# **Component Interaction**

* Interaction of component is nothing but sharing data between component or else communicating data between components.
* Communicating Data between components is done in 3 ways.

1. Parent to Child using **@Input()**

**parent.component.ts**

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

@Component({

  selector: 'app-parent',

  templateUrl: './parent.component.html',

  styleUrls: ['./parent.component.css']

})

export class ParentComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

  message = "Hello Child";

}

**child.component.ts**

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

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css']

})

export class ChildComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

  @Input() parentMessage:any;

}

**child.component.html**

<p>{{parentMessage}}</p>

**Parent.component.html**

<app-child [parentMessage]="message"></app-child>

1. Child to Parent using **@Output()**

**child.component.html**

<button (click)="sendMessage()">Send Message To Parent</button>

**child.component.ts**

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

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css']

})

export class ChildComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

  @Output() messageEvent = new EventEmitter();

  sendMessage(){

    this.messageEvent.emit("Hello Parent");

  }

}

**Parent.component.html**

<app-child (messageEvent)="receiveChildMessage($event)">

</app-child>

<p>{{childMessage}}</p>

**parent.component.ts**

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

@Component({

  selector: 'app-parent',

  templateUrl: './parent.component.html',

  styleUrls: ['./parent.component.css']

})

export class ParentComponent implements OnInit {

  childMessage: any;

  constructor() { }

  ngOnInit(): void {

  }

  receiveChildMessage(event:any){

    this.childMessage = event;

  }

}

1. Child to Parent using **@ViewChild()**

**child.component.ts**

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

@Component({

  selector: 'app-child',

  templateUrl: './child.component.html',

  styleUrls: ['./child.component.css']

})

export class ChildComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

  message = "Hi Parent";

}

**Parent.component.html**

<button (click)="viewChildMessage()">View Child Message</button>

<p>{{childsmessage}}</p>

**parent.component.ts**

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

@Component({

  selector: 'app-parent',

  templateUrl: './parent.component.html',

  styleUrls: ['./parent.component.css']

})

export class ParentComponent implements OnInit {

  childsmessage: string | undefined;

  constructor() { }

  ngOnInit(): void {

  }

  @ViewChild(ChildComponent) child!:ChildComponent;

  viewChildMessage(){

    this.childsmessage = this.child.message;

  }

}