5)Two-way Binding

1)app.module.ts

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

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

import { AppRoutingModule } from './app-routing.module';

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

import { ChildComponent } from './child/child.component';

@NgModule({

  declarations: [

    AppComponent,

    ChildComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

2)app.component.ts

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

@Component({

  selector: 'app-root',

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

  styleUrls: ['./app.component.css']

})

export class AppComponent {

  public cdata: string|undefined;

}

3)app.component.html

<h2>Parent Component</h2>

This is Parent Component<br>

Enter Text:

<input type="text" #ptext (keyup)="0"/><br>

The value of Child component is: {{cdata}}

<app-child (cevent)="cdata=$event" [pdata]="ptext.value"></app-child>

4)child.component.html

<p>child works!</p>

<h2>Child Component</h2>

This is Child Component<br>

Enter Text:

<input type="text" #cdata (keyup)="onChange(cdata.value)"/><br>

The value od Parent component is: {{pdata}}

5)child.component.ts

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

@Component({

  selector: 'app-child',

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

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

  inputs: [`pdata`],

  outputs: [`cevent`]

})

export class ChildComponent implements OnInit {

  constructor() { }

  ngOnInit() {

  }

  public pdata: string|undefined;

  cevent= new EventEmitter<string>();

  onChange(value:string){

    this.cevent.emit(value);

  }

}

Output:

