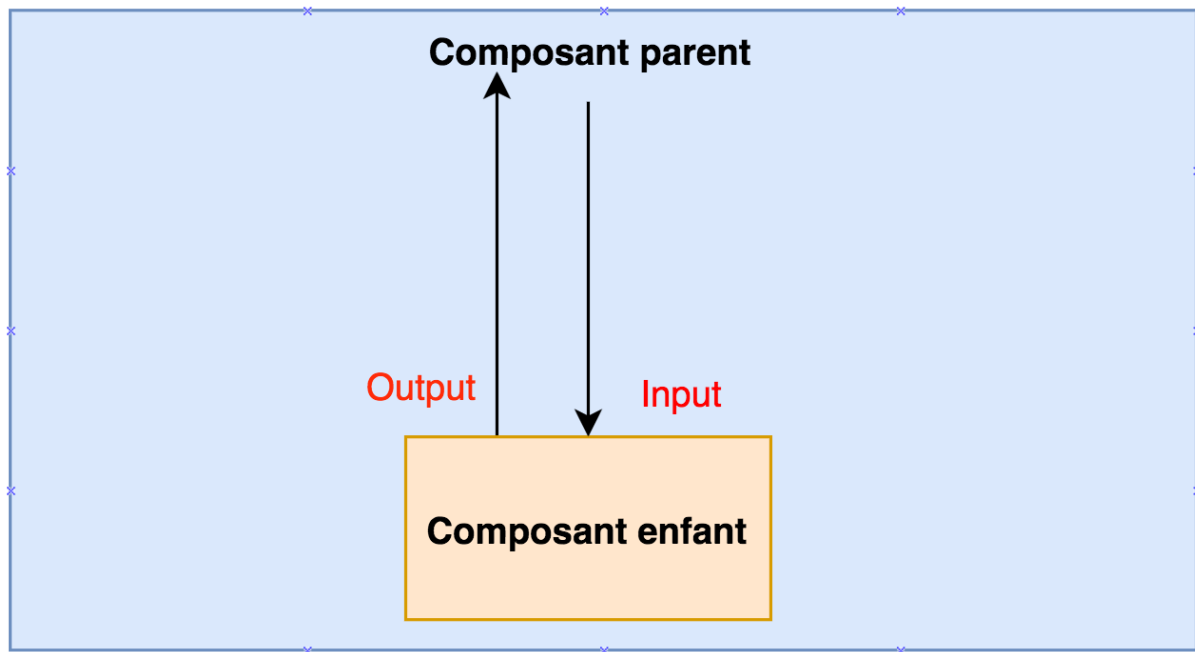


Propriété output

Communiquer des données d'un composant enfant vers un composant parent

Pour communiquer des données d'un composant enfant vers un composant parent, il faut utiliser une propriété

Les termes `output` et `input` (c'est-à-dire *entrée et sortie*) sont caractérisés du point de vue du composant enfant :



Déclaration d'une propriété

Une propriété `output` permet de transmettre des données du composant enfant vers le composant parent.

Comme pour une propriété `input`, il y a deux façons de la déclarer sur un composant.

Premièrement, **en utilisant le décorateur** `Output`.

Il est nécessaire d'importer `Output` et `Component` depuis `@angular/core` pour pouvoir utiliser ce décorateur.

Prenons un exemple très simple de compteur : le composant enfant contient des boutons permettant de décrémenter et d'incrémenter un compteur et transmet cette valeur au composant parent qui l'affiche.

<https://stackblitz.com/edit/angular-c4-l3-1>

Dans cet exemple, `ChildComponent` est une instance de la classe `Component` en charge d'envoyer un événement lorsque l'on presse sur l'un des boutons.

Il existe une méthode sur les instances d'`Component`, `emit`, qui permet d'envoyer un événement contenant une valeur, qui peut être par exemple un objet.

Sur le composant parent nous déclarons qu'à chaque fois qu'un événement `childEvent` est reçu, la méthode `onChildEvent` doit être exécutée, auquel on passe `event` qui contient un objet

Pour rappel: `childEvent` est bien la **cible d'une liaison d'événement**. On ne peut accéder à des cibles de liaison de propriété `input`, ou d'événement `output` qu'en déclarant explicitement sur le composant enfant qu'il s'agit de

propriétés ou .

Deuxièmement, comme pour , il est possible de déclarer sur les métadonnées du composant enfant (c'est-à-dire sur l'objet passé au décorateur) les propriétés avec .

2) Utiliser des alias pour les propriétés et

Comme nous l'avons vu dans les vidéos il est très simple d'utiliser un alias avec les propriétés et .

Utiliser un alias permet d'avoir un nom différent entre le nom interne de la propriété, c'est-à-dire dans la classe du composant enfant, et le nom public, c'est-à-dire le nom utilisé pour déclarer la liaison de la propriété (c'est à dire sur).

Si vous utilisez les décorateurs il suffit de passer en paramètre l'alias au décorateur :

Si vous utilisez les propriétés sur le décorateur

il suffit d'indiquer l'alias :

En reprenant notre exemple précédent, attribuons un alias pour le nom public de la propriété et conservons pour le nom interne dans le composant enfant.

<https://stackblitz.com/edit/angular-c4-l3-2>

Continuer l'exemple

Reprenons le code d'exemple avec nos fruits et ajoutons la possibilité de supprimer un fruit.

Modification du template du composant

Dans le template `App.js`, nous avons simplement à ajouter une liaison d'événement sur le clic qui va déclencher une méthode :

Modification de la classe du composant

Dans la classe, nous allons déclencher l'émission d'un événement grâce à une propriété `onPress` et à un `onPress={() => {}}` :

A chaque clic sur un des composants fruit, nous déclenchons l'émission d'un événement avec la méthode `onPress`.

Cette méthode déclenche l'émission d'une valeur, ici du fruit, par l' que nous déclarons.

Modification du template du composant racine

Dans le composant racine, nous devons ajoutons une liaison d'événement, sur l'événement personnalisé que nous émettons depuis le composant enfant :

Dans le composant racine, qui est ici le composant parent, nous déclarons une méthode qui va recevoir la valeur émise par l' du composant enfant.

Modification de la classe du composant racine

Dans la classe, nous déclarons la méthode qui va traiter l'événement contenant le fruit :

Lorsque nous recevons le fruit, nous l'enlevons de la liste de fruits.

Code exécutable

Voici où nous en sommes :

<https://stackblitz.com/edit/angular-chap4-l3>