**ANGULAR 2**

We need two component to check this operation

1. Consider App Component as your parent component

2. Create New Component ( Here Tutorial) as child component

app.component.ts:

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

import {TutorialsComponent} from './tutorials/tutorials.component';

@Component({

selector: 'app-root',

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

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

//directives:[`TutorialsComponent`]

})

export class AppComponent {

title = 'app';

public childData:string;

}

app.component.html:

<h3> Enter Parent AppCompinent Value</h3>

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

<h3>{{childData}}</h3>

<app-tutorials (childEvent)="childData=$event"[parentData]="ptext.value"></app-tutorials>

tutorials.component.ts

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

//import { EventEmitter } from 'events';

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

@Component({

selector: 'app-tutorials',

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

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

inputs:[`parentData`],

outputs:[`childEvent`]

})

export class TutorialsComponent implements OnInit {

public parentData:string;

childEvent = new EventEmitter<string>();

onChange(value:string)

{

this.childEvent.emit(value);

}

constructor() { }

ngOnInit() {

}

}

tutorials.component.html:

<h2>

Tutorials works!

</h2>

<h3>Enter Child Component Values</h3>

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

<br>

<h3>{{parentData}}</h3>

app.module.ts:

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

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

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

import { TutorialsComponent } from './tutorials/tutorials.component';

@NgModule({

declarations: [

AppComponent,

TutorialsComponent

],

imports: [

BrowserModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }