Syntax of html:

<input [(name)]="variableNmae" (change)="fun($event.target.value)" />

* select tag
* eventEmitter in angular
* **@Input**
* **@Ouput**
* Subjects
* Observables.
* Behavioral
* decorators

**HookUps**

* **ngOnChange**()
* **onInit**()
* **ngDoCheck**()
* **ngAfterContentInit**(): Respond after Angular projects external content into the component's view. Called once after the first ngDoCheck().
* **ngAfterContentCheck**() : Respond after Angular checks the content projected into the component. Called after the ngAfterContentInit() and every subsequent ngDoCheck().
* **ngAfterViewInit**(): Respond after Angular initializes the component's views and child views. Called once after the first ngAfterContentChecked().
* **ngAfterViewCheck**(): Respond after Angular checks the component's views and child views. Called after the ngAfterViewInit and every subsequent ngAfterContentChecked().

**Subjects**

* **Behavioral Subject: it holds the state,means if you push the message using next but no one is subscribed to it,but whem someone subscribe the that will get the value;**

subject**:**BehavioralSubject<number>=new BehavioralSubject(0);

getMessage():Observable<number>

{

return this.subject.asObservable();

}

sendMessage(message:number){ this.subject.next(message);}

private messageSubcription:Subscription|undefined;

checkSubscription(){

this. messageSubcription=this.service.getMessage().subscribe((res:any)=>{

console.log(res));

}

ngOnDestroy()

{

This.messageSubcription.unsubscribe();

}

* **Subject: it doesn’t holds the state, means if you push the message using next but no one is subscribed to it then value will be missed.**

Decorators in Angular

**There are four main types of angular decorators:**

* Class decorators, such as **@Component** and **@NgModule**
* Property decorators for properties inside classes, such as **@Input** and **@Output**
* Method decorators for methods inside classes, such as **@HostListener**
  + @HostListener('click') myClick(){ } is exactly the same as (click)="myClick()"???
* Parameter decorators for parameters inside class constructors, such as **@Inject, @Injectable**