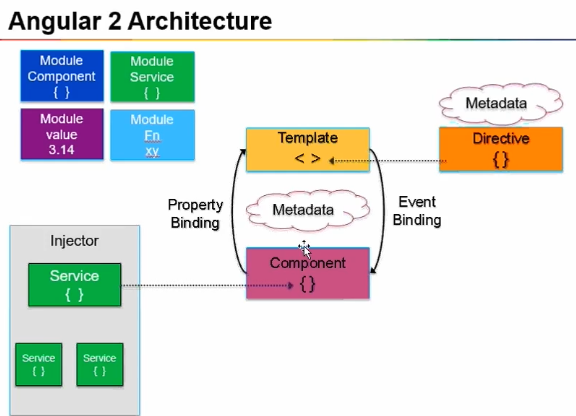
Angular 2+

Angular2 application has got 8 building blocks:

1. Modules
2. Components
3. Templates
4. Metadata
5. Data Binding
6. Directive
7. Services
8. Dependency Injection



The fundamental thing in Angular is Component. Developer spend most of their time in developing components. It is a javascript class.

Component = Component class + Metadata + template. Properties and Methods.

Template consists of HTML (view portion).

**Property binding:** Whenever any change happens in a **Component** property, same change is implemented in **Template** through Property binding.

**Event binding:** Whenever any change happens in a **Template** like if you clicked a button, same change is implemented in **Component property** through Event binding.

*Property binding (one way) + Event Binding (one way) 🡺 two way data binding*

AngularJS had two data binding by default. In Angular 2+ we mostly used one way binding.

**Metadata**: they provide some additional data about the component. Like Annotations.

Directive: Angular built-in and custom directed that we can create.

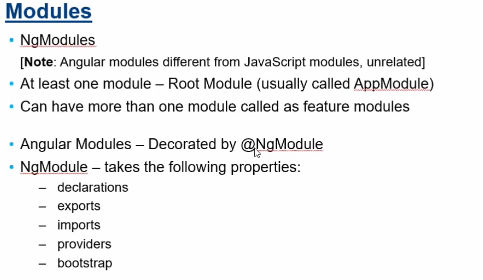
**Services** are supposed to contain all business & backend logic, connecting to server or pulling data from database all this should be handled by services. Components should use services to connect to server or database connection. Components are not supposed to contain business logic and should be independent of services. Component should just focus on UI part.

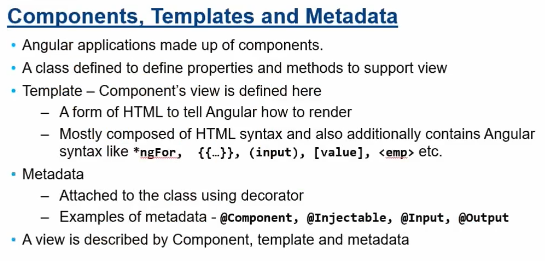
Service + Component => Module

Bootstrapping a module ---means--- > starting the application.

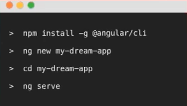
In angular there is one root module that loads other module. By default it is calles App.js module. Like main() function in Java, execution starts from main() function.

@NgModule declaratory is used to declare a module in angular. It tells that a particular JS class is angular module.





Angular CLI: starter application code

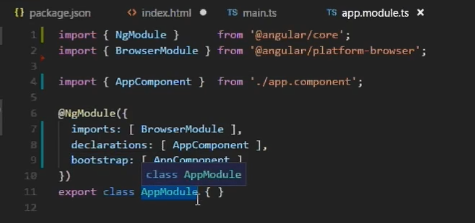


Ng serve is not production server, it is just a developmental server.

Ng server –open (this will start the web server, deploy the application on web server also). Default port : localhost:4200.

All the development is done in src - > app folder.

Command “tsc” creates new .js file of a .ts file. The newly generated .js file will not have type notations that we added in .ts file. Question comes - how does this makes sense that typescript is anyway going to be converted to regular javascript with no annotation. Answer to that is - when you make some error /warnings (like assigning number to a string variable) in you .ts file and try tsc command the you would see error on console saying number is not assignable to parameter of type string to a string variable. That is how transpolation throws error and tells that something is wrong with the code.



Bootstrap array: Bootstsrap component is starting component. This component invokes other components.

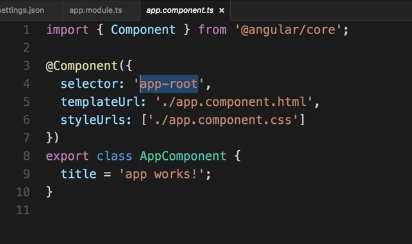
Therefore AppComponent is the top level component.

Browser module: Platform specific things arepresent In this module (like mobile or desktop brouser)

NgModule ({ imports : [..] }) – this module is to import only angular module while javascript import will import the physical file.

Providers: services are declared using providers.

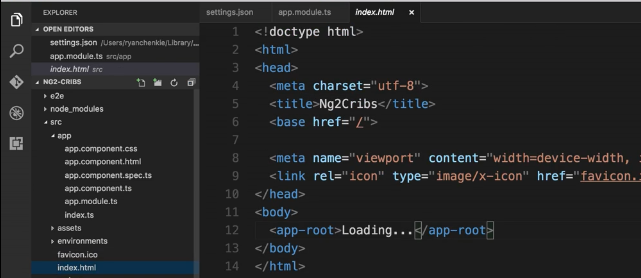
Main app component.



@Component takes an object that has configuration values for us. IN angular

**Selector** – app-root -- > this is name of this component. IN index.html file we can see that this selector name is used as an HTML tag.

<app-root></app-root>



templateUrl – to reference a view from another file.

styleUrls: to reference CSS style from another file.

Generate a new component

* Ng g component crib.listing (g is for generate)

This command will create a component with 4 files –

1. crib.listing.component.css
2. crib.listing.component.html
3. crib.listing.component.spec.ts
4. crib.listing.component.ts

and import it to app.module.ts file.

**LifeCycleHook**: ngOnInit

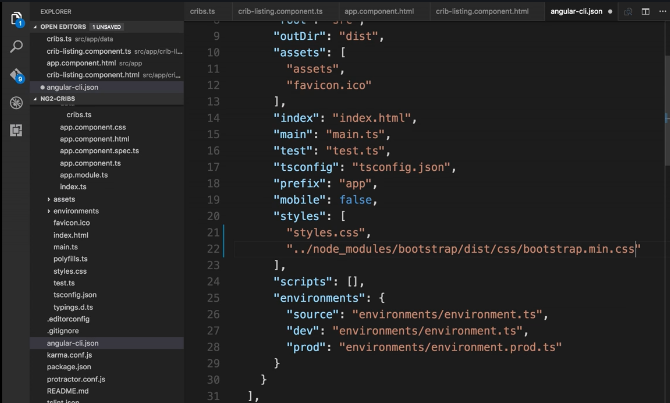
**noFor** is to iterate through an array. It is to repeat over data.

<div \*ngFor=”let crib of cribs”>

<h1> {{ crib.address }} </h1>

</div>

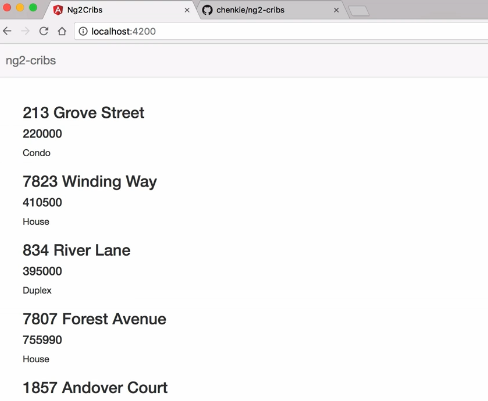
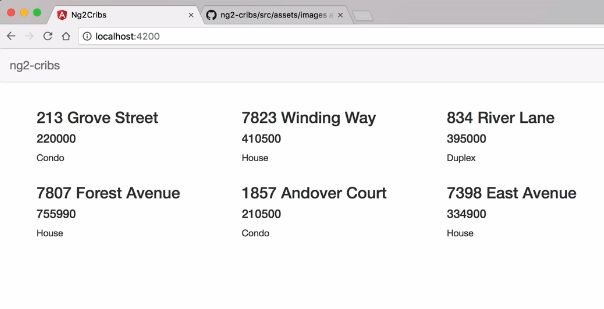
npm install bootstrap –save 🡪go to angular-cli.json 🡪 add bootstrap to styles array.



**Shortcut to write HTML in VS Code,**

Wirte element\_name.class\_name -- > hit Tab 🡪 example: nav.navbar.navbar-default -- > hit Tab 🡪 this’ll get converted to <nav class=”navbar navbar-default”>.

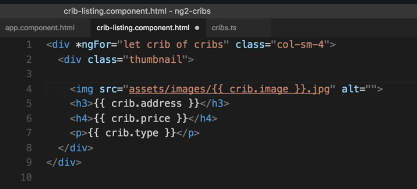
**Bootstrap class=”col-sm-4”** : <div \*ngFor=”let crib of cribs” class=”col-sm-4”> : col small 4 : this is going to tell this div that it should take up a third of the total 12 columns spaces across or page.

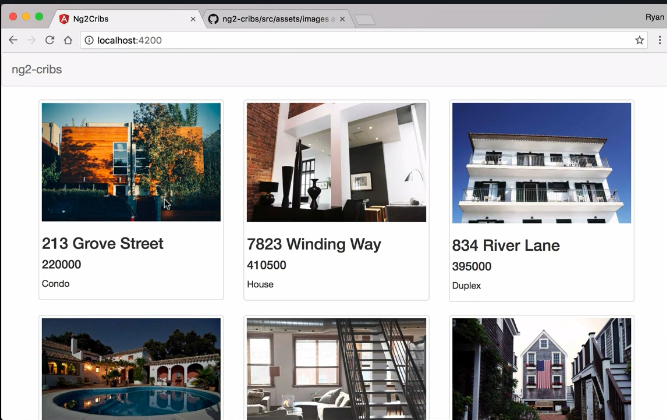
 ------------------------ > 

**Bootstrap Class=”thumbnail”**

This gives a border display. We can add CSS for this class as well. 

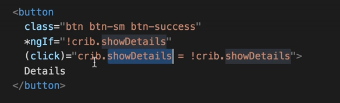
**Adding images:**





**\*ngIf directive:** it is used to conditionally show or hide.

To **add toggle of** true and false on button for click event.



**Currency pipe** and it takes an argument of whatever currency you want to use and the second argument here is whether or not we want to show the dollar sign. If we set second argument to false then instead of $ sign we would see “USD”

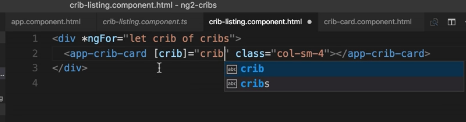


In HTML 4.01, the <i> tag was used to render text in italics. However, this is not necessarily the case with HTML5. Style sheets can be used to format the text inside the <i> element.

**Property Binding**

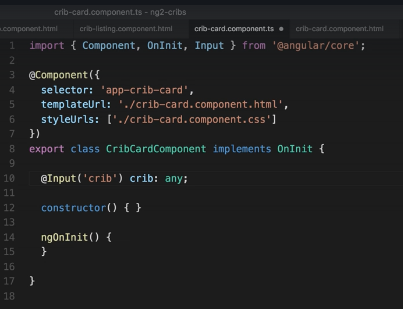
This is a feature of angular 2 that lets us pass data into components or some other things that it does too.

So we’re saying we want a property on this component called crib and we want to pass to it the current crib that we are iterating over.



To receive this data on the other end:

1. We have to take this binding as input -- > we go this Input class form @angular/core
2. Use Input as decorator -- > say we want to input as ‘crib’ and we want to assign it to a property called crib. Note: first crib is property passes and second crib is local variable



Interface

It is a class that is going to describe how something should be shaped and what kind of keys and values it should contain.

Command: ng g interface crib

This command get us a file created src/app/crib.ts

In image on right side: line 11: we say that property here needs to abide by the crib interface.