## **Angular常用语法：**

#### **1、事件绑定  （）:**

<button (click) = "share()"> share </button>

#### ****2、click** 点击事件：**

<button (click) = "share()"> share </button>

#### ****3、ng-hide/ng-show**设置应用部分是否可见：**

<p ng-hide="true"> //隐藏

<p ng-hide="false">//显示

#### **4、ngModelChange选择改变事件：**

=============Html=============

<div class="set-select">

<label for="rankbutton">选择平台</label>

<select id="rankbutton" [(ngModel)]="platform" (ngModelChange)="set\_platform()">

<select id="rankbutton" [(ngModel)]="platform">

<option \*ngFor="let item of platforms" [value]='item.key'>{{item.value}}</option>

</select>

</div>

============Ts================

platform = 'wx';

platforms: any = [

{ key: 'wx', value: '微信' },

{ key: 'tt', value: '百度' },

{ key: 'wb', value: '微博' },

{ key: 'bjh', value: '抖音' },

{ key: 'zcool', value: '淘宝' },

];

set\_platform() {

this.platform

console.log('this.platform:',this.platform)

}

#### **5、input事件在用户输入时触发：**

<input placeholder="input here" (input)="onInput($event)" />

#### **6、属性绑定   [ ]  语法：**

<a [title]="product.name+'描述'">

#### **7、[(ngModel)] ：双向绑定：**

NgModel 指令允许你显示数据属性并在用户进行更改时更新该属性。这是一个例子：

src/app/app.component.html (NgModel example)

content\_copy

<input [(ngModel)]="currentItem.name" id="example-ngModel" name='currentName'> //注意某些情况下需要加name表示唯一标识，不加的话可能会报错

导入 FormsModule 以使用 ngModel  
要想在双向数据绑定中使用 ngModel 指令，必须先导入 FormsModule 并将其添加到 NgModule 的 imports 列表中。要了解关于 FormsModule 和 ngModel 的更多信息，参阅表单一章。

记住，要导入 FormsModule 才能让 [(ngModel)] 可用，如下所示：

src/app/app.module.ts (FormsModule import)

content\_copy

import { FormsModule } from '@angular/forms'; // <--- JavaScript import from Angular

/\* . . . \*/

@NgModule({

/\* . . . \*/

imports: [

BrowserModule,

FormsModule // <--- import into the NgModule

],

/\* . . . \*/

})

export class AppModule { }

[https://angular.cn/guide/built-in-directives#ngModel](https://angular.cn/guide/built-in-directives" \l "ngModel" \t "http://t.zoukankan.com/_blank)

#### **8、插值语法 {{...}}：**

花括号之间的文本通常是组件属性的名字。Angular 会把这个名字替换为响应组件属性的字符串值。

<p>{{title}}</p>

<div><img src="{{itemImageUrl}}"></div>

#### **9、Angular使用[InnerHtml]中正常显示富文本内容：**

<div class="text" [innerHTML]="richText"></div>

[https://blog.csdn.net/a1056244734/article/details/106802008](https://blog.csdn.net/a1056244734/article/details/106802008" \t "http://t.zoukankan.com/_blank)

#### **10、Angular ngFor循环的使用:**

属性index、count、first、last、even、odd

* index属性提供当前对象的索引
* count提供当前数据集的长度，类似于datasource.length
* first返回当前列表项是否为第一个
* last返回当前列表项是否为最后一个
* even返回当前列表项index是否为偶数，通常用在增加样式用来区分行与行之间
* odd返回当前列表项index是否为奇数

<ul>

<li \*ngFor="let item of datasource; let o=odd,let e=even" [ngClass]="{odd-action: o,even-action: e}">

<card-item [item]="item"></card-item>

</li>

</ul>

[https://www.jianshu.com/p/a35dc3e283cd](https://www.jianshu.com/p/a35dc3e283cd" \t "http://t.zoukankan.com/_blank)

#### **11、AngularJS ng-repeat 循环使用：**

<h1 ng-repeat="x in records">{{x}}</h1>

<script>

var app = angular.module("myApp", []);

app.controller("myCtrl", function($scope) {

$scope.records = [

"菜鸟教程1",

"菜鸟教程2",

"菜鸟教程3",

"菜鸟教程4",

]

});

</script>

#### **12、Angular ng-if判断使用：**

//在angular中没有else只能都通过ng-if来判断

<p ng-if="OwnStatus==0">准备中</p>

<p ng-if="OwnStatus==1">进行中</p>

<p ng-if="OwnStatus==2">已经完成</p>

#### **13、this.router.navigateByUrl跳转：**

<button (click)="Transfer()">页面调转</button>

在typescript中引用Route：

import { Router } from '@angular/router';

在构造函数里注入：

constructor(private router: Router) { }

调转事件：

Transfer() {

//调转到指定的页面

this.router.navigateByUrl("workspace/ClassInfo");

}

#### **14、routerlink跳转：**

routerLink本身支持两种写法：

<a routerLink="UserDetail"></a>

<a [routerLink]="['UserDetail']"></a>

#### **15、[angular]遍历Array的方法：for, forEach, every，some：**

****for…of：****

let someArray = [1, "string", false];

for (let entry of someArray) {

console.log(entry); // 1, "string", false

}

****for循环：****

let someArray = [1, "string", false];

for (var i = 0; i < someArray.length; i ++) {

console.log(someArray[i]); // 1, "string", false

}

****for…in：****

let list = [4, 5, 6];

for (let i in list) {

console.log(i); // "0", "1", "2",

}

for (let i of list) {

console.log(i); // "4", "5", "6"

}

****forEach：****

forEach其实是JavaScript的循环语法，TypeScript作为JavaScript的语法超集，当然默认也是支持的。

let list = [4, 5, 6];

list.forEach((val, idx, array) => {

// val: 当前值

// idx：当前index

// array: Array

});

****every和some：****

every和some也都是JavaScript的循环语法，TypeScript作为JavaScript的语法超集，当然默认也是支持的。因为forEach在iteration中是无法返回的，所以可以使用every和some来取代forEach。

let list = [4, 5, 6];

list.every((val, idx, array) => {

// val: 当前值

// idx：当前index

// array: Array

return true; // Continues

// Return false will quit the iteration

});

参考文章：[https://www.jianshu.com/p/2f0c4d81e934](https://www.jianshu.com/p/2f0c4d81e934" \t "http://t.zoukankan.com/_blank)

#### **AngularJS 指令大全：**

|  |  |
| --- | --- |
| **指令** | **描述** |
| [ng-app](https://www.runoob.com/angularjs/ng-ng-app.html) | 定义应用程序的根元素。 |
| [ng-bind](https://www.runoob.com/angularjs/ng-ng-bind.html) | 绑定 HTML 元素到应用程序数据 |
| [ng-bind-html](https://www.runoob.com/angularjs/ng-ng-bind-html.html) | 绑定 HTML 元素的 innerHTML 到应用程序数据，并移除 HTML 字符串中危险字符 |
| [ng-bind-template](https://www.runoob.com/angularjs/ng-ng-bind-template.html) | 规定要使用模板替换的文本内容 |
| [ng-blur](https://www.runoob.com/angularjs/ng-ng-blur.html) | 规定 blur 事件的行为 |
| [ng-change](https://www.runoob.com/angularjs/ng-ng-change.html) | 规定在内容改变时要执行的表达式 |
| [ng-checked](https://www.runoob.com/angularjs/ng-ng-checked.html) | 规定元素是否被选中 |
| [ng-class](https://www.runoob.com/angularjs/ng-ng-class.html) | 指定 HTML 元素使用的 CSS 类 |
| [ng-class-even](https://www.runoob.com/angularjs/ng-ng-class-even.html) | 类似 ng-class，但只在偶数行起作用 |
| [ng-class-odd](https://www.runoob.com/angularjs/ng-ng-class-odd.html) | 类似 ng-class，但只在奇数行起作用 |
| [ng-click](https://www.runoob.com/angularjs/ng-ng-click.html) | 定义元素被点击时的行为 |
| [ng-cloak](https://www.runoob.com/angularjs/ng-ng-cloak.html) | 在应用正要加载时防止其闪烁 |
| [ng-controller](https://www.runoob.com/angularjs/ng-ng-controller.html) | 定义应用的控制器对象 |
| [ng-copy](https://www.runoob.com/angularjs/ng-ng-copy.html) | 规定拷贝事件的行为 |
| [ng-csp](https://www.runoob.com/angularjs/ng-ng-csp.html) | 修改内容的安全策略 |
| [ng-cut](https://www.runoob.com/angularjs/ng-ng-cut.html) | 规定剪切事件的行为 |
| [ng-dblclick](https://www.runoob.com/angularjs/ng-ng-dblclick.html) | 规定双击事件的行为 |
| [ng-disabled](https://www.runoob.com/angularjs/ng-ng-disabled.html) | 规定一个元素是否被禁用 |
| [ng-focus](https://www.runoob.com/angularjs/ng-ng-focus.html) | 规定聚焦事件的行为 |
| ng-form | 指定 HTML 表单继承控制器表单 |
| [ng-hide](https://www.runoob.com/angularjs/ng-ng-hide.html) | 隐藏或显示 HTML 元素 |
| [ng-href](https://www.runoob.com/angularjs/ng-ng-href.html) | 为 the <a> 元素指定链接 |
| [ng-if](https://www.runoob.com/angularjs/ng-ng-if.html) | 如果条件为 false 移除 HTML 元素 |
| [ng-include](https://www.runoob.com/angularjs/ng-ng-include.html) | 在应用中包含 HTML 文件 |
| [ng-init](https://www.runoob.com/angularjs/ng-ng-init.html) | 定义应用的初始化值 |
| ng-jq | 定义应用必须使用到的库，如：jQuery |
| [ng-keydown](https://www.runoob.com/angularjs/ng-ng-keydown.html) | 规定按下按键事件的行为 |
| [ng-keypress](https://www.runoob.com/angularjs/ng-ng-keypress.html) | 规定按下按键事件的行为 |
| [ng-keyup](https://www.runoob.com/angularjs/ng-ng-keyup.html) | 规定松开按键事件的行为 |
| [ng-list](https://www.runoob.com/angularjs/ng-ng-list.html) | 将文本转换为列表 (数组) |
| [ng-model](https://www.runoob.com/angularjs/ng-ng-model.html) | 绑定 HTML 控制器的值到应用数据 |
| [ng-model-options](https://www.runoob.com/angularjs/ng-ng-model-options.html) | 规定如何更新模型 |
| [ng-mousedown](https://www.runoob.com/angularjs/ng-ng-mousedown.html) | 规定按下鼠标按键时的行为 |
| [ng-mouseenter](https://www.runoob.com/angularjs/ng-ng-mouseenter.html) | 规定鼠标指针穿过元素时的行为 |
| [ng-mouseleave](https://www.runoob.com/angularjs/ng-ng-mouseleave.html) | 规定鼠标指针离开元素时的行为 |
| [ng-mousemove](https://www.runoob.com/angularjs/ng-ng-mousemove.html) | 规定鼠标指针在指定的元素中移动时的行为 |
| [ng-mouseover](https://www.runoob.com/angularjs/ng-ng-mouseover.html) | 规定鼠标指针位于元素上方时的行为 |
| [ng-mouseup](https://www.runoob.com/angularjs/ng-ng-mouseup.html) | 规定当在元素上松开鼠标按钮时的行为 |
| [ng-non-bindable](https://www.runoob.com/angularjs/ng-ng-non-bindable.html) | 规定元素或子元素不能绑定数据 |
| [ng-open](https://www.runoob.com/angularjs/ng-ng-open.html) | 指定元素的 open 属性 |
| [ng-options](https://www.runoob.com/angularjs/ng-ng-options.html) | 在 <select> 列表中指定 <options> |
| [ng-paste](https://www.runoob.com/angularjs/ng-ng-paste.html) | 规定粘贴事件的行为 |
| ng-pluralize | 根据本地化规则显示信息 |
| [ng-readonly](https://www.runoob.com/angularjs/ng-ng-readonly.html) | 指定元素的 readonly 属性 |
| [ng-repeat](https://www.runoob.com/angularjs/ng-ng-repeat.html) | 定义集合中每项数据的模板 |
| [ng-selected](https://www.runoob.com/angularjs/ng-ng-selected.html) | 指定元素的 selected 属性 |
| [ng-show](https://www.runoob.com/angularjs/ng-ng-show.html) | 显示或隐藏 HTML 元素 |
| [ng-src](https://www.runoob.com/angularjs/ng-ng-src.html) | 指定 <img> 元素的 src 属性 |
| [ng-srcset](https://www.runoob.com/angularjs/ng-ng-srcset.html) | 指定 <img> 元素的 srcset 属性 |
| [ng-style](https://www.runoob.com/angularjs/ng-ng-style.html) | 指定元素的 style 属性 |
| [ng-submit](https://www.runoob.com/angularjs/ng-ng-submit.html) | 规定 onsubmit 事件发生时执行的表达式 |
| [ng-switch](https://www.runoob.com/angularjs/ng-ng-switch.html) | 规定显示或隐藏子元素的条件 |
| ng-transclude | 规定填充的目标位置 |
| [ng-value](https://www.runoob.com/angularjs/ng-ng-value.html) | 规定 input 元素的值 |