



第六讲 过滤器

过滤器

AngularJS过滤器用来格式化输出给用户的数据。除了格式化数据，过滤器还能修改DOM。

教学目标

1. 过滤单个数据的值
2. 过滤集合
3. 自定义过滤

1

过滤单个数据的值

①

为什么使用过滤器

过滤器将数据在被指令处理并显示到视图中之前进行转换，而不必修改作用域中原有的数据，这样能够允许数据在应用中的不同部分以不同形式得以展示。

①

过滤器

过滤器	描述
currency	格式化数字为货币格式。
number	格式化通用的数字值
date	格式化日期
uppercase	格式化字符串为大写。
lowercase	格式化字符串为小写。
json	生成js对象的JSON表达式

②

格式化货币值

过滤器可以使用一个管道字符 “|” 添加到表达式和指令中。

currency 过滤器将数字格式化为货币格式：

```
{{ 99.98 | currency }}
```



\$99.98

Demo: Listing02.html

例：

数量:

价格:

<p>总价 = {{ (num * price) | currency }}</p>

②

更换货币符号

格式化货币时可以指定参数设置货币符号。

```
{{p.price | currency:"¥" }}
```

名称	类型	有效期	价格
苹果	水果	10	¥1.20
香蕉	水果	7	¥2.42
梨	水果	6	¥2.02
金枪鱼	鱼	3	¥20.45
鲈鱼	鱼	2	¥17.93
鲤鱼	鱼	4	¥12.93
啤酒	酒	365	¥2.99
葡萄酒	酒	365	¥8.99
鸡尾酒	酒	365	¥45.99

Demo:Listing03.html

3

格式化数字

number过滤器格式化数字类型的数据值，调整小数位数，根据需要进行四舍五入。

{{p.price | **number: 0**}}

→ 保留小数位数

名称	类型	有效期	价格
苹果	水果	10	\$1
香蕉	水果	7	\$2
梨	水果	6	\$2
金枪鱼	鱼	3	\$20
鲈鱼	鱼	2	\$18
鲤鱼	鱼	4	\$13
啤酒	酒	365	\$3
葡萄酒	酒	365	\$9
鸡尾酒	酒	365	\$46

number过滤器会自动在千分位处插入逗号，例如12345将会被 转成12,345

Demo:Listing04.html

格式化日期

date过滤器用于格式化日期。日期可以是字符串、js日期对象、毫秒数等等。

```
{{p.price | date: '格式化字符串' }}
```

格式化字符串	描述
yyyy	对年份的四位数表示（如：2017）
yy	对年份的两位数表示（如：17）
MMMM	月份的全称（如：january）
MM	数字形式的月份，补齐为两字符（例：01）
M	数字形式的月份，无补齐字符（例：1）
dd	每月的第几日，补齐为两字符（例：02）
d	每月的第几日，无补齐字符（例：2）

格式化日期

格式化字符串	描述
HH	24小时制的小时数，补齐为两个字符（如：03）
H	24小时制的小时数，无补齐（如：3）
hh	12小时制的小时数，补齐为两个字符（如：04）
h	12小时制的小时数，无补齐（如：4）
mm	分钟数，补齐为两个字符（如：05）
m	分钟数，无补齐（如：5）
ss	秒钟数，补齐为两个字符（如：06）
s	秒钟数，无补齐（如：6）
a	上午/下午的标志

格式化日期

date过滤器用于格式化日期。日期可以是字符串、js日期对象、毫秒数等等。

```
{{ date1 | date:"yy 年 MM 月 dd 日" }}
```

```
{{ date2 | date:"yy-MM-dd" }}
```

```
{{ date3 | date:"yy-MM-dd hh:mm:ss" }}
```

Demo:Listing 05.html

5

字符串大小写

uppercase和lowercase过滤器用于将字符串转换为全大写或全小写，适用于英文字符。

{{p.name | **uppercase** }} → 转为全大写

{{p.category | **lowercase** }} → 转为全小写

Demo: Listing 06.html

5

生成json

json过滤器用于从js对象创建json字符串。

{{p.name | **json** }}

→ 转为json

名称	类型
{ "name": "苹果", "category": "水果", "price": 1.2, "expiry": 10 }	
{ "name": "香蕉", "category": "水果", "price": 2.42, "expiry": 7 }	
{ "name": "梨", "category": "水果", "price": 2.02, "expiry": 6 }	
{ "name": "金枪鱼", "category": "鱼", "price": 20.45, "expiry": 3 }	
{ "name": "鲈鱼", "category": "鱼", "price": 17.93, "expiry": 2 }	
{ "name": "鲤鱼", "category": "鱼", "price": 12.93, "expiry": 4 }	
{ "name": "啤酒", "category": "酒", "price": 2.99, "expiry": 365 }	
{ "name": "葡萄酒", "category": "酒", "price": 8.99, "expiry": 365 }	
{ "name": "鸡尾酒", "category": "酒", "price": 45.99, "expiry": 365 }	

Demo:Listing 07.html

2

过滤集合

①

限制项目的数量

limitTo过滤器可以限制从一个数据对象构成的数组中取出的项目数量。

```
ng-repeat=" p in prods | limitTo : num "
```

过滤器关键字 显示的个数

也就是说limitTo过滤器来限制ng-repeat指令，使其重复数组的前n（n：limitTo的参数）个元素。

Demo:Listing 09.html

②

选取项

filter过滤器用于从数组中选出一些对象。选取条件可以指定为一个表达式，或者一个用于匹配属性值的对象，或者是一个函数。

```
ng-repeat="p in prods | filter : { name: '啤酒' }"
```



过滤器关键字 对象

Demo:Listing 10.html

3

对项目排序

orderBy过滤器可以对数组中的对象进行排序。

```
ng-repeat="p in prods | orderBy : 'price' "
```



过滤器关键字 匹配的属性

注意：属性名使用了引号，如果不加引号，orderBy过滤器会认为是一个作用域或者控制器变量，将不能实现排序。

Demo:Listing 12.html

③

对项目排序—设置排序方向

orderBy过滤器默认排序为升序排序，可以通过“+”和“-”字符显示设置排序方向。

```
ng-repeat="p in prods | orderBy : 'price'
```

```
ng-repeat="p in prods | orderBy : '-price' "
```

-号为降序排序，+号为升序排序

Demo:Listing 13.html

3

对项目排序—使用函数排序

orderBy过滤器的参数可以使用一个基于多属性执行排序功能的函数。

```
ng-repeat="p in prods | orderBy : myfun"
```

↓ ↓

过滤器关键字 函数名

Demo:Listing 14.html

③

对项目排序—多参数排序

orderBy过滤器的参数可以设置多个，当一个条件无法达到预期想要的排序效果时，可再增设过滤参数，后面的参数将会在第一个参数过滤后的基础上再进行过滤。



Demo:Listing 16.html

4

链式过滤器

可以将过滤器串联起来使用，创建出更复杂的效果。

```
ng-repeat= “ p in products  
            | orderBy:[myCustomSorter, '-price']  
            | limitTo: 5 ”
```

Demo:Listing 17.html

3

自定义过滤

1

创建格式化数据值的过滤器

1、使用module.filter方法定义过滤器

```
angular.module("exampleApp")  
  .filter("labelCase", function () { ..... });
```

2、调用定义的过滤器

```
<td>{{p.name|labelCase}}</td>  
<td>{{p.category|labelCase:true}}</td>
```

Demo:Listing 19.html

①

创建格式化数据值的过滤器

```
angular.module("exampleApp")
  .filter("labelCase", function () {
    过滤器名称      工厂函数
    return function (value, reverse) {
      .....
    };
  });
```

Diagram illustrating the filter function parameters and the filter definition:

- The filter function signature is `function (value, reverse) {`.
- The parameter `value` is highlighted in blue.
- The parameter `reverse` is highlighted in red.
- The filter definition is shown as `{{p.category | labelCase:true}}`.
- A blue arrow points from the `value` parameter to `p.category` in the filter definition.
- A red arrow points from the `reverse` parameter to `labelCase:true` in the filter definition.

Demo:Listing 19.html

②

创建集合过滤器

1、使用module.filter方法定义过滤器

```
angular.module("exampleApp")  
  .filter("skip", function () { ..... });
```

2、调用定义的过滤器

```
<tr ng-repeat="p in products | skip:2 | limitTo: 5">
```

Demo:Listing 22.html

3

在过滤器基础上搭建新的过滤器

```
.filter("labelCase", function () {  
    return function (value, reverse) {..... };  
})  
  
.filter("skip", function () {  
    return function (data, count) {..... }  
})  
  
.filter("take", function ($filter) {  
    return function (data, skipCount, takeCount) {  
        var skippedData = $filter("skip")(data, skipCount);  
        return $filter("limitTo")(skippedData, takeCount);  
    }  
});
```

Demo:Listing 24.html

本课小结

1. 过滤单个数据的值
2. 过滤集合
3. 自定义过滤

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net