**Components and DataBinding :**

**How to pass data from parent component to a child component?**

* So here I need the property in the child component to be accessible from the parent component.
* We have property element defined in the **server.component.ts** and want to access it from the parent component which is **app.component.ts**.

Une image contenant texte

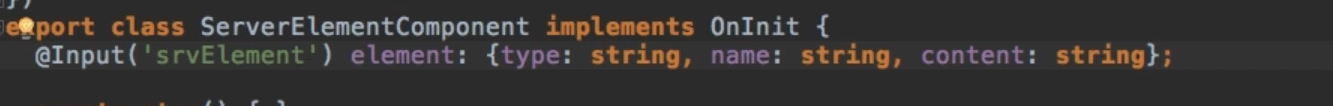
Description générée automatiquement

* You should add decorator **@Input()** in front of the public property element.

Une image contenant texte

Description générée automatiquement

* We can also assign an alias to a custom property, and tell Angular, i want you to call it with this alias within the parent component.



* In general, @Input() let the parent component update data in the child component.
* Conversely, @Output() lets the child send data to a parent component.

**How to send data from the child to parent component :**

1. Configuring the child component :

Une image contenant texte

Description générée automatiquement

-----------------------------------

Une image contenant texte

Description générée automatiquement

* The ***addNewItem()*** function uses the ***@Output()***, ***newItemEvent***, to raise an event with the value the user types into the ***<input>***.

Une image contenant texte

Description générée automatiquement

1. Configuring the parent component :

Une image contenant texte

Description générée automatiquement

* The addItem takes an argument as a string and then add this argument to the items array.



* The $event contains the data the user types into the <input> in the child template UI.
* Which means, when something happens in the **app-item-output** that triggers the **newItemEvent**, we will call the **addItem** function that exists in the **parent component**.

**View Encapsulation :**

* Means will we expose styling of our view or no.
* ***Emulated*** by default, means we get our styling for component only from the associated component.
* ***None*** means, styling will be set globally.

Une image contenant texte

Description générée automatiquement

**Getting access to the template & DOM with @ViewChild :**

* This method can be used, if you don’t want to use the two-way-data binding ;
* First, define the local reference. Then, define the attribute with the corresponding decorator.
* Finally, call the attribute and access its value via : **nativeElement.value**

Une image contenant texte

Description générée automatiquement



Une image contenant texte

Description générée automatiquement

* Strong recommendation : not access the DOM with ElementRef and update its value
  + *This.serverContentInput.nativeElement.value = ‘something’* **BAD.**

**Component lifecycle :**

Une image contenant texte

Description générée automatiquement