**Component Interaction**

**Parent to Child: Sharing Data via Input**

This is probably the most common and straightforward method of sharing data.

It works by using the [**@Input() decorator**](https://angular.io/docs/ts/latest/api/core/index/Input-interface.html) to allow data to be passed via the template.

parent.component.html

<p>parent</p>

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

parent.component.ts

import {Component} from '@angular/core'

@Component({

  selector:'app-parent',

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

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

})

export class appParent{

  parentMessage="message from parent";

}

child.component.html

<p>child ===> {{childMessage}}</p>

child.component.ts

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

@Component({

  selector: "app-child",

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

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

})

export class appChild {

  @Input() childMessage: string;

}

app.component.html

<app-parent></app-parent>

app.component.ts

import { NgModule } from "@angular/core";

import { BrowserModule } from "@angular/platform-browser";

import { FormsModule } from "@angular/forms";

import { AppComponent } from "./app.component";

import { HelloComponent } from "./hello.component";

import { appParent } from "../parent/parent.component";

import { appChild } from "../child/child.component";

@NgModule({

  imports: [BrowserModule, FormsModule],

  declarations: [AppComponent, HelloComponent, appParent, appChild],

  bootstrap: [AppComponent]

})

export class AppModule {}