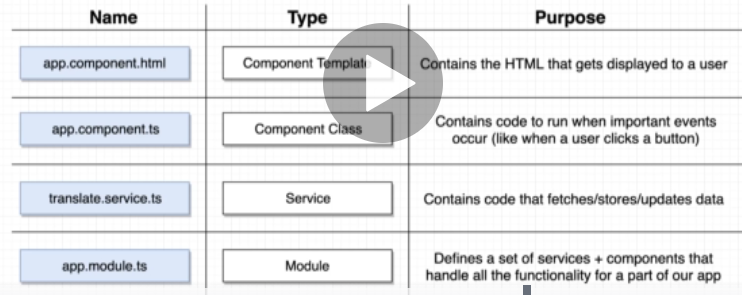
OHM SHRIMA AIM SHRIM SWAHA ANGULAR



>component - templateURL ---where we define our component , link our html template and

styleURLS----define whc style will applied to our component .

app.component.spec.ts –testing

index.html – first thing that gets sent to our browser when trying to access application.

app.component.ts looks where @component is called decoratoe.

When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.

The **HTML DOM** model is constructed as a tree of **Objects**:

**Module:**Module types: includes component , directive , pipes son on

**Domain** (wraps up all the component needed , implement one single feature) , ---------------**Routed** (same as domain , tied to route (displayed only when user is at some particular address)) **Routing** – Defines routing rules (shows todo list when user is at route /todos) ---------------**Service**–defines service that will be used in multiple parts of the app fetch/store/update data from ntwk.Data flows from service to component,its implemented as classes,its signle instance -**Widget** - defines some reusable components that will be used in multiple other module. (shared module)

**Shared module:** export component from module (eg : export divider from shared module) -------------import it in the other modules

<ng-content></ng-content> -- which is used in shared component ,

**To create project :**

go to directory -- open coomand prompt

type to create angular project > **ng new projectname --routing**

to run on browser > **ng serve** or **npm start**

**localhost:4200**

**npm install bootstrap** - to install any module

**ng g c anyname** -- to create component

**ng g s anyname** -- to create service

**ng generate class User** -- create User class

**ng g i payloads** --- create interface

**ng generate pipe anyname** – to create custom pipes

**ng generate directive anymane** – to create custom class which can be applied to any html tag

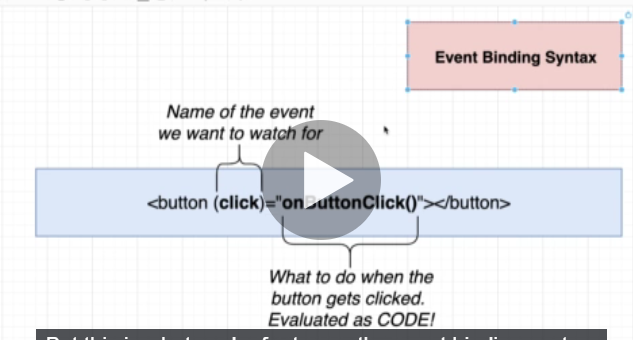
**ng serve --port 4401**

**for animation: npm install animate.css** ( add it in the angular.json file in the styles section – )

**ng g m anyname –routing** -- to create module (module is a collection of component service , pipes , directives so on)

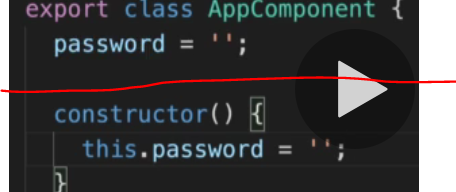
**Event Binding::**

onButtonClick() is a function



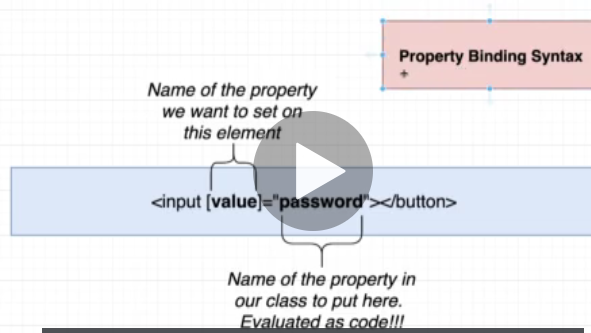


Standard HTML events <https://www.w3schools.com/jsref/dom_obj_event.asp> link is here .



Red line below and above code 100% equivalent to each other.

**Property Binding Syntax:**



Html elements are such as div,input all the tags property value can be changed like this. Where password is a variable which defined and set the value inside ts component . instead we can call function which returns the value.

Event binding example:

(input)=”onInputChange($event.target.value)”

(change)=”onChange()” //present in <input>

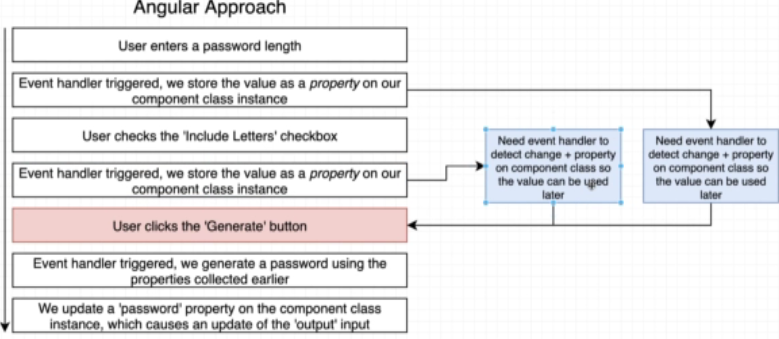
(click)=”onclick()”

[disabled]=”can pass any Boolean/any function / validation”

[class]=”call any function evaluate any value and return according classname , in css we can define properties”

**Interpolation syntax :**

{{password}} or {{nonvoidFun()}}– this prints the value directly on browser



Traditional JS approach : directly get info from DOM

Angular: get info out of events .

**Bulma** is free , open source css framework based on flexbox

**Directives :::** structural( which change the structure) , attribute ( which changes the properties) types:

**Structural** dir: add/remove html element – eg: <div \*ngIf=”length”> hide the legth if it false</div>

**Attribute** dir: changes the properties of the html gets applied to

Empty is treated as false in js

**ngIf** – if assigned value is empty/false then its invisible visa versa .

“ ” - whatever inside the double quote is treated as code and variable

Steps to deploy application on now.in (its totally free)

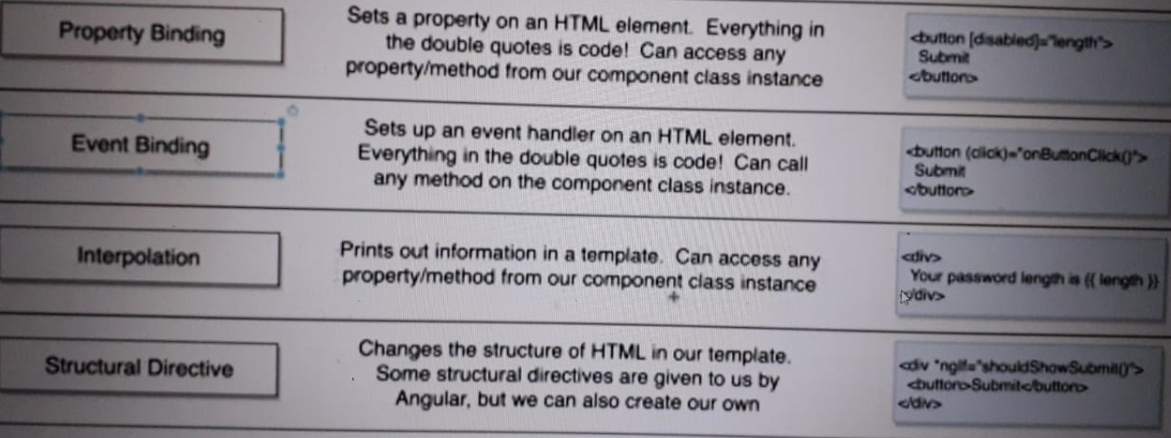
COMMANDS : now login 🡪 now ( to deploye)

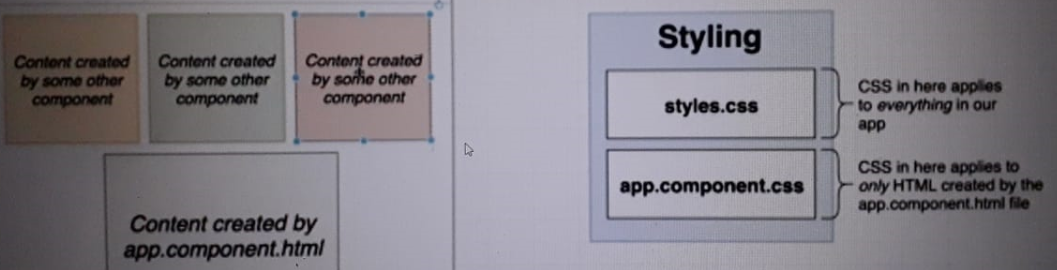
Custom structural directives : import TemplateRef (which refer the element -div/ anything), ViewContainerRef (refer the element which is inside the element which is refered by templateref such as ul (which present inside the li)) ,Input – those to ref used to customize the arguments that are accepted into the constructor,. Times is defined in the directive that many times ViewContainerRef is created and passed /displayed inside the TemplateRef.

Or ElementRef in this refer the elements means tags (div ,li)

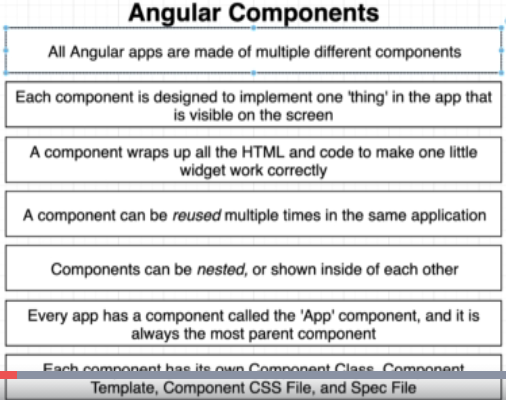
<hr /> -- divider , <br /> -- line braek

**Recap in one snap::**

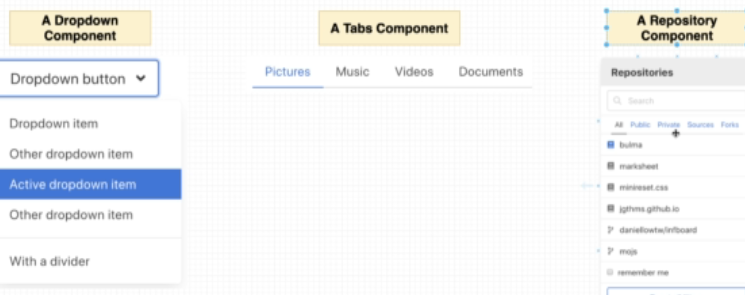


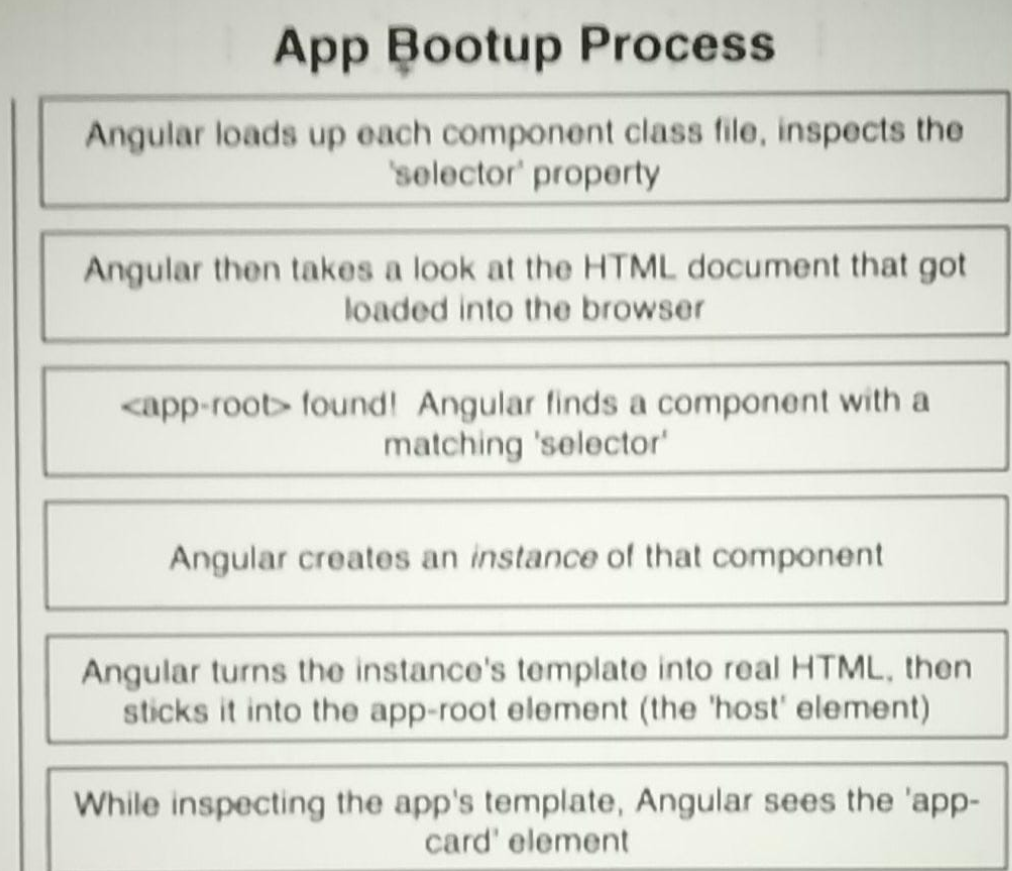


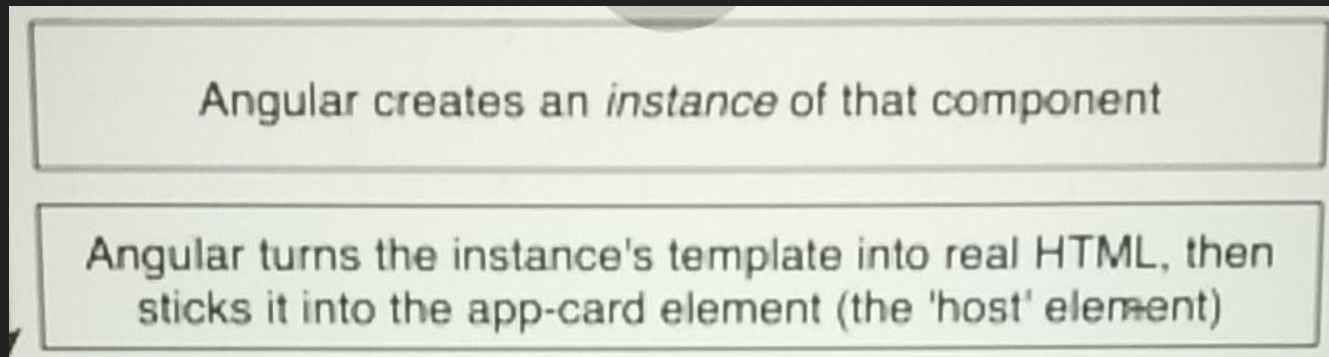
**COMPONENTS**::



EXAMPLES OF COMPNETS:







**Communication between parent and child components**

App template :<app-card [title]=’post[0].title’ ><app-card> //property binding

Card-class : @Input() title=’’;

Card-template: {{title}} //interpolation

\*ngFor=”let i of posts” // iterate through array posts or we can iterate the method which return array

Event binding in input ( passing input value to ts) – (input) = “getOnClick($event.target.value)”

**Pipes**: functions to format the data for use in a template . only used in template

**Href** is the basic attribute provided by Html to navigate through pages which reloads the page on click. **routerLink** is the attribute provided by angular to navigate to different components without reloading the page. -- <a routerLink=”/elements”>Elements</a>

**Router**-**outlet**  works as a placeholder which is used to load the different components dynamically based on current route state/address. Navigation can be done using **router**-**outlet** directive and the activated component will take place inside the **router**-**outlet** to load its content.

If any parameters want to pass from parent module to child component:

Create data in main module component class pass it through template class , in the child import Input and define passed data with no value @Input() data=[]; , iterate the data value in child template class with \*ngFor =”let post of data” then access with {{post.value}}

How to create route module connect to main template :

1st:add link to the header app template file.

2nd:in routing module add the path (lazing loading)

3rd:generate other module and Home component which is to be called

4th:add a rule to new module routing to show home component

5th: show modal(other components of module) in home component by tag <app-modal> (modal is one of the component of module)

Life Cycle Methods:

**ngOnInit**:called once after this component is 1st displayed on screen &after set properties passed down from parent component . (its method of OnInit interface). To used app-module as parent ElementRef is imported. And define instance of ElementRef in constructor.

Document.body.appendchild(this.el.nativeElement); ////el is a instance of ElementRef

**ngOnDestroy**:to remove this component(when we navigate to other route) which is refering the main app-module. Below statement is defined in this method.

This.el.nativeElement.remove()

**ngOnChanges**:called anytime a property of component is changed( eg: when parent pass new data)

**communication btw parent and child component : passing something from child to parent:**

import : output , EventEmitter

child: @Output close = new EventEmitter(); // emitter instance of close event

create one method to emit the event (to happen event close/anything) this.close.emit(); // along with emit() method we can send info to parent

to stop something progation from parent : (eventName)=”**$event.stopPropagation()”**

**ng-container** is used instead of **div**  when to stop creating a unnessary space, when inside content is empty.

[ngClass]=”{active:i===openedIndex}”

Communicating btw service and main component:

Service is a class , define a method (eg:search() it should be public to access in main component) , create instance in main component(passing in constructor) , access method by using object in onsearch() method.

**ISSUES::**

# [Could not find module “@angular-devkit/build-angular”](https://stackoverflow.com/questions/50333003/could-not-find-module-angular-devkit-build-angular) -- Solution : npm install -> npm update -> ng serve

# [Cannot find module 'http' or its corresponding type declarations while I try include ytdl-core](https://stackoverflow.com/questions/64713636/cannot-find-module-http-or-its-corresponding-type-declarations-while-i-try-inc) – remove http import

# [“Port 4200 is already in use” when running the ng serve command](https://stackoverflow.com/questions/39091735/port-4200-is-already-in-use-when-running-the-ng-serve-command) (command to kill) netstat -ano | findstr :4200 🡪 taskkill /PID 27868 /F

# If you get any syntax error which is not valid – go to view- command pallete – reload window

[Cannot find module '@angular/compiler'](https://stackoverflow.com/questions/42585663/cannot-find-module-angular-compiler)- >>

npm uninstall angular-cli ->npm install @angular/cli --save-dev

You can see this errors by typing **F12** (Developer Tools) in your browser and checking the console.

**TYPESCRIPT::**

“TypeScript is JavaScript for application-scale development.”

**Data types:** strng,number,Boolean,undefined,null,any

A TypeScript program is composed of −

Modules

Functions

Variables

Statements and Expressions

Comments

**Const add: (a:number):number=> {return a+2};** // where :number to ref return type its optional

**Interface** post{

Title:string ; desc:string;

} // can define proper structure for post

Can define constructor inside the class.

**Constructor(private color:string,private brand:string){}** // it will automatically assign the value

**npx typescript –init >** to support decoration

built-in types : number,string ,boolean,void , null, undefined

var msg:string=”hello”;

User-defined types include Enumerations (enums), classes, interfaces, arrays, and tuple.

Type Assertion(casting): var str2:number = <number> <any> str

Convert string to array --- **randomtext.split(‘’);**

Iterator – **let letter of letters** – letters should be array

Decorators: called when first time executed , not when instance of class created . it can be applied to class/property/method/accessor /argument of method .. receives diff arguments depending on where it gets used , can be a plain / factory decorator .

Const parseVaue=parseInt(value);

**Generic type of class:**

Class ValueHolder<T>{ value : T };

Generic type of Functions:

Const valueWrapper = <T>(value:T): T[] =>{ return [value]; }

var x = 123e5;    // 12300000  
var y = 123e-5;   // 0.00123

isNaN(x);

numbers methods::

var x = 9.656;  
x.toExponential(2);     // returns 9.66e+0

x.tostring();

var x = 9.656;   
x.toFixed(0);           // returns 10

var x = 9.656;  
x.toPrecision(2);        // returns 9.7

var x = 123;  
x.valueOf();            // returns 123 from variable x

parseFloat() parses a string and returns a number. Spaces are allowed. Only the first number is returned:

String methods::

var sln = txt.length;

var pos = str.indexOf("locate");

var pos = str.lastIndexOf("locate");

var pos = str.indexOf("locate", 15);

var pos = str.search("locate");

* slice(*start*, *end*)
* substring(*start*, *end*)
* substr(*start*, *length*)

join() method also joins all array elements into a string.

var x = fruits.pop();   // Removes the last element

var x = fruits.push("Kiwi");   //  the value of x is 5

fruits.shift();    // Removes the first element "Banana" from fruit

fruits.unshift("Lemon");    // Returns 5 adds at beginning

var myChildren = myGirls.concat(arr1,arr2);  //concatenation of 2 array

fruits.sort();

fruits.reverse();

iteration:

numbers.forEach(myFunction);

var over18 = numbers.filter(myFunction);  
  
function myFunction(value, index, array) {  
  return value > 18;  
}