@Input

@Output

1. 父元件和1~多個子元件間的共享資料,可用

* Input:把某Class欄位標示為輸入屬性,並提供設定metadata(元資料)，屬性繫結到樣板的某個DOM屬性,當變更DOM屬性值,自動將此值用來更新輸入屬性 (DOM變動->變動)
* Output: 把某Class欄位標示為輸出屬性,並提供設定metadata，輸出屬性的值發生改變時,凡繫結到輸出屬性的DOM屬性會被自動更新

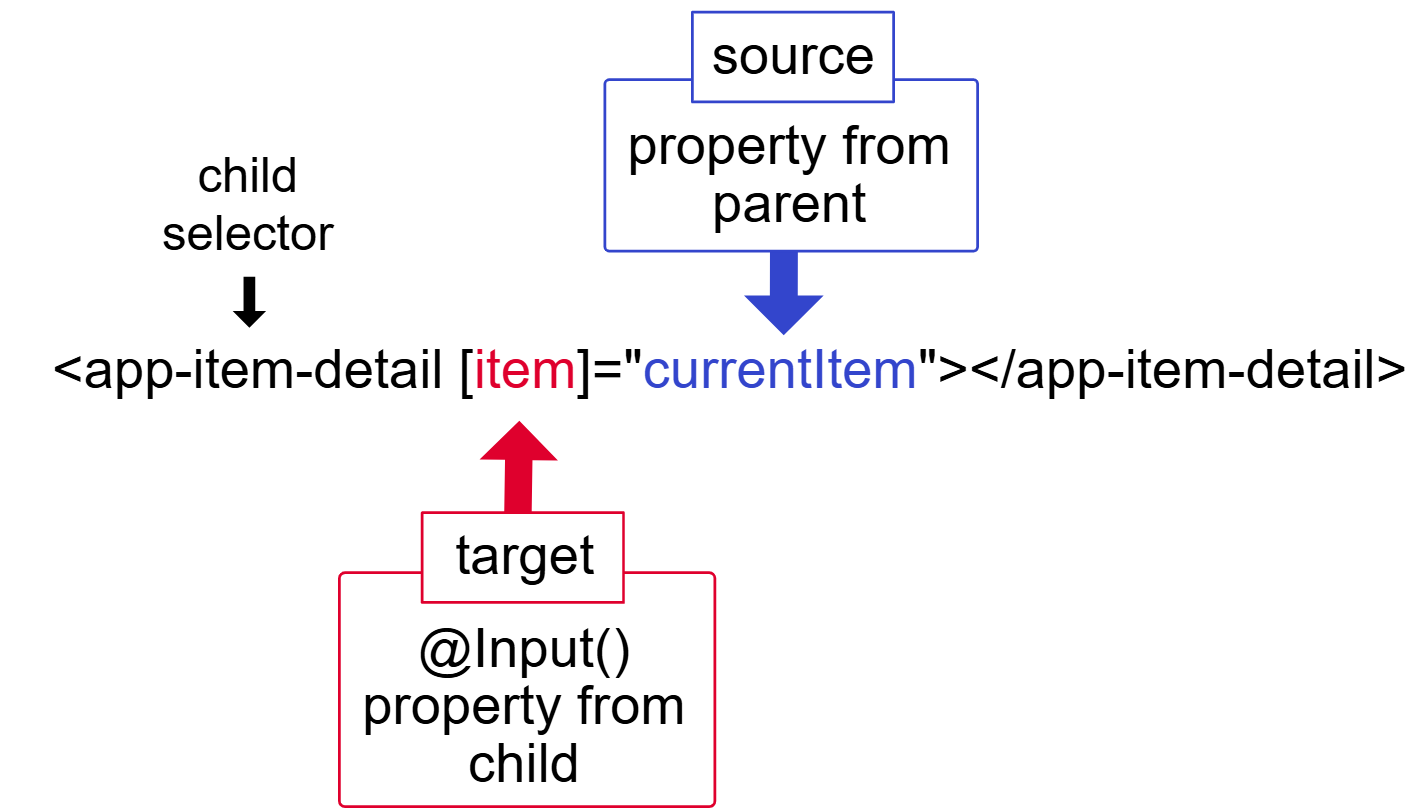
(Class屬性變動->DOM變動)

1. 資料傳送方式設定

* input: 父元件傳給子元件:

1. 先設定小孩,在class中宣告@Input()的屬性
2. 在爸爸的html設定小孩的selector 並在標籤中設定[小孩的屬性]="爸爸的屬性"
3. 在爸爸的class設定屬性和值

此會單向給資料到小孩設定的@Input()



* output: 子元件傳給父元件: (output的資料需要的type: [EventEmitter](https://angular.tw/api/core/EventEmitter))

範例以子元件中輸入資料,把資料傳遞給父元件

1. 先import { Output, EventEmitter } from '@angular/core';
2. 子元件class中增加物件

@Output() forParentData = new EventEmitter<string>();

1. 子元件class增加一個方法,把資料從input標籤放入傳遞的屬性中

addMsgToParent(msg: string){

this.forParentData.emit(msg);

}

1. 子元件的html增加input

<label>

<span>Add msg to parent: </span><br/>

<input type="text" #newMsg>

<button type="button" (click)="addMsgToParent(newMsg.value)">add msg</button>

</label>

1. 父元件中增加一個屬性和一個放資料到此屬性的方法
2. 父元件的html中設定子元件的output為事件,並設定父元件的方法

<child-selector (forParentData)="addMsg($event)"></child-selector>