Angular 5

Directives and Components

1) What are the Directives? Explain attribute Directive?
Ans: - Directives are instructions which tell angular to do something.
- Directives allow you to attach behavior to element in the DOM.

2) How many types of Directives are there in Angular 2?
Ans: - There are 3 types of directives:
1. component 2. Structural 3. Attribute

3) How events are attached to Directive?
Ans: By using @HostListener event-listener

4) How do we pass external data into Directive?
Ans: by using @Input DATA Binding

5) What are structural Directives? Name few Angular built-in structural Directives?
Ans:

1. Structural directives are responsible for HTML Layout.
2. The shape or reshape the DOM'S structure, typically by adding and removing or manipulating events.
3. Example: <div *ngIf="hero">{{hero.name}}
4. ngIf inserts DOM element if the condition were true. or else removes element from DOM.

HTTP and Routing

6) How Routing works in Angular 2?
Ans:

1. Routing is mechanism which enables user to navigate between diff views or component.
2. Angular has single instance of router server. (lazy loading)
3. Route Guards:
route guard is an interface method that routes on to check after authorization.

7) we can use selectors in component by 3 ways:

1. by using custom tag: <app-test><app-test>
2. by making class for selectors: selector: ".app-test" by preceding '.'
3. by making attribute using [] symbol: selector: "[app-test]"

8) Interpolation {{ }} :

1. we can bind data to template.
2. can invoke methods in interpolation: {{getData()}}
3. in ts: getData () {return "Hello" +this.name;
4. can Access JavaScript properties and methods
5. can't use assignments and can't access js global variable.

9) Attribute Vs Property [Property Binding]

1. attribute and property are not same
2. attributes are defined by HTML
3. properties defined by DOM (Document Object Model)
4. Attributes initialize DOM Properties and then they are done. Attributes values can't change once they are initialized.
5. Property value however can change.
6. Attribute specifies initial value whereas Property specifies current value.
7. DOM value property is a current value.
8. Value attribute remain same but Value property gets changed.
9. Attribute values can't work with Boolean values (like <input disabled="false"></input> cant set disable input element) whereas property values can change values or enables (like <input [disabled]="false"></input>)

10) Class binding:

1. When we use class attribute (class="") and class binding ([class]), class attribute becomes dummy attribute in the presence of class binding. you have to use one or the other not both.
2. To use conditionally apply multiple classes use [ngClass] Directives
3. Directives is nothing but a custom html attribute that angular provides.
4. ngClass allows dynamically to add or remove classes to html element based on certain user interaction.
5. [ngClass] Ex:

```
ts =>        public hasError=true;
        public msgClasses= {
                "text-success": !this.hasError;
                "text-danger":! this.hasError;
                }
html=>  <div [ngClass]="msgClasses"></div>
```

11) Style Binding:

1. style binding is used for inline styles for html element. Ex: [style. color] ="'red'"
2. in style binding possible to use conditional binding;
   Ex: html : <h2 [style. color] ="hasError? 'red': 'green' ">Style</h2>
   ts : public hasError=false;
3. can also component class properties during binding:

Ex: html: <h2 [style. color] ="{{highlight}}"> Style</h2>
ts : public highlight="blue";

4. to apply multiple styles, we use [ngStyle] Directive:

Ex: html: <h2 [style. color] ="{{styleMessage}}"> Style</h2>
ts : public styleMessage= {
color: "blue",
fontSize:12,
fontStyle: "italic"
}

12) Event Binding:

1. Captures any DOM element and perform prescribed action.
2. (click)="onClick ()";
3. (click)="onClick($event)"; to get information of event itself. (Information about Click Event)
4. $event gives all the information about the DOM Event that was raised.

13) Template Reference Variable:

1. to easily access DOM Element and their properties, Angular provides templateRef variables.
2. can access DOM Element by using # symbol followed by variable. ex: <input #name type="text">

14) Two Way Binding:

- [(ngModel)]

15) Structural Directives: {ngIf, ngSwitch, ngFor}

- Add or Remove HTML elements.

1) ngIf:

- syntax: <div *ngIf="expression">
<div *ngIf="expression"; else elseBlock>
<ng-template #elseBlock></ng-template>
<div *ngIf="expression"; then thenBlock; else elseBlock>
<ng-template #thenBlock></ng-template>
<ng-template #elseBlock></ng-template>

2) ngSwitch:

- syntax: <div [ngSwitch]="color">
<div *ngSwitchCase="'red'">
<div *ngSwitchCase="'blue'">
<div *ngSwitchDefault>

3) ngFor:

- syntax: <div *ngFor="let data of data">
{{data}}
</data>

16) Component Interaction

1. Input Decorator- @Input: sends parent to child component.
2. Output Decorator- @Output: sends child to parent component.

17) Pipe: Transforms data before displaying it to view.

18) Service:

1. Service is class with a specific purpose.
   - to share a data across multiple components.
   - implement application logic
   - External interactions such as connecting databases.
2. can use services using Dependency Injection.

19) Dependency Injection:

- DI as Design Pattern:
- DI is coding pattern in which a class receives its dependencies from external sources rather than creating them itself.

20) Service:

21) HTTP and Observables:

- Observable: A sequence of items that arrive asynchronously over time.
- RxJS: external library to work with observables.