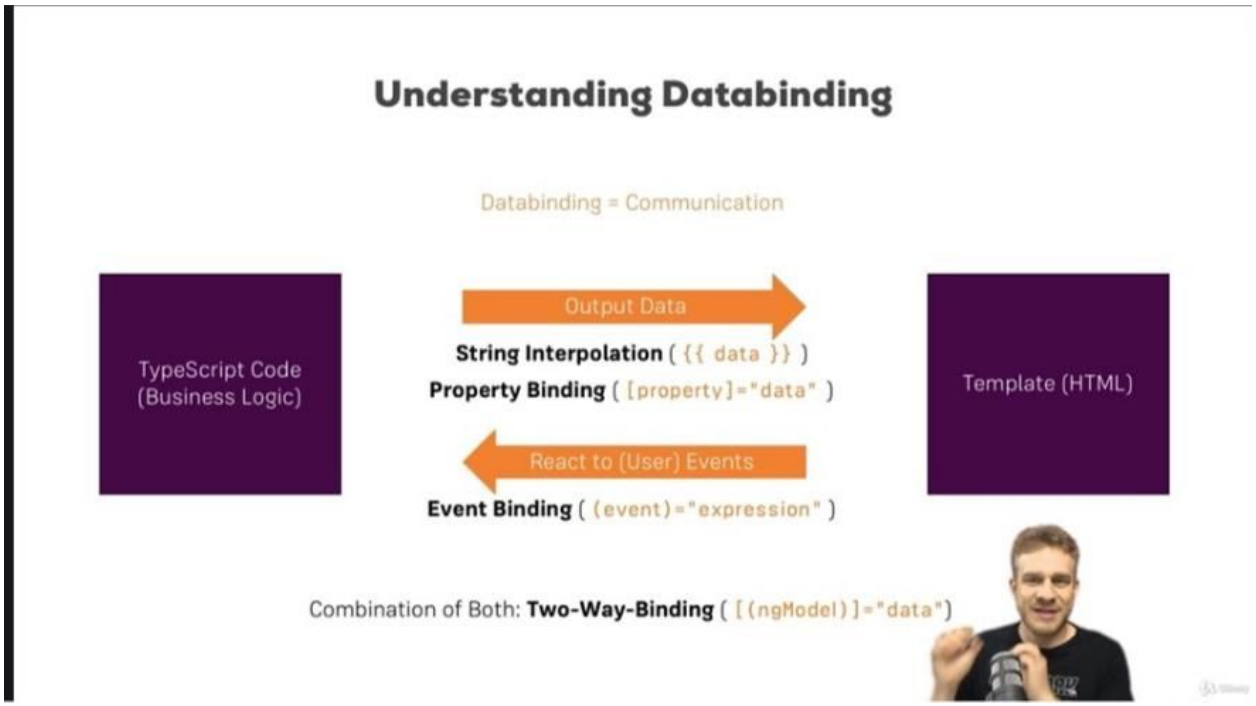
DATABINDING



Data binding is nothing but the 2 way communication between the typescript file and the template,

- **whenever you have to access some of the fields in the typescript to display in the html page**
- **Or whenever a user clicks a button in the rendering page and an action needs to be triggered in the typescript**

Databinding - To parse values from ts to html (Input)

```
//server.component.ts =>

import { Component , OnInit ,<b> Input </b>};
// @Input is a decorator , without it element can't be imported
@Input() element : {type:string,name:string,content: string};
// if we use @Input('srvElem') element
// using element is no longer possible , can be imported only using srvElem
=> [srvElem]
// app.component.ts  =>
 serverElements = [{type:'server', name:'Server_tomcat' , content: 'For
developers only'}];
 //server.component.html =>
 <app-server * ngFor ="let serverElement of serverElements "
[element]="serverElements"></app-server>
 <app-server * ngFor ="let serverElement of serverElements "
[srvElem]="serverElement"></app-server>
```

2 way Databinding - To emit event / cache the data passed (Input)

```
// app.component.html =>
<app-cockpit (serverCreated)="onServerAdded($event)"></app-cockpit>
// app.component.ts  =>
onServerAdded(serverData: {serverName:string, serverContent:string}){
this.serverElements.push({
type :'server',
name : serverData.serverName,
content : serverData.serverContent
})

//cockpit.component.ts
newserverName='';
newserverContent='';
@Output() serverCreated = new EventEmitter<serverName: string , serverContent:
string}>();

onAddServer(){
this.serverCreated.emit({
serverName : this.newserverName,
serverContent : this.newserverContent
});
}
```