# Angular

## 变量绑定

**ng-app** 指令定义一个 AngularJS 应用程序。

**ng-model** 指令把元素值（比如输入域的值）绑定到应用程序。

<div ng-app="">

<input type="text" ng-model="name" /> {{name}}

</div>

QQ截图20170619140228

## 初始化变量

<div ng-app="" ng-init="firstName='John'">

<p>姓名为 <span ng-bind="firstName"></span></p>

</div>

## 应用

<div ng-app="myapp" ng-controller="mycontrol">

姓<input type="text" ng-model="firstname"/><br />

名<input type="text" ng-model="lastname"/><br />

姓名：{{firstname+" "+lastname}}

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycontrol",function($scope){

$scope.firstname="zhang";

$scope.lastname="san";

});

</script>

## 表达式

### 运算

<div ng-app="" ng-init="price=20;number=2">

总价为：{{price\*number}}

</div>

等价于

<div ng-app="" ng-init="quantity=1;cost=5">

<p>总价： <span ng-bind="quantity \* cost"></span></p>

</div>

### 对象

<div ng-app="" ng-init="apple={price:20,number:2}">

总价为：{{apple.price\*apple.number}}

</div>

### 数组

<div ng-app="" ng-init="fruit=['apple','orange']">

第二个水果是：<span>{{fruit[1]}}</span>

</div>

## 指令

### 数据绑定

通过单个一个ng-model进行数据同步

<div ng-app="" ng-init="name='zhangsan'">

初始化后的值：<input type="text" ng-model="name"/><br />

改变初始化后的值：<span>{{name}}</span>

</div>

通过两个ng-model进行数据同步

<div ng-app="" ng-init="price=0;number=0">

价格：<input type="text" ng-model="price"/><br />

数量：<input type="text" ng-model="number" /> <br />

总价：{{price\*number}}

</div>

### 循环数组

<div ng-app="" ng-init="nums=[1,2,3,4,5,6]">

循环数组结果为:<span ng-repeat="x in nums">{{x+" "}}</span>

</div>

<div ng-app="" ng-init="fruit=[{name:'apple',price:10},{name:'orange',price:20}]">

循环数组结果为:

<span ng-repeat="x in fruit">{{x.name+":"+x.price+" "}}</span>

</div>

### 自定义指令

自定义指令可通过 元素名 属性 类名 注解 来调用

元素名 E

<runoob-directive></runoob-directive>

属性 A

<div runoob-directive></div>

类名 C

<div class="runoob-directive"></div>

注释 M

<!-- directive: runoob-directive -->

<body ng-app="myapp">

<runoob-directive></runoob-directive>

<script>

var app=angular.module("myapp",[]);

app.directive("runoobDirective",function(){

return {

restrict:"", (EACM限制使用)

template:"<h1>自定义指令</h1>"

};

});

</script>

</body>

## 模型

### 双向绑定

将需要初始化的变量用模型来初始化（scope定义的变量与更改后同步改变）

<div ng-app="myapp" ng-controller="mycontrol" >

水果：<span>{{fruit}}</span><br />

单价：<input type="text" ng-model="price" /><br />

数量：<input type="text" ng-model="number" /><br />

总价：{{price\*number}}

</div>

<script>

var app = angular.module("myapp",[]);

app.controller("mycontrol",function($scope){

$scope.fruit="apple";

$scope.price=0;

$scope.number=0;

});

</script>

### 应用状态

在输入框中输入你的邮箱地址，如果不是一个合法的邮箱地址，会弹出提示信息。

<form ng-app="" name="myform">

email:<input type="email" name="myaddress" ng-model="myemail" />

<span ng-show="myform.myaddress.$error.email">邮箱不合法</span>

<p>编辑邮箱地址，查看状态的改变。</p>

Valid: {{myform.myaddress.$valid}} (输入的值是合法的则为 true)。<br />

Dirty: {{myform.myaddress.$dirty}} (值改变则为 true)。<br />

Touched: {{myform.myaddress.$touched}} (通过触屏点击则为true)。<br />

</form>

## Scope

### 案例

<div ng-app="myapp" ng-controller="mycontrol">

精确查找：<input type="text" ng-model="select" />

<table>

<tr ng-repeat="fruit in fruits | filter:select | orderBy:'price'">

<td>水果：{{fruit.name}}</td>

<td>单价：{{fruit.price | currency}}</td>

<td>数量：<input type="text" ng-model="fruit.number" /></td>

<td>总价：{{(fruit.price\*fruit.number) | currency}}</td>

</tr>

</table>

<button ng-click="totalprice()">合计：</button>

<span>{{tprice | currency}}</span>

</div>

<script>

var app = angular.module("myapp", []);

app.controller("mycontrol", function($scope) {

$scope.fruits = [{

name: 'apple',

price: 16,

number: 0

}, {

name: 'orange',

price: 24,

number: 0

}, {

name: 'banana',

price: 14,

number: 0

}];

$scope.totalprice = function() {

$scope.tprice = 0;

for (var i = 0; i < $scope.fruits.length; i++) {

$scope.tprice = $scope.tprice + $scope.fruits[i].price

\* $scope.fruits[i].number;

};

}

});

</script>

## 过滤器

|  |  |
| --- | --- |
| 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。 |

### 格式过滤

<div ng-app="" ng-init="nums=['Jhon','JACK']">

<span ng-repeat="x in nums">{{(x+" ") | uppercase}}</span>

</div>

### 排序

<div ng-app="myapp" ng-controller="mycontrol">

排序输出数组:

<span ng-repeat="x in fruits | orderBy:'price'">

{{x.name+":"+x.price+" "}}

</span>

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycontrol",function($scope){

$scope.fruits=[

{name:'apple',price:13},

{name:'orange',price:9},

{name:'banana',price:16}

];

});

</script>

### 输入过滤

<div ng-app="myapp" ng-controller="mycontrol">

<input type="text" ng-model="fit" />

排序输出数组:

<span ng-repeat="x in fruits | filter:fit | orderBy:'price'">

{{x.name+":"+x.price+" "}}

</span>

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycontrol",function($scope){

$scope.fruits=[

{name:'apple',price:13},

{name:'orange',price:9},

{name:'banana',price:16}

];

});

</script>

## Service

### $http

data.json

[{

"name":"test1"

},{

"name":"test2"

},{

"name":"test3"

}]

<div ng-app="myapp" ng-controller="mycontrol">

<span ng-repeat="x in names">{{x.name+" "}}</span>

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycontrol",function($scope,$http){

$http.get("data.json").success(function(data){

$scope.names=data;

（特殊情况下获取返回值为function（response）{

$scope.names=response.records;}）

});

});

</script>

### $timeout服务（只执行一次）

<div ng-app="myapp" ng-controller="mycontrol">

<h1>{{hw}}</h1>

</div>

<script>

var app = angular.module("myapp",[]); app.controller("mycontrol",function($scope,$timeout){

$scope.hw="hello world";

$timeout(function(){

$scope.hw="Do you happy at this moment"

},2000);

});

</script>

### $interval服务（循环执行）

<div ng-app="myapp" ng-controller="mycontrol">

{{nowtime}}

</div>

<script>

var app = angular.module("myapp",[]);

app.controller("mycontrol",function($scope,$interval){

$scope.nowtime=new Date().toLocaleTimeString();

$interval(function(){

$scope.nowtime=newDate().toLocaleTimeString();

},1000);

});

</script>

## Select

### ng-repeat（常用）

对象数组

<div ng-app="myapp" ng-controller="mycontrol">

<select ng-model="pwd">

<option ng-repeat="user in users"

value="{{user.pwd}}">{{user.name}}

</option>

</select>

您选择的用户密码为：{{pwd}}

</div>

<script>

var app = angular.module("myapp", []);

app.controller("mycontrol", function($scope) {

$scope.users = [{

name: '张三',

pwd: '111'

}, {

name: '李四',

pwd: '222'

}, {

name: '王五',

pwd: '333'

}];

});

</script>

### ng-options

键值对数组

<div ng-app="myapp" ng-controller="mycontrol">

<select ng-model="class" ng-options="x for (x,y) in classes">

</select>

您选择的class信息为：<br />

方向:{{class.direction}}<br />

岗位：{{class.station}}<br />

等级：{{class.grade}}<br />

</div>

<script>

var app = angular.module("myapp", []);

app.controller("mycontrol", function($scope) {

$scope.classes = {

first: {

direction: "PHP",

station: "架构师",

grade: "高级"

},

second: {

direction: "前端开发",

station: "程序员",

grade: "高级"

},

third: {

direction: "全栈工程师",

station: "系统分析师",

grade: "中级"

}

}

});

</script>

## 表格

### $even和$odd

<tr ng-repeat="x in names">

<td ng-if="$odd" style="background-color:#f1f1f1">

{{ x.Name }}</td>

<td ng-if="$even">

{{ x.Name }}</td>

</tr>

### 实例

<style>

table,th,td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f1f1f1;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

</style>

<div ng-app="myapp" ng-controller="mycontrol">

精确查找：<input type="text" ng-model="select" />

<table>

<tr ng-repeat="fruit in fruits | filter:select | orderBy:'price'">

<td>序列号：{{$index+1}}</td>

<td>水果：{{fruit.name}}</td>

<td>单价：{{fruit.price | currency}}</td>

<td>数量：<input type="text" ng-model="fruit.number" /></td>

<td>总价：{{(fruit.price\*fruit.number) | currency}}</td>

</tr>

</table>

<button ng-click="totalprice()">合计：</button>

<span>{{tprice | currency}}</span>

</div>

<script>

var app = angular.module("myapp", []);

app.controller("mycontrol", function($scope) {

$scope.fruits = [{

name: 'apple',

price: 16,

number: 0

}, {

name: 'orange',

price: 24,

number: 0

}, {

name: 'banana',

price: 14,

number: 0

}];

$scope.totalprice = function() {

$scope.tprice = 0;

for (var i = 0; i < $scope.fruits.length; i++) {

$scope.tprice = $scope.tprice + $scope.fruits[i].price \* $scope.fruits[i].number;

};

}

});

</script>

## 事件

### 单击双击

<div ng-app="myapp" ng-controller="mycontrol">

<button ng-click="oclick()">单击</button>

<button ng-dblclick="dclick()">双击</button>

</div>

<script>

var app=angular.module("myapp",[]);

app.controller("mycontrol",function($scope){

$scope.oclick=function(){

alert("单击一次");

};

$scope.dclick=function(){

alert("双击");

}

});

</script>

### 隐藏显示

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

<p ng-hide="myVar">(将ng-hide改为ng-show则变为显示事件)

姓名:<input type=text ng-model="myName"><br>

级别:<input type=text ng-model="myGrade"><br>

信息: {{myName + " " + myGrade}}

</p>

<button ng-click="toggle()">隐藏/显示</button>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', function($scope) {

$scope.myName = "宁泽涛";

$scope.myGrade = "冠军";

$scope.myVar = false;

$scope.toggle = function() {

$scope.myVar = !$scope.myVar;

}

});

</script>

### 不可用事件

<div ng-app="">

<form>

<textarea cols="10" rows="10" ng-disabled="all"></textarea>

<br />

<input type="text" ng-disabled="all">

<br />

<input type="radio" ng-disabled="all">

<select ng-disabled="all">

<option>web前端开发</option>

<option>PHP开发</option>

<option>H5开发</option>

<option>3Dweb开发</option>

</select>

<br /> 禁用所有表单:<input type="checkbox" ng-model="all">

</form>

</div>