# 5.AngularJS Scope(作用域)

Scope(作用域) 是应用在 HTML (视图) 和 JavaScript (控制器)之间的纽带。

Scope 是一个对象，有可用的方法和属性。

Scope 可应用在视图和控制器上。

## 如何使用 Scope

当你在 AngularJS 创建控制器时，你可以将 **$scope** 对象当作一个参数传递:

控制器中的属性对应了视图上的属性：

[5-1]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <h1>{{carname}}</h1>  
  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope) {  
 $scope.carname = "Volvo";  
 });  
</script>  
  
<p>控制器中创建一个属性名 "carname"，对应了视图中使用 {{ }} 中的名称。</p>  
  
</body>  
</html>

当在控制器中添加 **$scope** 对象时，视图 (HTML) 可以获取了这些属性。

视图中，你不需要添加 **$scope** 前缀, 只需要添加属性名即可，如： **{{carname}}**。

## Scope 概述

AngularJS 应用组成如下：

* View(视图), 即 HTML。
* Model(模型), 当前视图中可用的数据。
* Controller(控制器), 即 JavaScript 函数，可以添加或修改属性。

scope 是模型。

scope 是一个 JavaScript 对象，带有属性和方法，这些属性和方法可以在视图和控制器中使用。

如果你修改了视图，模型和控制器也会相应更新：

[5-2]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
 <input ng-model="name">  
 <h1>{{greeting}}</h1>  
 <button ng-click='sayHello()'>点我</button>  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
 ***app***.controller('myCtrl', function($scope) {  
 $scope.name = "Runoob";  
 $scope.sayHello = function() {  
 $scope.greeting = 'Hello ' + $scope.name + '!';  
 };  
 });  
</script>  
  
<p>当你修改输入框中的值时，会影响到模型(model),当然也会影响到控制器对应的属性值。</p>  
  
</body>  
</html>

## Scope 作用范围

了解你当前使用的 scope 是非常重要的。

在以上两个实例中，只有一个作用域 scope，所以处理起来比较简单，但在大型项目中， HTML DOM 中有多个作用域，这时你就需要知道你使用的 scope 对应的作用域是哪一个。

当我们使用 ng-repeat 指令时，每个重复项都访问了当前的重复对象:

[5-3]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <ul>  
 <li ng-repeat="x in names">{{***x***}}</li>  
 </ul>  
  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
  
 ***app***.controller('myCtrl', function($scope) {  
 $scope.names = ["Emil", "Tobias", "Linus"];  
 });  
</script>  
  
</body>  
</html>

每个 **<li>** 元素可以访问当前的重复对象，这里对应的是一个字符串， 并使用变量 **x** 表示。

## 根作用域

所有的应用都有一个 **$rootScope**，它可以作用在 **ng-app** 指令包含的所有 HTML 元素中。

**$rootScope** 可作用于整个应用中。是各个 controller 中 scope 的桥梁。用 rootscope 定义的值，可以在各个 controller 中使用。

创建控制器时，将 $rootScope 作为参数传递，可在应用中使用：

[5-5]

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <script src="js/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
  
 <h1>姓氏为 {{lastname}} 家族成员:</h1>  
  
 <ul>  
 <li ng-repeat="x in names">{{***x***}} {{lastname}}</li>  
 </ul>  
  
</div>  
  
<script>  
 var ***app*** = angular.module('myApp', []);  
  
 ***app***.controller('myCtrl', function($scope, $rootScope) {  
 $scope.names = ["Emil", "Tobias", "Linus"];  
 $rootScope.lastname = "Refsnes";  
 });  
</script>  
  
<p>注意 $rootScope 在循环对象内外都可以访问。</p>  
  
</body>  
</html>