



# AngularJS 基础(二)

讲师：文元



泛IT职业教育机构

## 信·机构

- 慧科教育集团旗下泛IT职业教育培训机构。
- 与找座儿形成国内第一家O2O混合式教学落地基地。
- 良心教学，可信赖大机构！

## 新·模式

- 同时采用最合理的TS教学模式和最合理教学时间安排。
- 全程手把手教学、边学边练，7\*12小时答疑指导，保证学员即学即会。

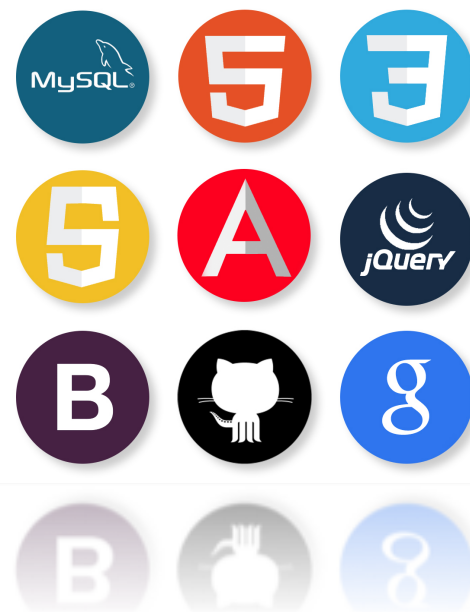
## 心·服务

- 多年教学经验资深讲师倾心教学
- 配备专门的生活导师，提供入学接送、精选宿舍、生活指导、学习督促等一条龙服务。
- 提供专业就业指导服务，简历指导+模拟面试，保证快速就业。

## 薪·未来

- 无限互联学生毕业平均薪资12755元，比同行业机构平均高出12%。
- 98%的无限互联学生在毕业4周内成功就业！

- AngularJS 过滤器
- \$scope 的 \$apply 方法和 \$watch 方法
- AngularJS 工具方法(一)
- AngularJS 工具方法(二)
- 更多指令的讲解(一)
- 更多指令的讲解(二)



# 一、AngularJS 过滤器

1. 几种常用过滤器
2. 指令中添加过滤器
3. 控制器中使用过滤器
4. 自定义过滤器

- 过滤器(filter)接收一个输入,通过某个规则进行处理,然后返回处理后的结果
- 主要用在数据的格式化上
- 用法:可以使用一个管道字符(|)添加到表达式和指令中

- uppercase, lowercase 将字符串大小写转换

```
<div ng-app = "myApp" ng-controller="firstCtrl">  
  {{ 'we are family' | uppercase }} <!-- 结果:WE ARE FAMILY -->  
  {{ 'We Are Family' | lowercase }} <!-- 结果:we are family -->  
</div>
```

- json 将js对象格式化为json字符串

```
<div ng-app = "myApp" ng-controller="firstCtrl">  
  {{ {title: "男神", age: 18} | json }} <!-- 结果{"title": "男神", "age": 18 } -->  
</div>
```

- date 格式化时间

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{1472227200000 | date}}      <!-- 结果:Aug 27, 2016 -->
  {{1472227200000 | date:"MM/dd//yyyy @ h:mm:ss"}} <!-- 08/27//2016 @ 12::00 -->
  {{1472227200000 | date:"yyyy-MM-dd hh:mm:ss"}}   <!-- AM 2016-08-27 12:00:00 -->
</div>
```

- number 格式化

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{1234567 | number}}      <!-- 1,234,567 -->
  {{1.234567 | number}}     <!-- 1.235 -->
  {{1.234567 | number:1}}   <!-- 1.2 -->
</div>
```

- date 格式化时间

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{1472227200000 | date}}      <!-- 结果:Aug 27, 2016 -->
  {{1472227200000 | date:"MM/dd//yyyy @ h:mm:ss"}} <!-- 08/27//2016 @ 12::00 -->
  {{1472227200000 | date:"yyyy-MM-dd hh:mm:ss"}}   <!-- AM 2016-08-27 12:00:00 -->
</div>
```

- number 格式化

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{1234567 | number}}      <!-- 1,234,567 -->
  {{1.234567 | number}}     <!-- 1.235 -->
  {{1.234567 | number:1}}   <!-- 1.2 -->
</div>
```



- currency 货币格式化

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{250 | currency}}           <!-- 结果: $250.00 -->
  {{250 | currency:'¥'}}      <!-- 结果: ¥250.00 -->
</div>
```

- limitTo 字符串,对象的截取

```
<div ng-app = "myApp" ng-controller="firstCtrl">
  {{'I love you' | limitTo:6}}  <!-- 结果: I love -->
  {{'I love you' | limitTo:-6}} <!-- 结果: ve you -->
  {{{['a','b','c','d'] | limitTo:1}} <!-- 结果:['a'] -->
</div>
```

- filter 查找

<!--查找含有2的-->

```
{{ [{ "age": 20, "id": 10, "name": "iphone"}, {"age": 12, "id": 11, "name": "sunm  
xing"}, {"age": 44, "id": 12, "name": "test abc"}  
] | filter:'2' }}
```

<br>

<br>

<!--查找name为iPhone的-->

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

- orderBy 对象排序

*<!--按照id降序排序-->*

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

<br>

<br>

*<!--升序排序-->*

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

- 可以几个过滤器一起使用,只需在一个过滤器后面再接上另外一个过滤器即可

```
<!--查找含有2的,按照升序排列-->
```

```
{{ [{ "age": 20, "id": 15, "name": "iphone"}, {"age": 12, "id": 11, "name": "sunm  
xing"}, {"age": 44, "id": 12, "name": "test abc"}  
] | filter:'2' | orderBy: 'id' }}
```

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

<div>

- 除了在表达式及指令中可以使用过滤器,在控制器中也是可以使用控制器的
- 在控制器中加入另外一个形参 `$filter` 就可以使用控制器了,用法与前面一值

//创建控制器

```
app1.controller('firstCtrl',function ($scope,$filter) {  
    $scope.name1 = $filter('date')('236478234','yyyy-MM-dd HH:mm:ss');  
    $scope.name2 = $filter('uppercase')('heLLo');  
    $scope.name3 = $filter('filter')([{"age": 20,"id": 10,"name": "iphone"},  
        {"age": 12,"id": 11,"name": "sunm xing"},  
        {"age": 44,"id": 12,"name": "test abc"}  
    ],{"name": 'iphone'});  
    console.log($scope.name1);  
    console.log($scope.name2);  
    console.log($scope.name3);  
});
```

- 系统提供的过滤器有的时候并不足以满足项目的需求,这个时候,就需要自定义过滤器
- 调用模块的 fliter 方法即可,第一个参数为过滤器的名称,第二个参数为实现过滤器功能的函数

- 自定义过滤器及调用自定义过滤器

```
//将hello变为'你好',  
// input 为需要添加过滤器的字符串或者对象  
// n1,n2相当于过滤器后面的条件  
app1.filter('rHello',function(){  
    return function(input,n1,n2){  
        console.log(input);  
        console.log(n1);  
        console.log(n2);  
        return input.replace(/hello/, "你好");  
    }  
});
```

```
<!--调用自定义过滤器-->
```

```
{{'hello 中国' | rHello:2:3}}
```



- 完成一个搜索功能

输入过滤:

- 年龄为: HYH 名字为: hyh
- 年龄为: MJJ 名字为: mjj
- 年龄为: CYKK 名字为: cykk

显示界面

输入过滤:

- 年龄为: HYH 名字为: hyh
- 年龄为: CYKK 名字为: cykk

搜索界面

## 二、\$scope 的\$apply 方法和\$watch 方法

1. \$apply 方法的作用
2. \$apply 方法的使用
3. \$watch 方法的使用

- 传播 Model 数据的变化
- 刷新数据
- 使用情景: AngularJS 外部的控制器(DOM 事件、外部的回调函数延迟调用等)调用了 AngularJS 函数之后,必须调用\$apply。在这种情况下,你需要命令AngularJS 刷新自己(模型、视图等),\$apply 就是用来做这件事情的,必须调用\$apply。在这种情况下,你需要命令 AngularJS 刷新自己(模型、视图等),\$apply 就是用来做这件事情的

- 只要可以,请把要执行的代码和函数传递给\$apply 去执行,而不要自己执行那些函数然后再调用\$apply


```
//2秒钟过后,界面上的女神变为了男神|
$scope.title = '女神';
setTimeout(function () {
    $scope.$apply(function () {
        $scope.title = '男神';
    });
}, 2000);
```

# \$watch 方法的作用

- \$watch 方法主要是用来监视 Model 的变化
- \$watch 方法中有两个参数,第一个参数为要观察的属性,第二个参数为执行函数,执行函数中会有两个值,一个变化后的值,一个变化前的值

```
app.controller('firstCtrl',function ($scope,$interval) {  
    //2秒钟过后,界面上的女神变为了男神  
    $scope.count = 10;  
    //interval相当于js中的定时器  
    $scope.timer = $interval(function () {  
        $scope.count--;  
    },1000);  
  
    $scope.$watch('count',function (newVal,oldVla) {  
        console.log(newVal + ',' + oldVla);  
        //到5秒钟的时候关闭定时器  
        if (newVal == 5) {  
            $interval.cancel($scope.timer);  
        }  
    });  
});
```

**观察的属性**



- 模拟淘宝,根据商品价格和个数,显示出费用运费和总额,当费用超过100时,则运费可免,如下图所示:

价格:	5
个数:	1

费用:\$5.00  
运费:\$10.00  
总额:\$15.00

价格:	102
个数:	1

费用:\$102.00  
运费:\$0.00  
总额:\$102.00

## 三、AngularJS 工具方法(一)

1. 简单的工具方法
2. 重点讲述的方法
3. 复杂的工具方法

- `angular.isArray`: 判断一个数据是否是数组
- `angular.isDate`: 判断是否是 `Date` 类型
- `angular.isDefined`: 判断一个数据是否是 `defined` 类型
- `angular.isUndefined`: 判断一个数据是否是 `undefined` 类型
- `angular.isFunction`: 判断一个数据是否是函数
- `angular.isNumber`: 判断一个数据是否是 `number` 类型
- `angular.isObject`: 判断一个数据是否是对象
- `angular.isString`: 判断一个数据是否是字符串



- `angular.uppercase`: 将字符串小写转化为大写
- `angular.lowercase`: 将字符串大写转化为小写
- `angular.equals`: 比较两个数据是否相等
- `angular.fromJson`: 将字符串转化为 json 对象
- `angular.toJson`: 将 json 对象转化为字符串
- `angular.copy`: 拷贝一个对象

- angular.forEach: 对象的遍历

```
app.controller('firstCtrl',function ($scope) {  
    $scope.person = {name: '男神',age: 18};  
    angular.forEach($scope.person,function(value,key){  
        console.log('value:' + value +',key: '+key );  
        // value:男神,key:name  
        // value:18,key:age  
    });  
});
```

遍历的对象



- angular.forEach: 数组的遍历

```
var name = ['zs', 'ls', 'ww', 'tq'];  
angular.forEach(name, function(value, index){  
    console.log(index + ':' + value);  
    //index代表数组的下标, value代表下标对应的元素值  
});
```

- angular.bind: 绑定对象,作为函数的上下文

```
//绑定对象,作为函数的上下文  
var self = {name: '张三'};  
var f = angular.bind(self,function(age){  
    $scope.info = this.name + 'is' + age;  
    console.log($scope.info); //输出结果为 张三is30  
});  
f(30);
```

## 四、AngularJS 工具方法(二)

1. bootstrap 方法的使用
2. 加载多个module
3. 在 AngularJS 中使用jQuery

- 在 AngularJS 中, App 只会初始化一次,即只加载第一个 module,后面的 module 就不加载了,这里是模块的自动加载
- bootstrap 方法可以加载多个 module,这里其实是手动加载 module
- 先在 js 文件中获取到对应的 DOM 元素
- 再调用 bootstrap 方法,方法的第一个参数为 DOM 元素,第二个参数为以该 App 命名的字符串数组

```
var div1 = document.getElementById('div1');  
var div2 = document.getElementById('div2');  
  
angular.bootstrap(div1, ['myApp1']);  
angular.bootstrap(div2, ['myApp2']);
```

- AngularJS 通常情况下只会加载一个 module ,当要加载多个 module 的时候,可以使用 bootstrap 方法,但此种方式较为麻烦
- 一般加载其他 module 的方式为
  - 将 ng-app 设置在 body 里面
  - 新建一个 js 文件,在此文件中 创建 module 以及控制器
  - 在 html 文件里面导入此 js 文件,在给 body 里面的 app 添加此 app 作为依赖

```
var app1 = angular.module('myApp',['myApp2']);
```

- 有的时候,可能在使用 AngularJS 过程中想使用 jQuery 的方法,在 AngularJS 中,有两种方式使用 jQuery
- 第一种: 直接导入 jQuery 文件使用,此种方式得注意,jQuery 文件的导入必须在导入 AngularJS 文件之前导入

```
app1.controller('firstCtrl',function ($scope) {  
    //直接使用jQuery  
    $('#box').html('<span>这是span标签</span>');  
});
```



- 为了方便调用 jQuery 里面的方法,AngularJS 里面内置了一个 jqLite, jqLite是一个微型的、API兼容的jQuery子集,允许Angular在跨浏览器兼容的方式下操纵DOM。jqLite只实现了所需的最常用功能,并没有原来的 jQuery 强大
- 推荐直接使用第一种方式在 AngularJS 文件中使用jQuery

```
//利用内置的jqLite
app1.controller('firstCtrl',function ($scope) {
    //直接使用jQuery
    var div = document.getElementById('box');
    angular.element(div).html('<span>这是span标签</span>');
});
```

## 五、更多指令的讲解(一)

1. 事件指令
2. 实现防止闪烁的指令
3. input 相关指令

- AngularJS 里面有许多的事件指令,与 js里面的事件类似
- ng-click/dbclick: 双击事件
- ng-mousedown/up: 鼠标在指定的 HTML 元素上按下/松开鼠标按钮时要执行的操作
- ng-mousemove/over/out: 鼠标移动/移入/移出
- ng-keydown/up/press: 在指定 HTML 元素上按下/抬起/按下时要执行的操作(一般用于输入框)

- ng-focus/blur: 输入框获得/失去焦点需要执行的操作
- ng-submit: 表单提交后执行的函数
- ng-copy: 按下复制键的时候执行的事件
- ng-cut: 按下剪切键的时候执行的事件
- ng-paste: 按下粘贴键的时候执行的事件

- ng-cloak :在 angular 中为我们提供了 ng-cloak 来实现防止闪烁的方案, 我们只需要在需要的地方加上 ng-cloak。同时对于 `bing` 文字 (`{{ express }}`) 我们也可以改为 ng-bind 来实现避免
- ng-non-bindable: 如果我们就想要一个 `{{}}` 这样的内容就可以使用 ng-non-bindable ,原型 " 输出 "

- ng-disabled: 设置表单输入字段的 disabled 属性(input, select, 或 textarea), 如果 ng-disabled 中的表达式返回 true 则表单字段将被禁用
- ng-readonly: 用于设置表单域(input 或 textarea) 的 readonly 属性
- ng-checked: 用于设置复选框(checkbox)或单选按钮(radio)的 checked 属性
- ng-value: 用于设置 input 或 select 元素的 value 属性

## 六、更多指令的讲解(二)

1. 样式指令
2. DOM 操作指令

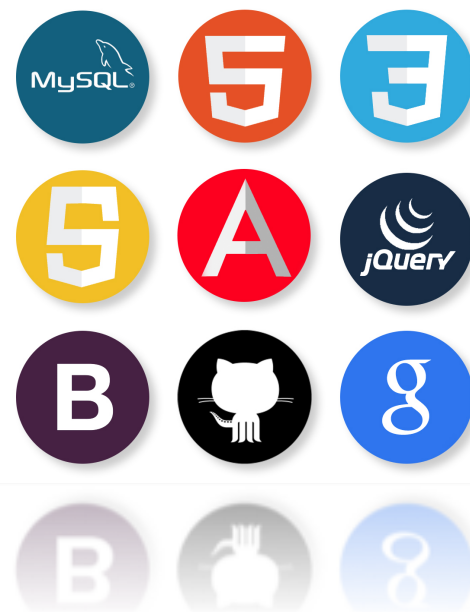
- ng-class: 用于给 HTML 元素动态绑定一个或多个 CSS 类
- ng-class-even: 为表格的偶数行设置 class
- ng-class-odd: 为表格的奇数行设置 class
- ng-style: 为 HTML 元素添加 style 属性
- ng-href: 覆盖了原生的 <a> 元素 href 属性
- ng-attr-(suffix): html属性多了,AngularJS 提供了一种通用的写法



- ng-show: 值设置为 true 的时候显示元素
- ng-hide: 值设置为 true 的时候隐藏元素
- ng-if: 值设置为 true 的时候删除元素
- ng-switch: 指令根据表达式显示或隐藏对应的部分。
  - 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。
  - 你可以通过使用 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

- ng-open: 设置 details 列表的 open 属性
- ng-bind: 防止闪烁效果
- ng-bind-template: 解决 ng-bind 只能绑定一个的问题
- ng-bind-html: 解析 html 代码(得导入 sanitize.js 文件,并给 app 添加 ngSanitize 依赖)
- ng-include: 加载外部页面(传入文件路径就行)
- ng-model-options: 控制双向事件绑定的时候,触发事件的方式
  - updateOn: 'blur' 失去焦点时触发
  - debounce : 1000 1秒后触发

- AngularJS 过滤器
- \$scope 的 \$apply 方法和 \$watch 方法
- AngularJS 工具方法(一)
- AngularJS 工具方法(二)
- 更多指令的讲解(一)
- 更多指令的讲解(二)



- 过滤器的使用必须得有管道符号 | ,自定义过滤器的时候得有过滤器的名字
- 需要刷新视图里面的数据的时候可以调用 \$scope 的 \$apply方法
- 需要观察数据变化的时候可以调用 \$scope 的 \$watch方法(观察的属性应该是以字符串的形式作为参数)
- AngularJS 中只会自动加载一个 module ,可以通过 bootstrap 的方式手动加载其他 module ,也可以通过添加依赖的方式加载其他模块
- ng-bind-html: 解析 html 代码(必须得导入 sanitize.js 文件,并给 app 添加ngSanitize 依赖)



慧科集团旗下企业



官方微信

**未经允许不得将视频用于商业用途，否则将追究其法律责任！**

**官方网站：** <http://www.wuxianedu.com/>

**无限智造 互联精英**

**版权所有：无限互联**