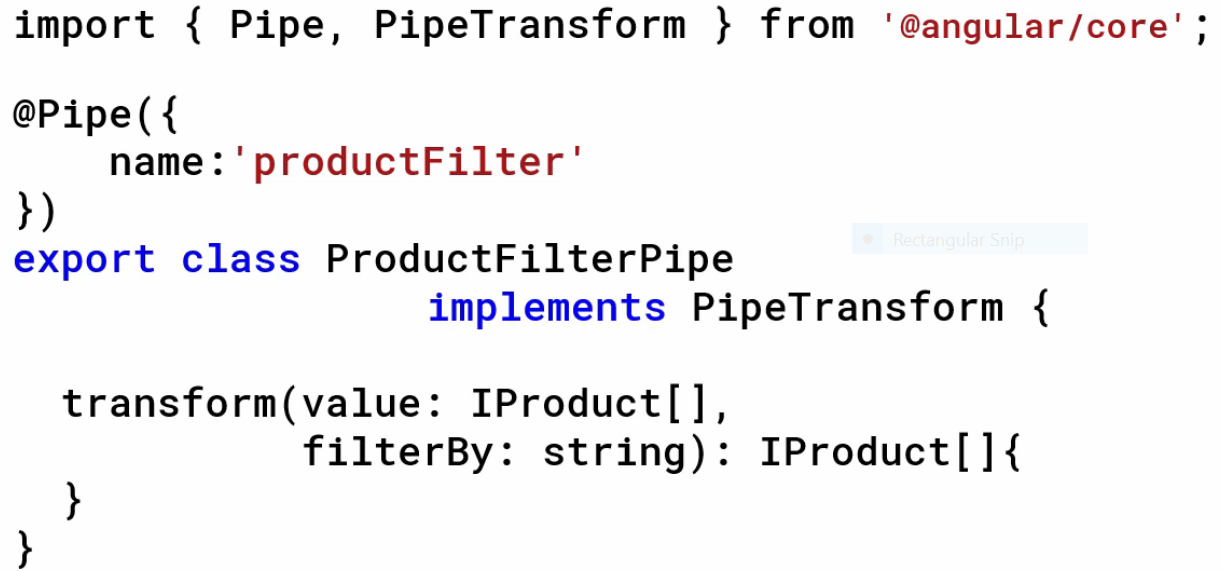
**Example:**



**Custom pipes:**

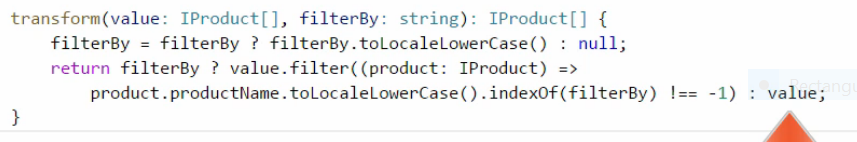
**PipeTranform interface we use for custom pipe development. This contains method called transform.**

**Example:**





**In transform method we can implement like below with condition**



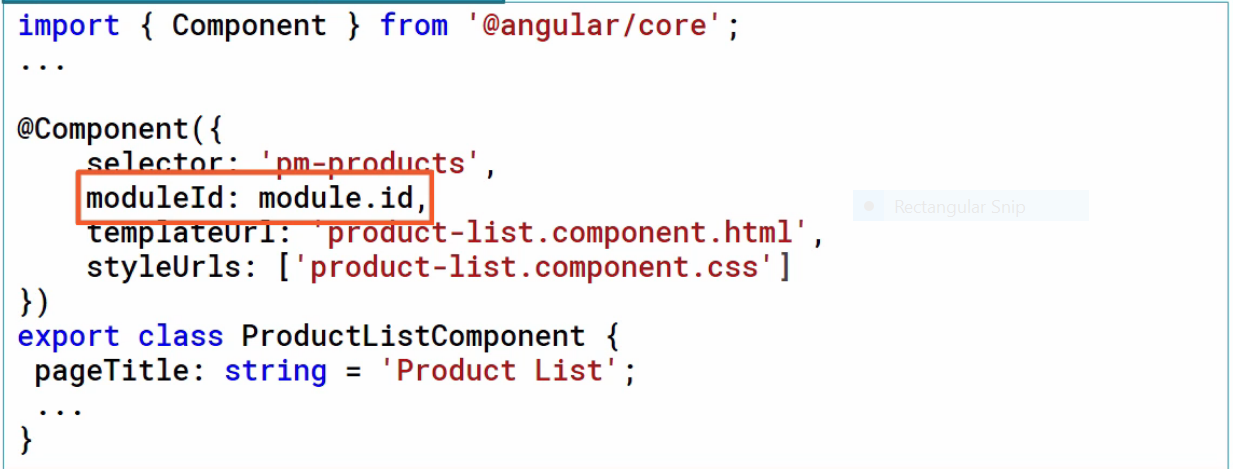
**Relative paths and ModuleID**

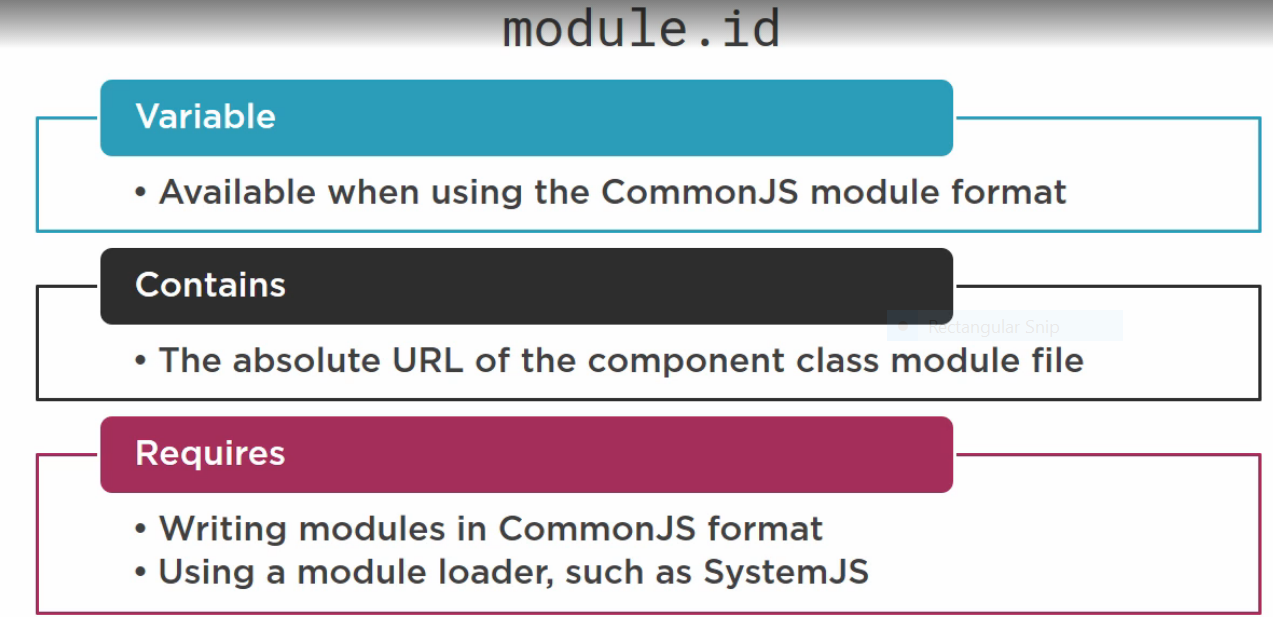
**Relative path we can remove when we use module Id, it automatically point out the relative path when we use module id.**

**Example:**



**In above example ‘’app/products..” is relative path, In below example we removed relative path hence we added module id.**



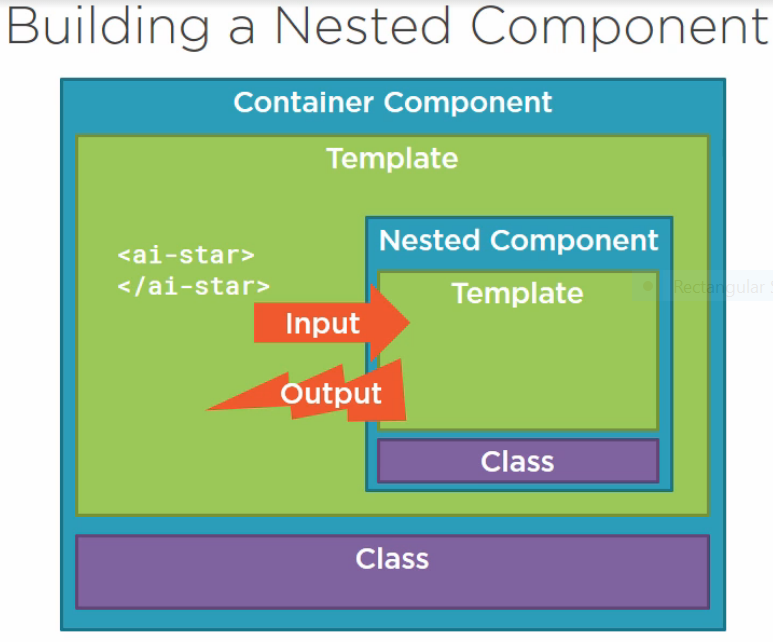


**Nested Components:**

**To make component as nested component ‘selector’ is required.**

**Syntax:**



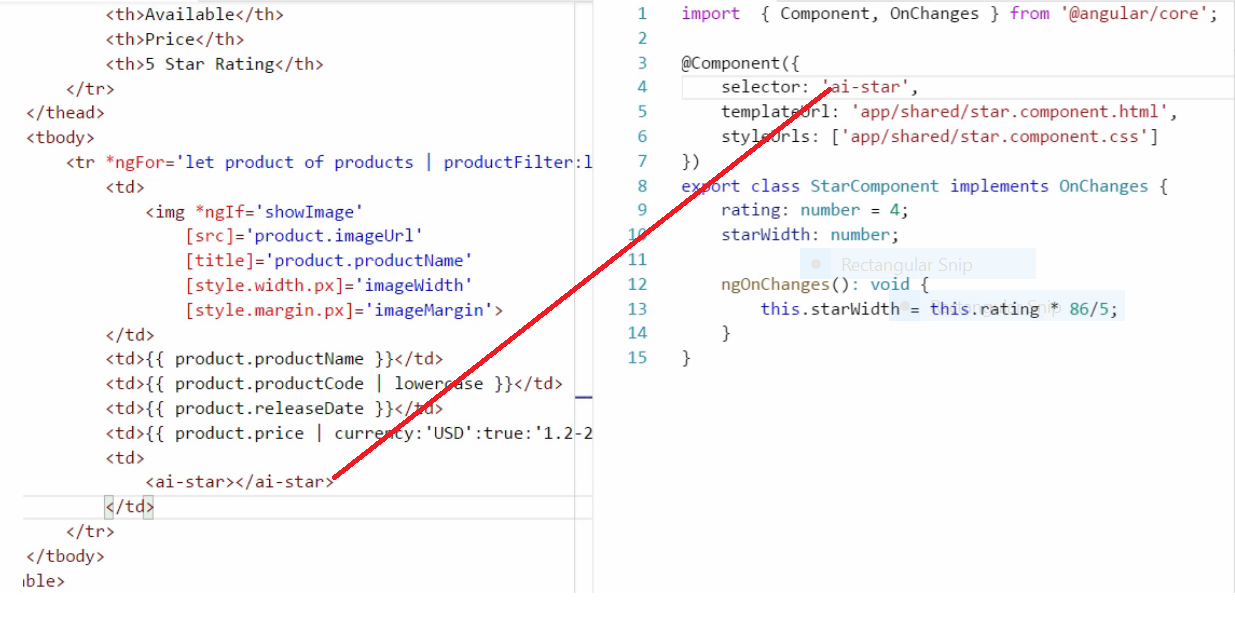


**Passing data to nested component we will use @input decorator**

**Getting data from nested component we will use @output decorator ,this datatype should be event.**

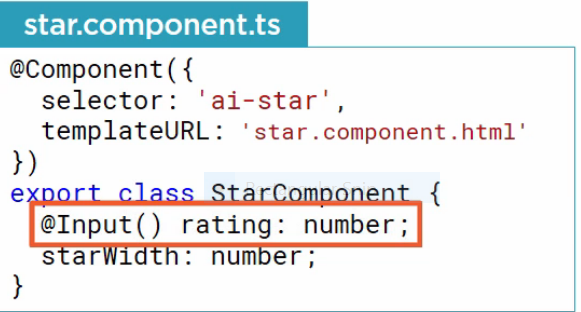
**In angular event defined with EventEmitter object, It’s generic object**

**If we want use nested component in any other module, we should declare nested component in their Appmodule.**



**To use input ,output decorator we have import these two from ‘@angule/core’**

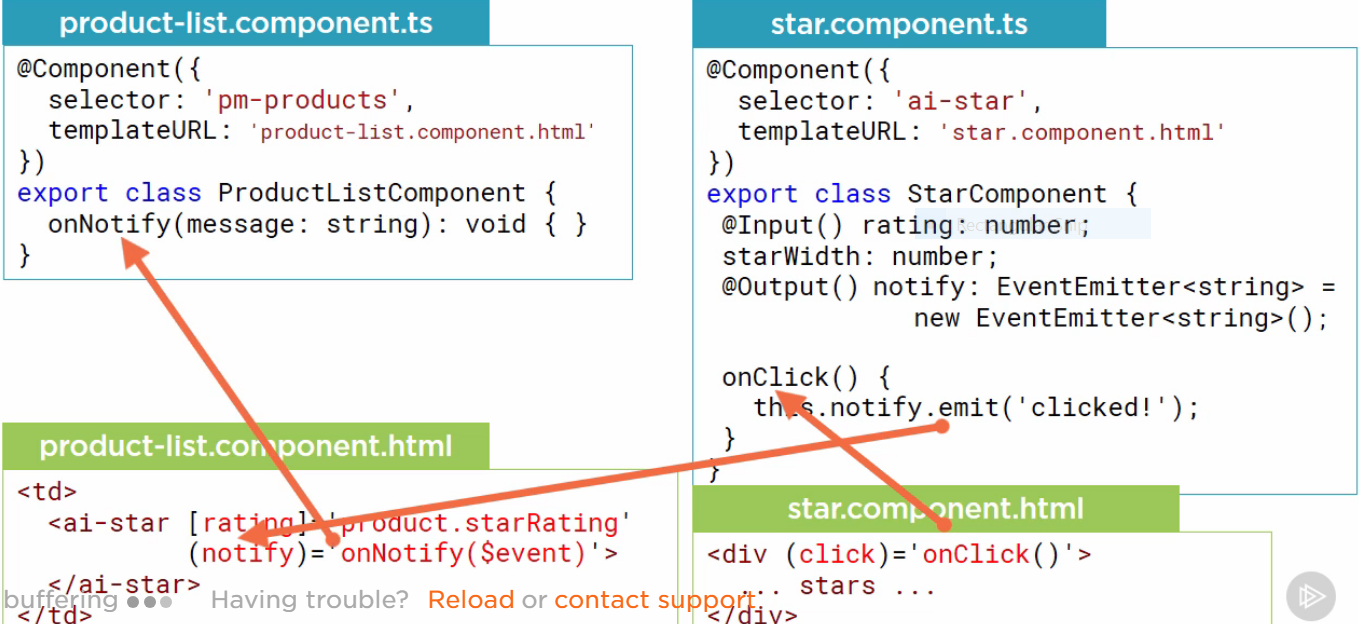
**Declaring @input decorator in nested component.**



**Passing value to nested component using input decorator.**



**@output decorator**



**We can pass output data through emit method of event.**

**In container using $event we can catch output data, since it an event we have to create method to fire this event. In above example onNotify is method.**

**Services**

**In angular 2 services implimentation is just like class**

**------------------------**

**Dependency Injection:**

**In AppModule class,Inside NgModule decarative contains providers property. This having list of services.If we register service under provider it automatically take care dependency injection**

**here no need to create Unity.Config file to map Interface with class.**

**After Register Service in AppModule class, In Component class Using Constructor we can Inherite Service.**

**Ex:**

**@Component({**

**moduleId: module.id,**

**selector: 'my-vehicles',**

**templateUrl: './vehicles.component.html',**

**})**

**export class VehiclesComponent {**

**vehicles = this.vehicleService.getVehicles();**

**constructor(private vehicleService: VehicleService) { }**

**}**