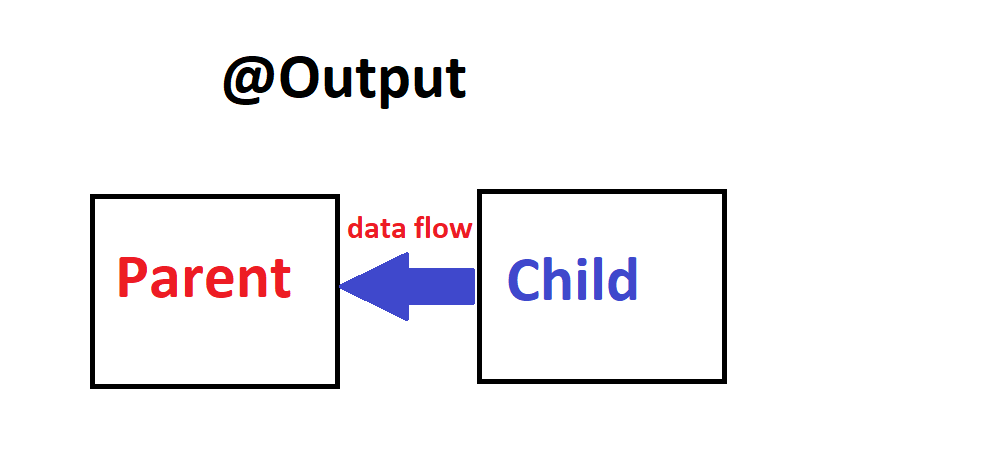
**@Output( ) Decorator**

***Use the @***[***Output***](https://angular.io/api/core/Output)***() decorator in the child component or directive to allow data to flow from the child*out*to the parent.***

***An @***[***Output***](https://angular.io/api/core/Output)***() property should normally be initialized to an Angular [EventEmitter](https://angular.io/api/core/EventEmitter) with values flowing out of the component as***[***events***](https://angular.io/guide/event-binding)***.***

******

***Just like with @***[***Input***](https://angular.io/api/core/Input)***(), you can use @***[***Output***](https://angular.io/api/core/Output)***() on a property of the child component but its type should be [EventEmitter](https://angular.io/api/core/EventEmitter).***

***@***[***Output***](https://angular.io/api/core/Output)***() marks a property in a child component as a doorway through which data can travel from the child to the parent. The child component then has to raise an event so the parent knows something has changed. To raise an event, @***[***Output***](https://angular.io/api/core/Output)***() works hand in hand with [EventEmitter](https://angular.io/api/core/EventEmitter), which is a class in @angular/core that you use to emit custom events.***

***When you use @***[***Output***](https://angular.io/api/core/Output)***(), edit these parts of your app:***

* ***The child component class and template***
* ***The parent component class and template***

***Decorator that marks a class field as an output property and supplies configuration metadata. The DOM property bound to the output property is automatically updated during change detection***

***In the child***

***First, be sure to import***[***Output***](https://angular.io/api/core/Output)***and [EventEmitter](https://angular.io/api/core/EventEmitter) in the child component class:***

***import {*** [***Output***](https://angular.io/api/core/Output)***,*** [***EventEmitter***](https://angular.io/api/core/EventEmitter) ***} from '@angular/core';***

***Next, still in the child, decorate a property with @***[***Output***](https://angular.io/api/core/Output)***() in the component class. The following example @***[***Output***](https://angular.io/api/core/Output)***() is called newItemEvent and its type is [EventEmitter](https://angular.io/api/core/EventEmitter), which means it's an event.***

***Item-output.component.ts***

***@Output() newItemEvent = new EventEmitter<string>();***

***The different parts of the above declaration are as follows:***

* ***@***[***Output***](https://angular.io/api/core/Output)***()—a decorator function marking the property as a way for data to go from the child to the parent***
* ***newItemEvent—the name of the @***[***Output***](https://angular.io/api/core/Output)***()***
* [***EventEmitter***](https://angular.io/api/core/EventEmitter)***<string>—the @***[***Output***](https://angular.io/api/core/Output)***()'s type***
* ***new [EventEmitter](https://angular.io/api/core/EventEmitter)<string>()—tells Angular to create a new event emitter and that the data it emits is of type string. The type could be any type, such as***[***number***](https://angular.io/api/common/DecimalPipe)***, boolean, and so on.***