# Angular Training

(Intermediate to Advanced)



#### Narasimha

Sr. Corporate Trainer, Mentor tnrao.trainer@gmail.com

### Schedule for Angular Training

Day#	Date	Topic		
Day-1	16-Nov-2023	Custom Pipes in Angular		
Day-2	17-Nov-2023	Parent-Child Communication		
Day-3	20-Nov-2023	Custom Directives in Angular		
Day-4	21-Nov-2023	Working with Reactive Forms		
Day-5	22-Nov-2023	Dependency Injection and Services in Angular		
Day-6	23-Nov-2023	Http Client – Server calls in Angular		
Day-7	24-Nov-2023	Routing and Security in Angular		
Day-8	27-Nov-2023	Unit Testing in Angular		

Duration: 8 days (2hours per day); 2pm to 4 pm; 16<sup>th</sup> Nov – 27<sup>th</sup> Nov

# Day2 Parent-Child Component Communication



#### Narasimha

Sr. Corporate Trainer, Mentor tnrao.trainer@gmail.com

### Index – Day2

- Component communication
- Data Sharing between components
- Parent- Child using @Input()
- Child Parent using @Output

### Sharing data between child and parent

- A common pattern in Angular is sharing data between a parent component and one or more child components.
- Implement this pattern with the @Input() and @Output() decorators.

### @Input and @Output

- @Input() and @Output() give a child component a way to communicate with its parent component.
- @Input() lets a parent component update data in the child component.
- Conversely, @Output() lets the child send data to a parent component.



# @Input()

**Sr. IT Trainer/Consultant** 

# @Input

Parent

data flow

Child

### @Input()

 To use the @Input() decorator in a child component class, first import Input decorator and then decorate the property with @Input()

#### greeting.component.ts

```
import { Component, Input } from '@angular/core';
export class GreetingComponent {
     @Input() uname = "; // decorate the property with @Input()
}
```

### @Input() - Configure Child

#### greeting.component.ts

```
import { Component, Input } from '@angular/core';
export class GreetingComponent {
     @Input() uname = "; // decorate the property with @Input()
}
```

#### greeting.component.html

<h3>Welcome to {{uname}}</h3>

### **Configure Parent component**

#### app.component.ts

```
import { Component, Input } from '@angular/core';
export class AppComponent {
        user1:string = "John";
}
```

#### app.component.html

```
<app-greeting uname="Narasimha"></app-greeting> <app-greeting [uname]="user1"></app-greeting>
```

#### **Parent & Child**

#### greeting.component.ts

```
import { Component, Input } from '@angular/core';
export class GreetingComponent {
         @Input() uname = ";
}
```

#### greeting.component.html

<h3>Welcome to {{uname}}</h3>

#### app.component.ts

```
import { Component, Input } from '@angular/core';
export class AppComponent {
        user1:string = "John";
}
```

#### app.component.html

```
<app-greeting uname="Scott"></app-greeting> <app-greeting [uname]="user1"></app-greeting>
```

# Practical Hands-Ons



# Output

Narasimha

**Sr. IT Trainer/Consultant** 

# @Output() decorator

- The @Output() decorator in a child component lets data flow from the child to the parent.
- @Output() marks a property in a child component as a doorway through which data can travel from the child to the parent.

# @Output

Parent

data flow

Child

# @Output() decorator

- The child component uses the @Output() property to raise an event to notify the parent of the change.
- To raise an event, an @Output() must have the type of EventEmitter,
- EventEmitter is a class in @angular/core that you use to emit custom events.

# @Output() decorator --- configure child

- 1. import { Output, EventEmitter } from '@angular/core';
- @Output() onDeptRemove:EventEmitter<number> = new EventEmitter();
- this.onDeptRemove.emit(dno);

## @Output() decorator --- configure parent

<app-dept-details
 (onDeptRemove)="deptRemove\_click\_parent(\$event)">
 deptRemove\_click\_parent(dno:number) { ... }

- The event binding, (onDeptRemove)="deptRemove\_click\_parent(\$event)",
  - connects the event in the child onDeptRemove,
  - to the method in the parent deptRemove\_click\_parent().
- The \$event contains the data that the user selected department in child template UI.

# Practical Hands-Ons

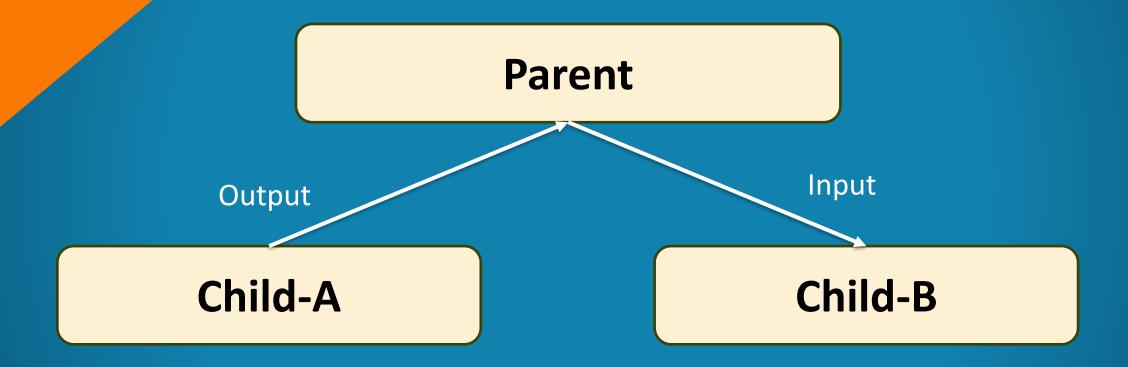


# Siblings Components Communication in Angular

Narasimha

Sr. IT Trainer/Consultant

# **Siblings Communication**



## **Siblings Communication**

**AppComponent** 

Output

Input, ngOnChanges

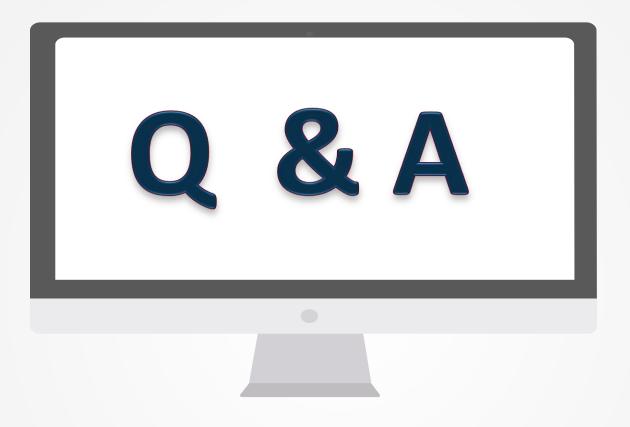
FilterOptions Component

**Emps Component** 



Select Job : MANAGER V

Emp Number	Emp Name	Emp Job	Emp Salary	Emp Deptno
6589	SRIRAM	MANAGER	3200	30
7566	JONES	MANAGER	2975	20
7698	BLAKE	MANAGER	2850	30
7782	CLARK	MANAGER	2450	10
8545	KIRAN	MANAGER	5890	10



#### Narasimha

Sr. Corporate Trainer, Mentor tnrao.trainer@gmail.com