
父子组件传值@Input @Output @ViewChild

合作网站: www.itying.com (IT 营)
合作网站: www.ionic.wang (Ionic 中文网)

目录

一、父组件给子组件传值 -@Input.....	1
二、父子组件传值的方式让子组件执行父组件的方法.....	2
三、子组件通过@Output 执行父组件的方法.....	4
四、父组件通过局部变量获取子组件的引用，主动获取子组件的数据和方法（一）	5
五、父组件通过 ViewChild 主动获取子组件的数据和方法.....	6

一、父组件给子组件传值 -@Input

1. 父组件调用子组件的时候传入数据

```
<app-header [msg]="msg"></app-header>
```

2. 子组件引入 Input 模块

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

3. [子组件中 @Input](#) 接收父组件传过来的数据

```
export class HeaderComponent implements OnInit {  
  
  @Input() msg:string  
  
  constructor() { }  
  
  ngOnInit() {  
  
  }  
  
}
```

4. 子组件中使用父组件的数据

```
<h2>这是头部组件--{{msg}}</h2>
```

二、父子组件传值的方式让子组件执行父组件的方法

1. 父组件定义方法

```
run(){  
  alert('这是父组件的 run 方法');  
}
```

2. 调用子组件传入当前方法

```
<app-header [msg]="msg" [run]="run"></app-header>
```

3. 子组件接收父组件传过来的方法

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

```
@Input() run:any;
```

```
export class HeaderComponent implements OnInit {  
  
  @Input() msg:string  
  @Input() run:any;  
  
  constructor() { }  
}
```

4. 子组件使用父组件的方法。

```
export class HeaderComponent implements OnInit {  
  
  @Input() msg:string;  
  
  @Input() run:any;  
  
  constructor() { }  
  
  ngOnInit() {  
  
    this.run(); /*子组件调用父组件的 run 方法*/  
  
  }  
}
```

三、子组件通过@Output 执行父组件的方法

1. 子组件引入 Output 和 EventEmitter

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

2. 子组件中实例化 EventEmitter

```
@Output() private outer=new EventEmitter<string>();  
  
/*用 EventEmitter 和 output 装饰器配合使用 <string>指定类型变量*/
```

3. 子组件通过 EventEmitter 对象 outer 实例广播数据

```
sendParent(){  
    // alert('zhixing');  
    this.outer.emit('msg from child')  
}
```

4. 父组件调用子组件的时候，定义接收事件，outer 就是子组件的 EventEmitter 对象 outer

```
<app-header (outer)="runParent($event)"></app-header>
```

5. 父组件接收到数据会调用自己的 runParent 方法，这个时候就能拿到子组件的数据

```
//接收子组件传递过来的数据
runParent(msg:string){
    alert(msg);
}
```

四、父组件通过局部变量获取子组件的引用，主动获取子组件的数据和方法（一）

1. 定义 footer 组件

```
export class FooterComponent implements OnInit {

    public msg:string;

    constructor() {

    }

    ngOnInit() {

    }

    footerRun(){

        alert('这是 footer 子组件的 Run 方法');
    }

}
```

2. 父组件调用子组件的时候给子组件起个名字

```
<app-footer #footer></app-footer>
```

3. 直接获取执行子组件的方法

```
<button (click)='footer.footerRun() '>获取子组件的数据</button>
```

五、父组件通过局部变量获取子组件的引用,通过 ViewChild 主动获取子组件的数据和方法

1.调用子组件给子组件定义一个名称

```
<app-footer #footerChild></app-footer>
```

2. 引入 ViewChild

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

3. ViewChild 和刚才的子组件关联起来

```
@ViewChild('footerChild') footer;
```

4.调用子组件

```
run(){  
  
    this.footer.footerRun();  
}
```
