

Angular4.x 创建组件、绑定数据、绑定属性、数据循环、条件判断、事件、表单处理、双向数据绑定

主讲教师：（大地）

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

目录

一、创建 angular 组件.....	1
二、angular4.0 绑定数据.....	2
三、数据循环 *ngFor.....	3
四、条件判断 *ngIf	4
五、执行事件 (click)="getData()".....	4
六、绑定属性.....	5
七、表单处理.....	5
八、双向数据绑定	5
九、TodoList 功能.....	6

一、创建 angular 组件

<https://github.com/angular/angular-cli>

Scaffold	Usage
Component	ng g component my-new-component 指定目录创建 : ng g component components/Footer
Directive	ng g directive my-new-directive
Pipe	ng g pipe my-new-pipe
Service	ng g service my-new-service
Class	ng g class my-new-class
Guard	ng g guard my-new-guard
Interface	ng g interface my-new-interface
Enum	ng g enum my-new-enum
Module	ng g module my-module

创建组件

```
ng g component components/header
```

使用组件：

```
<app-header></app-header>
```

二、angular4.0 绑定数据

1. 数据文本绑定

```
{{}}
```

```
<h1>
  {{title}}
```

```
</h1>
```

2. 绑定 html

```
this.h = "<h2>这是一个 h2 用 [innerHTML] 来解析</h2>"
```

```
<div [innerHTML]="h"></div>
```

三、数据循环 *ngFor

1、*ngFor 普通循环

```
<ul>
  <li *ngFor="let item of list">
    {{item}}
  </li>
</ul>
```

2、循环的时候设置 key

```
<ul>
  <li *ngFor="let item of list; let i = index;">
    {{item}} --{{i}}
  </li>
</ul>
```

2. template 循环数据

```
<ul>
  <li template="ngFor let item of list">
    {{item}}
  </li>
</ul>
```

四、条件判断 *ngIf

```
<p *ngIf="list.length > 3">这是 ngIF 判断是否显示</p>

<p template="ngIf list.length > 3">这是 ngIF 判断是否显示</p>
```

五、执行事件 (click)="getData()"

```
<button class="button" (click)="getData()">
  点击按钮触发事件
</button>
```

```
<button class="button" (click)="setData()">
  点击按钮设置数据
</button>
```

```
getData(){ /*自定义方法获取数据*/
  //获取
  alert(this.msg);
}
setData(){
  //设置值
```

```
this.msg='这是设置的值';  
}
```

六、绑定属性

```
<div [id]="id" [title]="msg">调试工具看看我的属性</div>
```

```
<app-root _ngcontent-c1 _ngghost-c3>/app-root</app-root>  
<app-footer _ngcontent-c1 _ngghost-c4>...</app-footer>  
<div _ngcontent-c1 id="123" title="你好angularjs4.0">调试工具看看我的属性</div> == $0  
/app-news>  
pp-root>
```



七、表单处理

```
<input type="text" (keyup)="keyUpFn($event)" />
```

```
<input type="text" (keyup)="keyUpFn($event)" />
```

```
keyUpFn(e){  
  console.log(e)  
}
```

八、双向数据绑定

```
<input [(ngModel)]="inputValue">
```

注意引入: FormsModule

```
import { FormsModule } from '@angular/forms';
```

```
@NgModule({  
  declarations: [  
    AppComponent,  
    HeaderComponent,  
    FooterComponent,  
    NewsComponent  
  ],  
  imports: [  
    BrowserModule,  
    FormsModule  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule { }
```

使用:

```
<input type="text" [(ngModel)]="inputValue"/>  
  
{{inputValue}}
```

九、Todolist 功能

html 代码

```
<input type="text" [(ngModel)]='username'>  
  
<button (click)='addData()'>增加</button>
```

```
<ul>
  <li *ngFor="let item of list">
    {{item}}
  </li>
</ul>
```

Js 代码

```
export class TodolistComponent implements OnInit {

  list:any[];
  username:any;

  constructor() {

  }

  ngOnInit() {
    this.list=[];
    this.username='';
  }
  addData(){

    alert(this.username);

    this.list.push(this.username);
  }
}
```