



第三讲 使用元素与事件指令

HTML元素与事件

Name:

Age:

Tel.:

E-mail:

Birth:

教学目标

1. 使用元素指令
2. 处理事件
3. 管理特殊属性

1

使用元素指令

元素指令

- 1、 **ng-if** : 从DOM中添加和移除元素
- 2、 **ng-hide** : 在DOM中显示和隐藏元素
- 3、 **ng-show** : 在DOM中显示和隐藏元素
- 4、 **ng-switch** : 根据表达式显示或隐藏对应的元素
- 5、 **ng-class** : 为某个元素设置class属性
- 6、 **ng-style** : 设置一个或多个CSS属性。

1

ng-if

ng-if 指令用于在表达式为false时移除HTML元素。如果为true，会添加移除元素，并显示。

语法：

```
<element ng-if="expression"> </element>
```

element : 所有HTML 元素

expression : 如果表达式返回 false 则会移除整个元素，
如果为 true，则会添加元素。

①

ng-if 范例

移除或保留 HTML:

```
<input type="checkbox" ng-model="myVar"
      ng-init="myVar = true">
<div ng-if="myVar">
  <h1>Welcome</h1>
  <p>Welcome to my home.</p>
  <hr>
</div>
```

demo:ex01.ng-if.html

②

ng-hide

ng-hide 指令在表达式为 true 时隐藏 HTML 元素。

ng-hide 是 AngularJS 的预定义类，设置元素的 **display** 属性值为 **none**。

语法：

```
<element ng-hide="expression"> </element>
```

element : 所有HTML 元素

expression : 表达式如果返回 true 则隐藏元素。

2

ng-hide 范例

隐藏 HTML:

```
<input type="checkbox" ng-model="myVar">
```

```
<div ng-hide="myVar">
```

```
  <h1>Welcome</h1>
```

```
  <p>Welcome to my home.</p>
```

```
</div>
```

demo:ex02.ng-hide.html

ng-show 指令在表达式为 true 时显示指定的 HTML 元素，否则隐藏指定的 HTML 元素。

语法：

`<element ng-show="expression"> </element>`

element : 所有HTML 元素

expression : 表达式如果返回 true 则显示指定的元素。

显示 HTML:

```
<input type="checkbox" ng-model="myVar">  
<div ng-show="myVar">  
  <h1>Welcome</h1>  
  <p>Welcome to my home.</p>  
</div>
```

demo:ex03.ng-show.html

4

ng-switch

ng-switch 指令根据表达式显示或隐藏对应的部分。

语法：

```
<element ng-switch=" expression " >
```

```
</ element >
```

element : form元素

expression : 表达式会让匹配项显示，不匹配项移除。

ng-switch

- ◆ 对应的子元素使用 **ng-switch-when** 指令，如果匹配选中选择显示，其他为匹配的则移除。
- ◆ 使用 **ng-switch-default** 指令设置默认选项，如果都没有匹配的情况，默认选项会显示。

```
<element ng-switch="expression">
```

```
  <element ng-switch-when="value"> </element>
```

```
  <element ng-switch-when="value"> </element>
```

```
  <element ng-switch-default> </element>
```

```
</element>
```

4

ng-switch

```
<select ng-model="like">
  <option value="运动">运动</option>
</select>
```

```
<div ng-switch="like">
  <div ng-switch-when="运动">
    <h1>运动</h1>... ..
  </div>

  ... ..
  <div ng-switch-default>
    <h1>默认情况</h1> ... ..
  </div></div>
```

demo:ex04.ng-switch.html

ng-class : 用于给 HTML 元素动态绑定一个或多个 CSS 类

ng-class指令的值可以是字符串、对象、数组。

字符串 : 多个类名使用空格分隔。

对 象 : 需要使用 key-value 对。

--- key 为你想要添加的类名。

--- value 是一个布尔值。

---只有在 value 为 true 时类才会被添加。

数 组 : 数组的元素可以是字符串或对象。

语法：

```
<element ng-class="expression"> </element>
```

element : 所有HTML 元素

expression : 表达式返回一个或多个类名。

5

ng-class 范例

```
<select ng-model="home">  
  <option value="sky">Sky</option>  
  <option value="tomato">Tomato</option>  
</select>  
  
<div ng-class="home">  
  <h1>Welcome Home!</h1>  
  <p>I like it!</p>  
</div>
```

demo:ex05.ng-class.html

ng-class-even

ng-class-even 指令用于为 HTML 元素动态的绑定一个或多个 CSS 类，但只作用于偶数行。

ng-class-even 指令需要与 ng-repeat 指令搭配使用。

ng-class-even 指令建议用在表格的样式渲染中，但是所有HTML元素都是支持的。

语法：

`<element ng-class-even="expression"> </element>`

element : 所有HTML 元素

expression : 表达式返回一个或多个类名。

ng-class-even 范例

```
<table ng-controller="myCtrl">  
  <tr ng-repeat="x in records" ng-class-  
    even="'striped'">  
    <td>{{x.Name}}</td>  
    <td>{{x.Country}}</td>  
  </tr>  
</table>
```

demo:ex05.ng-class-even.html

ng-class-odd

ng-class-odd 指令用于为 HTML 元素动态的绑定一个或多个 CSS 类，但只作用于奇数行。

ng-class-odd 指令需要与 ng-repeat 指令搭配使用。

ng-class-odd 指令建议用在表格的样式渲染中，但是所有HTML元素都是支持的。

语法：

```
<element ng-class-odd="expression"> </element>
```

element : 所有HTML 元素

expression : 表达指定一个或多个 CSS 类。

ng-class-odd 范例

```
<table ng-controller="myCtrl">  
  <tr ng-repeat="x in records" ng-class-  
    odd="'striped'">  
    <td>{{x.Name}}</td>  
    <td>{{x.Country}}</td>  
  </tr>  
</table>
```

demo:ex05.ng-class-odd.html

6

ng-style

ng-style 指令为 HTML 元素添加 style 属性。

ng-style 属性值必须是对象，表达式返回的也是对象。

对象由 CSS 属性和值注册，即 key=>value 对。

语法：

`<element ng-style="expression"></element>`

element : 所有HTML 元素

expression : 表达式返回由 CSS 属性和值组成的对象。

6

ng-style 范例

`<h1 ng-style="myObj">添加样式</h1>`

```
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
    $scope.myObj = {
        "color" : "white",
        "background-color" : "coral",
        "font-size" : "60px",
        "padding" : "50px"
    }
});
```

demo:ex06.ng-style.html

2

处理事件

①

事件

AngularJS 有自己的 HTML 事件指令。

ng-blur : 元素失去焦点时被触发

ng-focus : 元素获得焦点时触发

ng-change : 表单元素的内容状态发生变化时被触发

ng-submit : 表单被提交时触发

ng-keydown : 按下按键

ng-keypress : 按下按键

ng-keyup : 松开按键

ng-click : 用户点击鼠标或光标时触发

ng-dbclick : 用户双击鼠标/光标时触发

①

事件

AngularJS 有自己的 HTML 事件指令。

ng-mousedown : 按下鼠标按键时

ng-mouseenter : 鼠标指针穿过元素时

ng-mouseleave : 鼠标指针离开元素时

ng-mousemove : 鼠标指针在指定的元素中移动时

ng-mouseover : 鼠标指针位于元素上方时

ng-mouseup : 在元素上松开鼠标按钮时

ng-copy/ng-cut/ng-paste 为复制,剪切,粘贴事件指定自定义行为

②

ng-blur指令

语法：

`<element ng-blur="expression"> </element>`

element: `<a>`, `<input>`, `<select>`, `<textarea>`, 窗口对象

expression：失去焦点时执行的表达式。

ng-blur 指令用于告诉 AngularJS在HTML 元素失去焦点时需要执行的表达式。

ng-blur 指令不会覆盖原生的onblur事件，如果触发该事件，ng-blur 表达式与原生的onblur事件都会执行。

②

ng-blur 范例

例1：

```
<input ng-blur="count = count + 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

例2：

```
<input onblur="alert('失去焦点');" ng-blur="count = count  
+ 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

demo:ex07.ng-blur.html

3

ng-keydown指令

语法：

`<element ng-keydown="expression"> </element>`

element: `<input>`, `<select>`, `<textarea>`, 和其他可编辑元素

expression：按下按键时执行的表达式。

ng-keydown 指令用于告诉 AngularJS在HTML 元素上按下按键时需要执行的表达式。

ng-keydown 指令不会覆盖原生的onkeydown事件，如果触发该事件，ng-keydown 表达式与原生的onkeydown事件都会执行。

3

ng-keydown指令

按键敲击的事件顺序:

Keydown

Keypress

Keyup

3

ng-keydown范例

例1 :

```
<input ng-keydown="count = count + 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

例2 :

```
<input onkeydown="alert('按键');" ng-keydown="count = count  
+ 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

demo:ex08.ng-keydown.html

4

ng-click指令

语法：

```
<element ng-click="expression"> </element>
```

element: 所有html元素

expression：点击时执行的表达式。

ng-click指令用于告诉 AngularJS在HTML 元素被点击后需要执行的表达式。

ng-click指令不会覆盖原生的onclick事件，如果触发该事件，ng-click表达式与原生的onclick事件都会执行。

4

ng-click范例

例1：

```
<input ng-click="count = count + 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

例2：

```
<input type="button" value="点击" onclick="alert('按键');" ng-  
click="count = count + 1" ng-init="count=0" />  
<h1>{{count}}</h1>
```

demo:ex09.ng-click.html

5

ng-copy指令

语法：

```
<element ng-copy="expression"> </element>
```

element: 所有html元素

expression : 拷贝时执行的表达式。

ng-copy指令用于告诉 AngularJS在HTML 元素被复制时需要执行的表达式。

5

ng-copy范例

例1：

```
<input ng-copy="count = count + 1" ng-init="count=0" />
```

```
<h1>{{count}}</h1>
```

低版本不支持！

demo:ex10.ng-copy.html

3

管理特殊属性

1

ng-href

ng-href : 为<a>元素指定链接地址

语法 :

```
< a ng-href = " string " > < /a >
```

string: 字符串, 表达式的执行结果

- ◆ ng-href 指令覆盖了原生的<a>元素 href 属性,
- ◆ 如果在 href 的值中有 AngularJS 代码, 则需要使用 ng-href 而不是 href。
- ◆ ng-href 指令确保了链接是正常的, 即使在AngularJS执行代码前点击链接。

②

ng-src

ng-src : 为元素指定图片的链接地址

语法：

```
<img ng-src= " string " / >
```

string: 字符串，表达式的执行结果

- ◆ ng-src 指令覆盖了 元素的 src 属性。
- ◆ 如果使用了AngularJS代码设置图片地址，需用 ng-src 指令替代src属性。
- ◆ ng-src 指令确保的 AngularJS 代码执行前不显示图片。

本课小结

1. 使用元素指令
2. 处理事件
3. 管理特殊属性

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net