

## 5. Two-way Binding.

### 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>
```

### App.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  constructor() {
    this.cdata = '';
  }
  public cdata:string;
}
```

### Child.component.html

```
<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}}
```

### Child.component.ts

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

@Component({
  selector: 'app-child',
  templateUrl: './child.component.html',
  styleUrls: ['./child.component.css'],
})
export class ChildComponent implements OnInit {
  @Input() pdata: string;
  cdata: string;
  constructor() {}
  ngOnInit(): void {}
  onChange(cdata: string): void {}
}
```

```

    inputs: [`pdata`],
    outputs: [`cevent`]
  })
  export class ChildComponent implements OnInit {

    constructor() {
      this.pdata = '';
    }

    ngOnInit() {
    }

    public pdata: String;
    cevent= new EventEmitter<string>();

    onChange(value:string){
      this.cevent.emit(value);
    }
  }
}

```

### App.module.ts

```

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

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 { }

```

