



第二讲 数据绑定和模板指令

教学目标

1. 数据绑定
2. 模板指令

1

数据绑定

①

表达式

AngularJS 使用 **表达式** 把数据绑定到 HTML。

语法 : **{{ expression }}**

expression : 包含文字、运算符和变量

作用 :

AngularJS 将在表达式书写的位置"输出"该表达式的值。

①

表达式中可以执行哪些式子？

算术运算

`{{3+2}}`

5

比较运算

`{{3>4}}`

false

逻辑运算

`{{3>4 || 3>2}}`

true

三目运算

`{{20>18?"成年":"未成年"}}`

成年

赋值运算

`{{a=5}}`

5

demo:ex00.html

①

表达式中可以执行哪些式子？

变量

`{{a}}`

非表达式

~~`{{Hello World!}}`~~

调用string的方法和属性

`{{str.length}}`

创建新对象

`{{ man.name }}`

创建数组

`{{ arr[1] }}`

demo:ex00.html

2

声明变量初始值

1、ng-init 指令：初始化应用程序数据

例：

```
<div ng-init="man={name:'tom',age:17}">
    {{man.name}}
</div>
```

试一试：定义一个字符串，输出字符串的长度？

```
<div ng-init="str='hello world!'">
    {{str.length}}
</div>
```

demo:ex01.html

2

声明变量初始值

2、在AngularJS模型中声明变量

```
<script>  
var app=angular . module( 'md1',[ ]);  
app . controller ( 'mc1',function($scope){  
    $scope . man={"name":"Tom","age":20};  
})  
</script>
```

```
<div> {{man.name}} </div>
```


②

声明变量初始值

试一试：定义一个数组，并输出？

```
<script>  
var app=angular . module( 'md1',[ ]);  
app . controller ( 'mc1',function($scope){  
    $scope . arr=[10,20,30];  
})  
</script>
```

```
<div> {{arr[0]}} </div>
```

```
<div>{{arr[0]+' , '+arr[1]+' , '+arr[2]}}</div>
```

思考：可以使用for循环程序输出数组吗？ **不可以！**

注

AJS与JS不互通

AngularJS 表达式与 JavaScript 表达式不同：

- ◆ AngularJS 表达式可以直接写在 HTML 中。
- ◆ AngularJS 表达式不支持条件判断，循环及异常。
- ◆ AngularJS 表达式支持过滤器。

注

ng与JS不互通

举个例子：

```
<div>
```

```
  {{var aa=10,bb=20;}}
```

```
  {{aa+20}}
```

```
</div>
```

语法错误，将会原样输出

```
{{var a=10,b=20;}}{{a+b}}
```

注

ng与JS不互通

AngularJS基于js语法，但不能直接使用js.

错误：

```
<button onclick="alert(a++)"></button>
```



`a=a+1`

`{{a}}`

`a+=1` ✗

正确：

```
<button onclick="a=a+1">输出a</button>  
{{a}}
```

数据绑定使用模型中的值并将其插入到HTML文档中。

1、ng-bind指令

2、ng-bind-template指令

3、ng-model指令

3

数据绑定

1、ng-bind

绑定一个HTML元素的innerText属性，属于单向数据绑定，从数据模型中获得值并插入到HTML元素中。

```
<div ng-init=" str='Hello World!' ">
```

```
<p ng-bind="str"></p>
```

```
</div>
```

F12

```
<div ng-init="str='Hello World!'">  
  <p ng-bind="str" class="ng-binding">Hello World!</p>  
</div>
```

2、ng-bind-template

ng-bind-template 指令将给定表达式的值替换 HTML 元素的内容。ng-bind-template 允许在属性值中指定多个模板表达式。

```
<div ng-bind-template="First:{{5+5}}. Second:{{3>3}}">  
</div>
```

demo:ex02.html

3

数据绑定

3、ng-model

ng-model 创建一个**双向数据绑定**，绑定了 HTML 表单元元素到 **\$scope** 变量中。

语法:

```
<element ng-model="name"> </element>
```

element : <input>, <select>, <textarea>

4

双向数据绑定

例1 : `<input ng-model="name"> {{ name }}`

hello hello

双向数据绑定:

即HTML中呈现的view与AngularJS中的数据是一致的.
修改其一, 则对应的另一端也会相应地发生变化.

demo:ex03.html

4

双向数据绑定

例2 : `<input ng-model="name">`

`<input ng-model="name">`

hello	hello
hello word	hello word

demo:ex03.html

4

双向数据绑定

例3： 单价：<input **ng-model**= "price">

数量：<input **ng-model**= "num">

总价：{{price*num}}

初始：

单价： 数量： 总价： NaN

输入数据后：

单价： 数量： 总价： 2.5

demo:ex03.html

2

模板指令

①

使用模板指令

数据绑定是AngularJS视图中的核心特性，但是从其自身来说却受到许多限制。WEB应用程序（或其实是任何类型的应用程序）往往都在相似的数据对象集合上进行操作，并且使展示给用户的视图随着不同的数据值而变化。

①

使用模板指令

1、ng-cloak

在加载时防止 AngularJS 代码未加载完而出现的问题。

2、ng-repeat

对数组中或对象某个属性中的每个对象生成一个元素及其内容的若干新拷贝

3、ng-include

向DOM中加载、处理和插入一段HTML

1

ng-cloak

页面加载时防止应用闪烁

```
<div ng-app="" >  
  <p ng-cloak> {{ 5 + 5 }} </p>  
</div>
```

AngularJS 应用在加载时，文档可能会由于 AngularJS 代码未加载完而出现显示 AngularJS 代码，进而会有闪烁的效果，**ng-cloak** 指令是为了防止该问题的发生。所有 HTML 元素都支持。**ng-cloak** 指令没有参数。

②

ng-repeat

ng-repeat 指令用于**循环输出**指定次数的HTML元素。

语法：

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

`expression`：定义了如何循环**集合**(必须是数组或对象)。

expression表达式实例规则：

(1) `x in data`

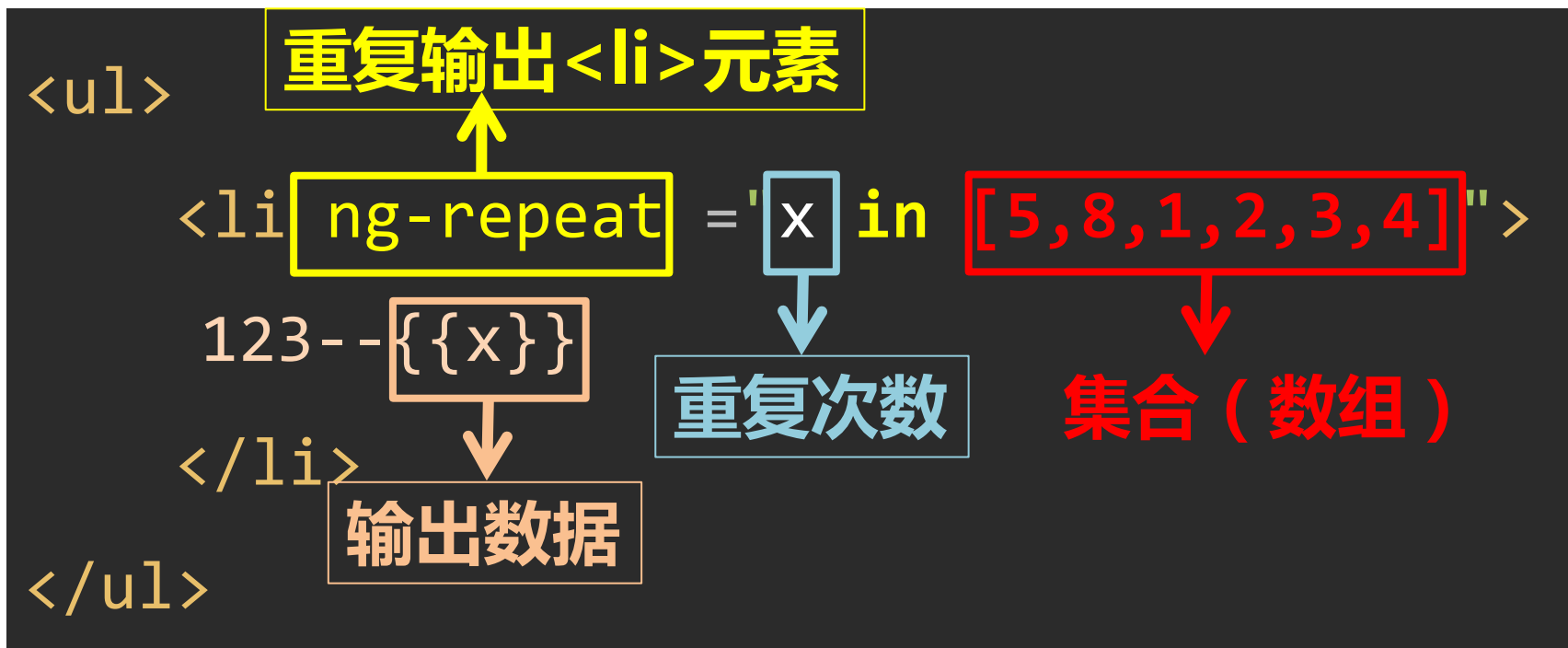
(2) `(key , value) in data`

(3) `x in data track by id`

2

ng-repeat

(1) x in data 重复输出**数组**数据

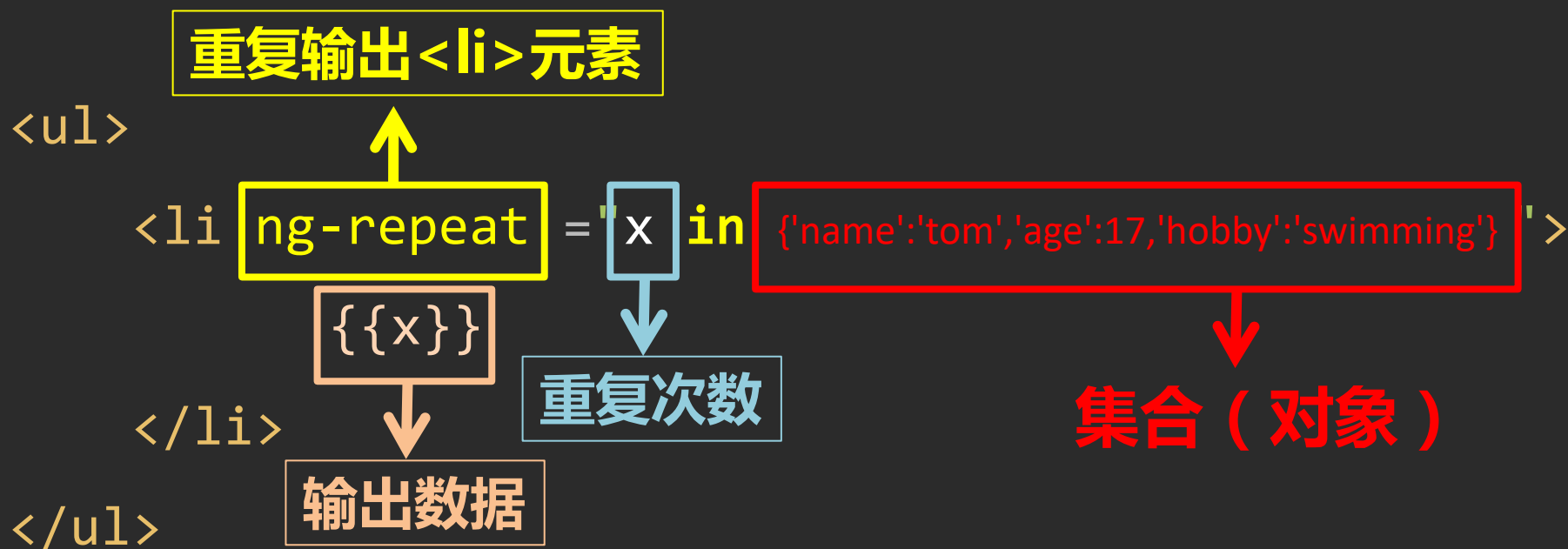


demo:ex05.html

2

ng-repeat

(1) x in data 重复输出**对象**数据



demo:ex06.html

实例：创建表格

试一试：使用ng-repeat指令输出如下表格。

姓名	年龄	爱好
TOM	17	singing
JIM	18	swimming
MARRY	15	dancing

demo:ex07.html

2

ng-repeat

(2) (key , value) in data 重复输出**对象**数据

```
<ul ng-init= " people1 =  
    {'name':'tom','age':17,'hobby':'swimming'} " >  
    <li ng-repeat = " ( x , y ) in people1 " >  
        {{ x }} : {{ y }}  
    </li>  
</ul>
```

demo:ex08.html

2

ng-repeat

(3) x in data track by id 解决重复数据遍历的错误

对于数字或者字符串等基本数据类型来说，它的id就是它自身的值。因此数组中是不允许存在两个相同的数字的。为了规避这个错误，需要定义自己的track by表达式。

```
<ol>
  <li ng-repeat="x in arr track by $index">
    <p>{{x}}</p>
  </li>
</ol>
```

demo:ex09.html

②

ng-repeat

ng-repeat的内置变量:

\$index :返回当前对象或属性的位置

\$first :在当前对象为集合中的第一个对象时返回true

\$middle: 在当前对象既不是集合中的第一个也不是最后一个对象时返回true

\$last :在当前对象为集合中的最后一个对象时返回true

\$even :对于集合中偶数编号的对象返回true

\$odd: 对于集合中奇数编号的对象返回true

这些变量的一个典型应用就是对表格元素生成经典的条纹效果



```
<div ng-controller="mc1">  
  <ul ng-init="data=[10,20,30,40,50]">  
    <li ng-repeat="x in data">  
      编号{{ $index+1 }}: {{ x }},  
      第一个元素: {{ $first }},  
      中间的元素: {{ $middle }},  
      最后一个元素: {{ $last }}  
    </li>  
  </ul>  
</div>
```

demo:ex10.html



ng-repeat-start 表示含有多个顶层元素的重复区域的开始部分

ng-repeat-end 表示含有多个顶层元素的重复区域的结束部分

```
<ul>
  <li ng-repeat-start="user in users">
    {{user.name}}</li>
  <li>the age is {{user.age}}</li>
  <li ng-repeat-end>{{user.email}}</li>
</ul>
```


3

ng-include指令

ng-include 指令用于包含外部的 HTML 文件。

语法1：

```
<element ng-include="filename" onload="expression"  
autoscroll="expression" >  
</element>
```

filename: 文件名，可以使用表达式来返回文件名。

onload: 可选，文件载入后执行的表达式。

autoscroll: 可选，包含的部分是否在指定视图上可滚动。

3

ng-include指令

ng-include 指令用于包含外部的 HTML 文件。

语法2：

```
<ng-include src="filename" onload="expression"  
autoscroll="expression" ></ng-include>
```

filename: 文件名，可以使用表达式来返回文件名。

onload: 可选，文件载入后执行的表达式。

autoscroll: 可选，包含的部分是否在指定视图上可滚动。

3

ng-include指令

```
<div ng-controller="mc1">  
  方法一:  
  <div ng-include="table"></div>  
  方法二: <br>  
  <ng-include src="table"></ng-include>  
</div>
```

demo:ex11.html

包含的内容将作为指定元素的子节点。

ng-include 属性的值可以是一个表达式，返回一个文件名。

默认情况下，包含的文件需要包含在同一个域名下。

本课小结

1. 数据绑定

{{ }}: 表达式输出

ng-init : 初始化数据

ng-bind : 绑定一个HTML元素的innerText属性

ng-model: 双向数据绑定

2. 模板指令

ng-repeat: 重复输出HTML元素

ng-include: 包含外部的 HTML 文件

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net