**Angularjs基础学习**

1. **表达式：**
   1. 类似于 JavaScript 表达式，AngularJS 表达式可以包含字母，操作符，变量。
   2. 与 JavaScript 表达式不同，AngularJS 表达式可以写在 HTML 中。
   3. 与 JavaScript 表达式不同，AngularJS 表达式不支持条件判断，循环及异常。
   4. 与 JavaScript 表达式不同，AngularJS 表达式支持过滤器。
2. **指令**
   1. AngularJS 通过被称为 指令 的新属性来扩展 HTML。
   2. AngularJS 通过内置的指令来为应用添加功能。
   3. AngularJS 允许你自定义指令。
   4. 除了 AngularJS 内置的指令外，我们还可以创建自定义指令。
   5. 你可以使用 .directive 函数来添加自定义的指令。
   6. 要调用自定义指令，HTML 元素上需要添加自定义指令名。
   7. 使用驼峰法来命名一个指令， runoobDirective, 但在使用它时需要以 - 分割, runoob-directive:
   8. 可以通过{元素名，属性，类名，注释}来调用指令。
   9. 限制使用(列子)：

通过添加 restrict 属性,并设置值为 "A", 来设置指令只能通过属性的方式来调用:

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

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

return {

restrict : "A",

template : "<h1>自定义指令!</h1>"

};

});

restrict 值可以是以下几种:

E 作为元素名使用

A 作为属性使用

C 作为类名使用

M 作为注释使用

restrict 默认值为 EA, 即可以通过元素名和属性名来调用指令。

**三．作用域**

scope: false, // 默认值，共享父级作用域

scope: true, // 继承父级作用域并创建指令自己的作用域

scope: {}, // 创建指令自己的独立作用域，与父级毫无关系

**四．控制器**

①{{fullName()}}：双括号添加函数

$scope.fullName = function() {  
        return $scope.firstName + " " + $scope.lastName;  
    }

②外部控制器：<script src="personController.js"></script>

五．过滤器

①<input type="text" ng-model="test">

| filter:test ，相当于模糊查询输入值

②自定义过滤器(app.filter()方法)

app.filter('reverse', function() { //可以注入依赖 return function(text) { return text.split("").reverse().join(""); } });

③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

④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

如：{{“jj” | myfilter:1:2:3:5}}

**六．服务**

①angularjs内置服务(30多种)

$location，$http，$timeout，$interval

②自定义服务：

app.service('hexafy', function() {  
    this.myFunc = function (x) {  
        return x.toString(16);  
    }  
});

使用：

app.controller('myCtrl', function($scope, **hexafy**) {  
    $scope.hex = **hexafy**.myFunc(255);  
});、

过滤器使用服务：

app.filter('myFormat',['hexafy', function(hexafy) {

return function(x) {

return hexafy.myFunc(x);

};

}]);

③函数监听/$watch监听 （<h2>{{ getFullName() }}</h2>）

$scope.getFullName =function(){ return $scope.lastName + " " + $scope.firstName;}

//监听lastName的变化，更新fullName

$scope.$watch('lastName', function() {

$scope.fullName = $scope.lastName + " " + $scope.firstName;

});

七．**Select**

①ng-repeat与ng-option

两个都可以创建下拉菜单，但是ng-repeat选择时，选择的是字符串，ng-option是对象。

<select ng-init="selectedName = names[0]" ng-model="selectedName" ng-options="x for x in names"> </select>

-------------------------------------------------------------------

<select>  
<option ng-repeat="x in names">{{x}}</option>  
</select>

**八．模块**

①在模块定义中[]参数用于定义模块得依赖关系。[模块名称]

[]表示该模块没有依赖，如果有依赖得话会在中括号上写上依赖的模块名字。

**九．表单**

①angularjs表单是输入控件的集合。

Input控件使用ng-model指令来实现数据绑定。

通过以上代码应用有了一个名为firstname属性。

它通过ng-model指令来绑定到你的应用。

这个属性firstname可以在controller中使用。

②ng-switch

单选框ng-model为同一个值，value不同，但是只有被选中的值被使用。

<div ng-switch="myVar">

<div ng-switch-when="dogs">

<h1>Dogs</h1>

<p>Welcome to a world of dogs.</p>

</div>

<div ng-switch-when="tuts">

<h1>Tutorials</h1>

<p>Learn from examples.</p>

</div>

<div ng-switch-when="cars">

<h1>Cars</h1>

<p>Read about cars.</p>

</div>

</div>

**十．输入验证**

①AngularJS 表单和控件可以验证输入的数据。

（客户端的验证不能确保用户输入数据的安全，所以服务端的数据验证也是必须的）

**十一.angularjs，bootstrap**

①angularjs的首选样式表是Twitter Bootsrap,Twitter Bootsrap是目前最受欢迎的前端框架。

②sass或者less

首先sass和less都是css的预编译处理语言，他们引入了mixins，参数，嵌套规则，运算，颜色，名字空间，作用域，JavaScript赋值等 加快了css开发效率,当然这两者都可以配合gulp和grunt等前端构建工具使用

　sass和less主要区别:在于实现方式 less是基于JavaScript的在客户端处理 所以安装的时候用npm，sass是基于ruby所以在服务器处理

③mixin(类似于函数，自定义样式，被调用)

less，sass这类css预处理语言中的混合，可以理解成自定义了一段代码，后面可以用@include调用，如设置一个有默认宽高的mixin（不带参数的类似）

④Responsive Design

**十二.依赖注入**

1. AngularJS 提供很好的依赖注入机制。以下5个核心组件用来作为依赖注入：

Value, factory, service, provider, constant

1.Value 是一个简单的 javascript 对象，用于向控制器传递值（配置阶段）：

Provider详解：

Angularjs中通过provider创建一个service，factory等(配置阶段)

2.Provider中提供了一个factory方法get()，它用于返回value/service/factory

3. factory 是一个函数用于返回值。在 service 和 controller 需要时创建

$scope.showAccInfo = **function** (item) {  
 $scope.**accInfo** = item;  
 **var** confirmPopup = $ionicPopup.alert({  
 **title**: **'<strong>台账详情</strong>'**,  
 **templateUrl**: **'labor/labor-dispatch-work-template.html'  
 scope**: $scope,  
 });  
 confirmPopup.then(**function** (res) {  
  
 });  
}

\*\*\*\*\*\*\*对于scope:$scope来说将变量赋值给当前会话。

**十三.路由**

AngularJS 路由允许我们通过不同的 URL 访问不同的内容。

通过 AngularJS 可以实现多视图的单页Web应用（single page web application，SPA）。

通常我们的URL形式为 **http://runoob.com/first/page**，但在单页Web应用中 AngularJS 通过 **# + 标记** 实现。