Angular Notes

===========

Commands

ng new AngularComponent --no-standalone

ng serve

ng g c mycomponent

npm install --save bootstrap@latest

Component is combination of

Component.css

Component.html. – View template

Component.ts. files.

Default component is app.component

A screenshot of a computer screen

Description automatically generated

When we run angular app, index.html file gets rendered in the

1. webpage. This is the main HTML file whose content will change when we

navigate around or do some other stuff on webpage.

1. An angular app consists of component and by default angular CLI

provides us with one App Component.

1. Each component has fours important files component.ts.

component.html, component.css & component.spec.ts file.

component.ts is the main component file.

1. HTML file of a component is called as its View Template.

Main.ts file is the entry point of the Angular application. Main module is the apple module and bootstrap the main module.

Style.css file is used to mentioned the global style sheets in the application.

add global styles to this file, and also import other style

angular.json file is the file contains all the configurations of the project.

**Why use TypeScript?**

1. TypeScript is a free and open-source programming language developed by Microsoft. It is a superset of JavaScript.
2. TypeScript has additional features that do not exist in current version of JavaScript supported by most browser.
3. TypeScript is strongly typed. But JavaScript is dynamically typed.
4. It has object oriented features which we do not have in interface, access modifiers, fields, properties, generics
5. With typescript, we can catch errors at compile time.

**Template property:** Template is used to remove the template-url property in the typescript file and write html code in the place of template-url. We need to use single quotes to write html code in sing line. If multiples lines required need to use ` ` back stoke instead of single quotes.

**Style Property:**  Style property is used to write inline styles instead of external style sheet url file instead of using style -url use styles property

**Selectors:**

1. We can use a sector like an HTML tag.
2. We can also use a selector like an HTML attribute – need to use [] in the component selector
3. We can also use a selector like a CSS class. Need to use . in the component selector.

**Data binding :**

Data binding in Angular allows communicating data between a component class and its corresponding view template.

A close-up of a white background

Description automatically generated

A screenshot of a computer screen

Description automatically generated

* **One way databinding :**

One way data binding is when we access the component class property in its corresponding view template

String interpolation {{data}}

Property binding [property] = data

A screenshot of a computer

Description automatically generated

One way data bind is also when we access the value from view template to corresponding component class property.

This can achieve by event binding in angular.

A white background with red text

Description automatically generated

**String interpolation :**

String interpolation is in angular is bind the date form component class to view template. That means data flow between component to view.

**Property binding:**

Property binding is used to binding the property of HTML element to the property or method of the component class.

A screenshot of a computer

Description automatically generated

**Event binding:**

Event binding is allows to us to webpage event to component property or methods. Using event we can pass the data from view to component.

**A text on a white background

Description automatically generated**

* **Two way data binding**

Two way binding is binds the data to component class to view template and view template to component class . this is combination of property binding and event binding.

A screenshot of a computer

Description automatically generated

Directives**:** Directive are simply instruction to the DOM.

* **Structural directives:** Changes the view of a webpage by adding and removing DOM element from the webpage.

**ngFor**

1. ngFor is a structural directive . that means it miniplate the DOM by adding and removing the elements from the DOM.
2. ngFor directive is used to repeat a portion of the HTML template once per each item from the irritable list.
3. If it is structural directive we have use \* before the directive.

**ngIf**

1. ngIf is structural directive.
2. ngIf directive is used to add and remove the element from the webpage based on the given condition.
3. If the condition assigned to ngIf returns true it will add an element on which is used to the webpage. Otherwise if the condition is false it will remove the element from the webpage

* **Attribute directives:** Used like an attribute on existing webpage element to change its look and behaviour.

**ngStyle**

1. ngStyle is an attribute directive, that means ngStyle changes the look and behaviour of the HTML element.
2. The ngStyle directive is used to set a CSS style dynamically for an HTML element based on the given typescript expression.

**ngClass**

1. ngClass is an attribute directive.
2. ngClass directive is used to add the CSS class dynamically to the webpage element.

**Custom property binding**

We can pass the data from parent component to child component using @Input decorator.

We can also called it is custom property binding because here we bind the custom properties of child component class with property or method of Parent component class.

**Template Reference variable**

A Template reference variable is a reference to any DOM element ,component or directive in the template**.**

Need to use #myvariable to create a template reference variable.

**@ViewChild decorator**

The @ViewChild decorator is used to get a reference to a child component or an element within a parent component. It allows the parent component to access and interact with the properties, methods, and events of the child component or element.

**View encapsulation**

The View encapsulation is concept or behaviour in angular ,where component CSS styles are encapsulated into the components view and don not effect to the other components or rest of the application.

There are 3 types of view encapsulation

1. ViewEncapsulation.None
2. ViewEncapsulation.Emulated (default)
3. ViewEncapsulation.ShadowDOM

**ng-content**

The ng-content is used when we want to insert the content dynamically inside the component that helps to

Increase the component reusability.

Using ng-content we can pass the content inside the component selector and when angular parses the content that appears at the place of ng-content directive.

**Angular Life cycle hooks**

Angular life cycle hooks are methods that angular invoke on directives and components as it creates ,changes and destroys them

* 1. When the angular application-starts, it first creates and renders the root component. Then, it creates and renders its Children's & their children. It forms a tree of components.
  2. Once Angular loads the components, it starts the process of rendering the view. To do that, it needs to check the input properties, evaluate the data bindings & expressions, render the projected content etc.
  3. Angular lets us know when these events happen using lifecycle hooks. For Example:
     + **ngOnInit** when Angular initializes the component for the first time.
     + When a component's input property change, Angular invokes **ngOnChanges**
     + If the component is destroyed, Angular invokes **ngOnDestroy**.

**What is Change detection cycle?**

Change detection is the mechanism by which angular keeps the template in sync with the component.

**Projected content:** Projected content is that HTML content which replaces the <ng-content>

directive in child component.

**Input bound properties**: These are those properties of a component class which is decorated

with @Input() decorator.

**@Input() message: string**

**Constructor of a Component**

1. Life Cycle of a component begins, when Angular creates the component class. First
2. method that gets invoked is class Constructor.

Constructor is neither a life cycle hook nor it is specific to Angular. It is a JavaScript feature. It is a method which gets invoked, when a class is created.

1. When a constructor is called, at that point, none of the components input properties are updated and available to use. Neither its child components are constructed. Projected contents are also not available.
2. Once Angular instantiates the class, it kick-start the first change detection

**Angular Life Cycle Hooks**

**ngOnchanges**

**A screenshot of a computer

Description automatically generated**

**ngOnInint**

**A screenshot of a computer

Description automatically generated**

**ngDoCheck**

**A screenshot of a computer

Description automatically generated**

**ngAfterContentInit**

**A screenshot of a computer

Description automatically generated**

**ngAftterContentChecked**

**A screenshot of a computer

Description automatically generated**

**ngAfterViewInit**

**A screenshot of a web page

Description automatically generated**

**ngAfterViewChecked**

**A screenshot of a computer

Description automatically generated**

**ngOnDestroy**

**A screenshot of a computer

Description automatically generated**

**@ContentChild**

**Custom Attribute directive:**

**A diagram of a brand

Description automatically generated with medium confidence**

**Renderer2**

The native element property contains the reference to the underlying DOM object which gives us direct

access to the DOM, bypassing the angular. This is not advisable for following reasons:

Angular keeps the Component & the view in Sync using Templates, data binding & change detection, etc.

All of them are bypassed when we update the DOM Directly.

DOM Manipulation works only in Browser. You will not able to use the App in other platforms like in a web

worker, in Server or in a Desktop, or in the mobile app, etc. where there is no browser.

The DOM APIs do not sanitize the data. Hence it is possible to inject a script, thereby, opening our app an

easy target for the XSS injection attack.

**@HostListner**

The @HostListner decorator listens to the DOM event on the host element and it reacts to that event by executing an event handler method.

@HostListener('mouseenter') onMouseOver(){

*this*.renderer.setStyle(*this*.elemet.nativeElement,'margin','5px 10px');

*this*.renderer.setStyle(*this*.elemet.nativeElement,'padding','30px 30px');

*this*.renderer.setStyle(*this*.elemet.nativeElement,'transition','0.5s');

}

**@HostBinding decorative**

The @HostBinding decorator binds a host element property to a variable in a directive or a component.

@HostBinding('style.backgroundColor') background:*string* ='transparent';

**Structural directives**

A structural directive manipulate the DOM by adding and removing the elements to or from the DOM on which we are use it.

ngSwitch structural directive.

**Services**

A service is a reusable typescript class that can be used in multiple components across Angular application.

* Services are easier to debug and test.
* Services provides re-usability of code.
* With services, we can communicate across different components.

**Angular dependency injector**

Dependency Injection is a technique in which a class receives its dependencies from external sources rather than creating them itself.

We have to use **Providers** to inject the service without creating instance.

**providers**: [Enrollservice]

**Hierarchical injections**

When we provide a service on AppComponent, that service will be instantiated and injected for AppComponent and all its child component and their child component. All of them will receive the same instance of that service.

the lowest level of service providing |s on the component with no child component. If we provide a service there, this component will have its own instance of that service.

