

first.component.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-first',
  templateUrl: './first.component.html',
  styleUrls: ['./first.component.css']
})
export class FirstComponent {

  // Message to send to the SecondComponent
  message: string = 'Hello from First Component!';

  // Message received from SecondComponent
  receivedMessage: string = '';

  // Method to receive the message from SecondComponent
  receiveMessage(message: string) {
    this.receivedMessage = message;
  }

  //Method to send message to second Component
  sendMessageToSecond() {
    this.message = 'Message updated by First Component';
  }
}
```

- **message:** sent to child using [parentMessage]="message"
- **sendMessageToSecond():** updates message for child
- **receiveMessage(\$event):** receives message from child

first.component.html

```
<!-- first.component.html -->
<div class="container">
  <h2>First Component</h2>

  <!-- Button to send message to SecondComponent -->
  <button (click)="sendMessageToSecond()">Send Message to
  Second Component</button>

  <!-- Passing message to SecondComponent via @Input and
  receiving with @Output -->
  <app-second
  [parentMessage]="message"
  (messageEvent)="receiveMessage($event)">
</app-second>

  <p>Message Received from Second Component: {{
receivedMessage }}</p>
</div>
```

using the **<app-second>** component **inside** the **FirstComponent**.

This is called a **child component**.

This line is doing **two-way communication**:

- **From Parent to Child** using `@Input()`
- **From Child to Parent** using `@Output()`

— @Input Binding

```
[parentMessage]="message"
```

This means:

"Send the value of the variable `message` from the parent (FirstComponent) to the child (SecondComponent) and store it in the **child's @Input() parentMessage.**"

```
// Message to send to the SecondComponent
// In FirstComponent.ts
message: string = 'Hello from First Component!';

// In SecondComponent.ts
@Input() parentMessage: string = '';
```

The child will receive:

```
// Message to send to the SecondComponent
message: string = 'Hello from First Component!';
```

(messageEvent)="receiveMessage(\$event)" — @Output Event Binding

"Listen for the custom event **messageEvent** coming from the child component and call the **receiveMessage()** method in the parent with the emitted value (**\$event**)."

// In SecondComponent.ts

```
// Sending message to FirstComponent via @Output
//Data goes from Child to Parent using @Output() +
EventEmitter
@Output() messageEvent = new EventEmitter<string>();
```

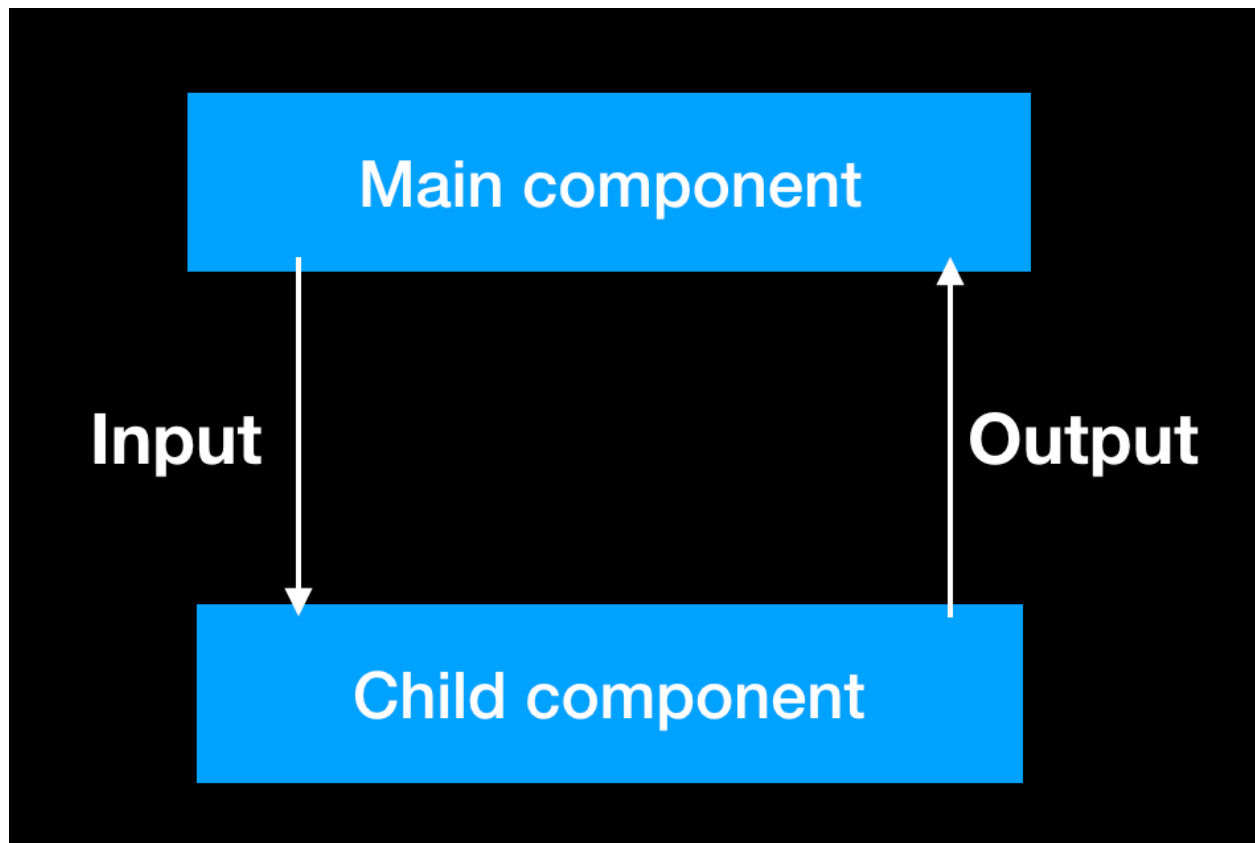
```
// Method to send message back to FirstComponent

this.messageEvent.emit(Message from Second Component.');
```

then in First component

```
// Method to receive the message from SecondComponent
receiveMessage(message: string) {
  this.receivedMessage = message;
}
```

So now the parent can receive data from the child.



second.component.ts

```
import { Component, EventEmitter, Input, Output } from
'@angular/core';

@Component({
  selector: 'app-second',
  templateUrl: './second.component.html',
  styleUrls: ['./second.component.css']
})
export class SecondComponent {

  message: string = 'Hello from Second Component!';

  // Receiving message from FirstComponent via @Input
  //Data goes from Parent to Child using @Input()
  @Input() parentMessage: string = '';
```

```

// Sending message to FirstComponent via @Output
//Data goes from Child to Parent using @Output() +
EventEmitter
@Output() messageEvent = new EventEmitter<string>();

// Method to send message back to FirstComponent
sendMessageToFirst() {
    this.messageEvent.emit('Hello First Component! Message
from Second Component. ');
}
}

```

- **@Input()** receives data from parent
- **@Output()** sends data back to parent
- **sendMessageToFirst()** emits the message to parent

second.component.html

```

<!-- second.component.html -->
<div>
    <h3>Second Component</h3>

    <!-- Display message received from FirstComponent -->
    <p>Message from First Component: {{ parentMessage }}</p>

    <!-- Button to send message back to FirstComponent -->
    <button (click)="sendMessageToFirst()">Send Message to
First Component</button>
</div>

```

