



AngularJS 教程

AngularJS 简介

AngularJS 表达式

AngularJS 指令

AngularJS 模型

AngularJS Scope(作用域)

AngularJS 控制器

AngularJS 过滤器

AngularJS Service

AngularJS Http

AngularJS Select

AngularJS 表格

AngularJS SQL

AngularJS HTML DOM

AngularJS 事件

AngularJS 模块

AngularJS 表单

AngularJS 输入验证

AngularJS API

AngularJS Bootstrap

← AngularJS 控制器

AngularJS Service →

AngularJS 过滤器

过滤器可以使用一个管道字符 (|) 添加到表达式和指令中。

AngularJS 过滤器

AngularJS 过滤器可用于转换数据：

过滤器	描述
currency	格式化数字为货币格式。
filter	从数组项中选择一个子集。
lowercase	格式化字符串为小写。
orderBy	根据某个表达式排列数组。
uppercase	格式化字符串为大写。

表达式中添加过滤器

过滤器可以通过一个管道字符 (|) 和一个过滤器添加到表达式中。

(下面的两个实例，我们将使用前面章节中提到的 person 控制器)

uppercase 过滤器将字符串格式化为大写：

AngularJS 实例

[AngularJS 包含](#)[AngularJS 动画](#)[AngularJS 依赖注入](#)[AngularJS 路由](#)[AngularJS 应用](#)

AngularJS 实例

[AngularJS 实例](#)

AngularJS 参考手册

[AngularJS 参考手册](#)

```
<div ng-app="myApp" ng-controller="personCtrl">

  <p>姓名为 {{ lastName | uppercase }}</p>

</div>
```

[尝试一下 »](#)

lowercase 过滤器将字符串格式化为小写：

AngularJS 实例

```
<div ng-app="myApp" ng-controller="personCtrl">

  <p>姓名为 {{ lastName | lowercase }}</p>

</div>
```

[尝试一下 »](#)

currency 过滤器

currency 过滤器将数字格式化为货币格式：

AngularJS 实例

```
<div ng-app="myApp" ng-controller="costCtrl">

  <input type="number" ng-model="quantity">
  <input type="number" ng-model="price">

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

</div>
```

[尝试一下 »](#)

向指令添加过滤器

过滤器可以通过一个管道字符 (|) 和一个过滤器添加到指令中。

orderBy 过滤器根据表达式排列数组：

AngularJS 实例

```
<div ng-app="myApp" ng-controller="namesCtrl">

<ul>
  <li ng-repeat="x in names | orderBy:'country'">
    {{ x.name + ', ' + x.country }}
  </li>
</ul>

</div>
```

尝试一下 »

过滤输入

输入过滤器可以通过一个管道字符 (|) 和一个过滤器添加到指令中，该过滤器后跟一个冒号和一个模型名称。

filter 过滤器从数组中选择一个子集：

AngularJS 实例

```
<div ng-app="myApp" ng-controller="namesCtrl">

<p><input type="text" ng-model="test"></p>

<ul>
  <li ng-repeat="x in names | filter:test | orderBy:'country'">
    {{ (x.name | uppercase) + ', ' + x.country }}
  </li>
</ul>

</div>
```

尝试一下 »

自定义过滤器

以下实例自定义一个过滤器 **reverse**，将字符串反转：

AngularJS 实例

```
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.msg = "Runoob";
});
app.filter('reverse', function() { //可以注入依赖
    return function(text) {
        return text.split("").reverse().join("");
    }
});
```

尝试一下 »

← AngularJS 控制器

AngularJS Service →



笔记列表



1、uppercase，lowercase 大小写转换

```
{{ "lower cap string" | uppercase }} // 结果: LOWER CAP STRING
{{ "TANK is GOOD" | lowercase }} // 结果: tank is good
```

2、date 格式化

```
{{1490161945000 | date:"yyyy-MM-dd HH:mm:ss"}} // 2017-03-22 13:52:25
```

3、number 格式化 (保留小数)

```
{{149016.1945000 | number:2}}
```

4、currency 货币格式化



```
{{ 250 | currency }} // 结果: $250.00
{{ 250 | currency:"RMB ¥ " }} // 结果: RMB ¥ 250.00
```

5、filter查找

输入过滤器可以通过一个管道字符 (|) 和一个过滤器添加到指令中，该过滤器后跟一个冒号和一个模型名称。

filter 过滤器从数组中选择一个子集

```
// 查找name为iphone的行
{{ [{ "age": 20,"id": 10,"name": "iphone"},
  {"age": 12,"id": 11,"name": "sunm xing"},
  {"age": 44,"id": 12,"name": "test abc"}
] | filter:{'name':'iphone'} }}
```

6、limitTo 截取

```
{{ "1234567890" | limitTo :6 }} // 从前面开始截取6位
{{ "1234567890" | limitTo:-4 }} // 从后面开始截取4位
```

7、orderBy 排序

```
// 根id降序排
{{ [{ "age": 20,"id": 10,"name": "iphone"},
  {"age": 12,"id": 11,"name": "sunm xing"},
  {"age": 44,"id": 12,"name": "test abc"}
] | orderBy:'id':true }}
```

```
// 根据id升序排
{{ [{ "age": 20,"id": 10,"name": "iphone"},
  {"age": 12,"id": 11,"name": "sunm xing"},
  {"age": 44,"id": 12,"name": "test abc"}
] | orderBy:'id' }}
```

尝试一下 »

Qyizos 9个月前 (03-22)



过滤器实例:

```

<div ng-controller="myCtrl">
  <div ng-bind="myName | uppercase"></div> <!--uppercase转换成大写-->
  <div ng-bind="myName | lowercase"></div> <!--lowercase转换成小写-->
  <div class="" ng-bind="money | currency : '¥'"> </div><!--currency 过滤器将数字格式化为货币格式-->
  <div class="" ng-repeat="v in city | orderBy:'id'">
    <p ng-bind="v.name"></p>
  </div><!--orderBy 过滤器根据表达式排列数组-->
  <div class="" ng-repeat="v in city | orderBy:'-id' | filter : ''">
    <p ng-bind="v.name" style="color:red;"></p>
  </div><!--orderBy 过滤器根据表达式排列数组 默认正序asc,倒序添加-负号-->
  <!--filter 过滤器根据表达式过滤不包含过滤器中的内容-->
  <!--自定义注入依赖-->
  <div class="" ng-bind="myName | aa" style="color:blue;"> <!--自定义过滤器aa-->
  </div>
</div>

```

尝试一下 »

修行 8个月前 (04-17)



自定义过滤器 -- 多参数

具体的操作就是在调用的时候用": 参数" 的格式在后面追加即可

```

var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
});
app.filter('myfilter', function() { //可以注入依赖
  return function(text) {
    var newArguments= Array.prototype.slice.call(arguments);
    return text+newArguments.join(',');
  }
});

```

尝试一下 »

iceluo 6个月前 (06-15)



date过滤器的功能是基于要求的格式格式化一个日期成为一个字符串。

格式化字符串的基本参数：

- yyyy: 用4位数字表示年（例如：AD 1 => 0001, AD 2010 => 2010）
- yy: 用两位数字表示年（00-99）（例如：AD 2001 => 01, AD 2010 => 10）
- y: 用一位数字代表年（例如：AD 1 => 1, AD 199 => 199）
- MMMM: 英文全称表示月（January-December）
- MMM: 英文缩写表示月（Jan-Dec）
- MM: 两位数字表示月（01-12）
- M: 月（1-12）
- dd: 两位数字表示日（01-31）
- d: 日（1-31）
- EEEE: 英文全称的一周中的天（Sunday-Saturday）
- EEE: 英文缩写的一周中的天（Sun-Sat）
- HH: 两位数表示24小时制的时（00-23）
- H: 24小时制的时（0-23）
- hh: 两位数字表示上午或下午的时（01-12）
- h: 上午或下午的时（1-12）
- mm: 两位数字表示分（00-59）
- m: 分（0-59）
- ss: 两位数字表示秒（00-59）
- s: 秒（0-59）
- sss: 毫秒（000-999）
- a: AM/PM
- Z: 4位数字（+符号）代表时区偏移量（-1200——+1200）
- ww: 用两位数字表示一年的周数（00-53），第一周（01）是一年中的第一个星期四
- w: 一年的周数(0-53)，第一周（1）是一年中的第一个星期四
- G,GG,GGG: 年代字符串的缩写形式，例如AD（公元）
- GGGG: 年代字符串的全称，例如Anno Domini（公元）

上面这些参数我们可以根据自己的意愿自由组合得到自己想要的格式，例如 yyyy-MM-dd 等。

格式化字符串也提供了一些预定义的本地化格式，可以方便我们使用：

- medium : 'MMM d,y h:mm:ss a' 例如：Sep 3, 2010 12:05:08 PM
- short : 'M/d/yy h:mm a' 例如：9/3/10 12:05 PM
- fullDate: 'EEEE,MMMM d,y' 例如：Friday, September 3, 2010
- longDate: 'MMMM d,y' 例如：September 3, 2010
- mediumDate: 'MMM d,y' 例如：Sep 3, 2010
- shortDate: 'M/d/y' 例如：9/3/10
- mediumTime: 'h:mm:ss a' 例如：12:05:08 PM
- shortTime: 'h:mm a' 例如：12:05 PM

格式化字符串可以包含文本值。这些需要被单引号包围（例如 "h 'in the morning'"），如果想输出一对单引号，就在一个序列中用两个双引号（例如："h 'o'clock'"）

date过滤器的用法：

1.在html中用法：{{ date_expression | date : format : timezone }}

实例：

```
<span>{{1288323623006 | date:'medium'}}</span><br>
<span>{{1288323623006 | date:'yyyy-MM-dd HH:mm:ss Z'}}</span><br>
<span>{{'1288323623006' | date:'MM/dd/yyyy @ h:mma'}}</span><br>
<span>{{'1288323623006' | date:"MM/dd/yyyy 'at' h:mma"}}</span><br>
```

尝试一下 »


2.在javascript中的用法：\$filter('date')(date, format, timezone)

实例：

```
var today = new Date();
$scope.formatDate = $filter('date')(today, 'yyyy-MM-dd');
```

尝试一下 »

学习的孩子 3周前 [11-27]

 点我分享笔记

笔记需要是本篇文章的内容扩展！

[注册邀请码获取方式](#)

Copyright © 2013-2017 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1

反馈/建议