

Input

[data] = "parentData"



<BOX> // @Input() data;



box.data = parentData;

Output

(notifyParent) = "handleEventFromBox(\$event)"
= "someOtherParentData = \$event"

↑
someValue === \$event



// \$event is made available to the above expression.
The expression is evaluated on every event emission.

<BOX> // @Output() notifyParent; = new EventEmitter();



box.notifyParent.emit(someValue)

Output Types:

DOM Events

- onclick = (click)

- onsubmit = (submit)

- onmouseenter = (mouseenter)

- \$event = Native DOM event object

Custom Events - @Output() myCustom = (myCustom)

@Output() foo = (foo)

- \$event = Value emitted most recently from EventEmitter

Two-way Binding

- Combination of @Input and @Output
- Relationship between input and output member names

Box

```
@Input() data;
```

```
@Output() dataChange = new EventEmitter();
```

// output property use input name with "Change" suffix

```
<BOX [data]="parentData"  
      (dataChange)="parentData = $event">
```

↑
↓ Equivalent to

```
<BOX [(data)]="parentData">
```

// The same way NgModel works

```
<input [ngModel]="someValue"  
      (ngModelChange)="someValue = $event">
```

↑
↓

```
<input [(ngModel)]="someValue">
```