

1. AngularJS 简介

AngularJS 是一个 JavaScript 框架。它可通过 `<script>` 标签添加到 HTML 页面。

AngularJS 通过 指令 扩展了 HTML，且通过 表达式 绑定数据到 HTML

AngularJS 使得开发现代的单一页面应用程序变得更加容易。

- AngularJS 把应用程序数据绑定到 HTML 元素。
- AngularJS 可以克隆和重复 HTML 元素。
- AngularJS 可以隐藏和显示 HTML 元素。
- AngularJS 可以在 HTML 元素"背后"添加代码。
- AngularJS 支持输入验证。

2. Angular JS 表达式

```
<div ng-app="" ng-init="firstName='Sean'">
  <p>我的第一个表达式: {{ 7 + 8 }}</p>
  <p>姓名为 <span ng-bind="firstName"></span></p>
</div>
```

我的第一个表达式: 15

姓名为 Sean

AngularJS 表达式写在双大括号内: `{{ expression }}` ,同时也可以使用 `ng-bind` 绑定
`ng-app` 指令定义一个 AngularJS 应用程序

3. Angular JS 指令

```
<div ng-app="" ng-init="firstName='John'">
  <p>在输入框中尝试输入: </p>
  <p>姓名: <input type="text" ng-model="firstName"></p>
  <p>你输入的为: {{ firstName }}</p>
```

在输入框中尝试输入:

姓名:

你输入的为: Sean

`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="" ng-init="names=[
{name:'Sean',country:'America'},
{name:'fiona',country:'Canada'},
{name:'Catherine',country:'London'}]">
  <p>循环对象: </p>
  <ul>
    <li ng-repeat="x in names">
      {{ x.name + ', ' + x.country }}
    </li>
  </ul>

```

循环对象:

- Sean, America
- fiona, Canada
- Catherine, London

ng-repeat 可以重复一个html元素

4.AngularJS 作用域

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, 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。

```

<div ng-app="myApp" ng-controller="myCtrl">
  <h1>{{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 = ["Sean", "Cici", "Emma"];
    $rootScope.lastname = "Ji";
  });
</script>

```

Ji 家族成员:

- Sean Ji
- Cici Ji
- Emma Ji

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。

5. AngularJS 控制器

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

```
  名: <input type="text" ng-model="firstName"><br>
  姓: <input type="text" ng-model="lastName"><br>
  <br>
  姓名: {{fullName()}}
```

名:

姓:

```
</div>
```

姓名: Sean Ji

```
<script>
```

```
  var app = angular.module('myApp', []);
  app.controller('personCtrl', function($scope) {
    $scope.firstName = "Sean";
    $scope.lastName = "Ji";
    $scope.fullName = function() {
      return $scope.firstName + " " + $scope.lastName;
    }
  });
```

```
</script>
```

AngularJS 应用程序由 ng-app 定义。应用程序在 <div> 内运行。

ng-controller="myCtrl" 属性是一个 AngularJS 指令。用于定义一个控制器。

myCtrl 函数是一个 JavaScript 函数。

AngularJS 使用 \$scope 对象来调用控制器。

在 AngularJS 中，\$scope 是一个应用象(属于应用变量和函数)。

控制器的 \$scope （相当于作用域、控制范围）用来保存 AngularJS Model(模型)的对象。

控制器在作用域中创建了两个属性 (firstName 和 lastName)。

ng-model 指令绑定输入域到控制器的属性 (firstName 和 lastName) 。

6. AngularJS 服务

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

现在时间是:

```
<p>现在时间是:</p>
```

9:20:49 PM

```
<h1>{{theTime}}</h1>
```

```
</div>
```

\$interval 访问在指定的周期(以毫秒计)来调用函数或计算表达式。

```
<p>$interval 访问在指定的周期(以毫秒计)来调用函数或计算表达式。</p>
```

```
<script>
```

```
  var app = angular.module('myApp', []);
  app.controller('myCtrl', function($scope, $interval) {
    $scope.theTime = new Date().toLocaleTimeString();
    $interval(function () {
      $scope.theTime = new Date().toLocaleTimeString();
    }, 1000);
  });
```

```
</script>
```

AngularJS \$interval 服务对应了 JS window.setInterval 函数。

```
<div ng-app="myApp" ng-controller="myCtrl">
  <p>255 的16进制是:</p>
  <h1>{{hex}}</h1>
</div>
```

255 的16进制是:

ff

<p>自定义服务，用于转换16进制数: </p>

自定义服务，用于转换16进制数:

```
<script>
  var app = angular.module('myApp', []);

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

  app.controller('myCtrl', function($scope, hexafy) {
    $scope.hex = hexafy.myFunc(255);
  });
</script>
```

“hexafy”即为自定义服务，此处为16进制转换。

7.AngularJS Http

```
<div ng-app="myApp" ng-controller="customersCtrl">
  <ul>
    <li ng-repeat="x in names">
      {{ x.Name + ', ' + x.Country }}
    </li>
  </ul>
</div>

<script>
  var app = angular.module('myApp', []);
  app.controller('customersCtrl', function($scope, $http) {
    $http.get("Customers_JSON")
      .success(function (response) {$scope.names = response.records;});
  });
</script>
```

- Alfreds Futterkiste, Germany
- Ana Trujillo Emparedados y helados, Mexico
- Antonio Moreno Taquería, Mexico
- Around the Horn, UK
- B's Beverages, UK
- Berglunds snabbköp, Sweden
- Blauer See Delikatessen, Germany
- Blondel père et fils, France
- Bólido Comidas preparadas, Spain
- Bon app', France
- Bottom-Dollar Marketse, Canada

\$http 调取后台json数据

8. AngularJS 事件

```
<div ng-app="myApp" ng-controller="personCtrl">
  <button ng-click="toggle()">>隐藏/显示</button>
  <p ng-hide="myVar">
    名: <input type="text" ng-model="firstName"><br>
    姓: <input type="text" ng-model="lastName"><br>
    Full Name: {{firstName + " " + lastName}}
  </p>
</div>

<script>
  var app = angular.module('myApp', []);
  app.controller('personCtrl', function($scope) {
    $scope.firstName = "Sean",
    $scope.lastName = "Ji"
    $scope.myVar = false;
    $scope.toggle = function() {
      $scope.myVar = !$scope.myVar;
    };
  });
</script>
```

>隐藏/显示

名: Sean

姓: Ji

Full Name: Sean Ji

点击“隐藏/显示”按钮，按钮下的姓名会隐藏或者显示
ng-hide 具有隐藏/显示功能，当它的属性值为false时，为显示，反之则为隐藏

9. AngularJS 输入验证

```
<p>邮箱:<br>
  <input type="email" name="email" ng-model="email" required>
  <span style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">
  <span ng-show="myForm.email.$error.required">邮箱是必须的。</span>
  <span ng-show="myForm.email.$error.email">非法的邮箱。</span>
</span>
</p>
```

用户名:

用户名是必须的。

邮箱:

597100700 非法的邮箱。

Submit

ng-show 验证属性是否为真，属性为真则显示，反之则不显示
\$error.required 验证是否存在
\$error.email 验证是否为邮箱

10.AngularJS 综合应用

```
<body ng-app="myApp" ng-controller="todoCtrl">

<h2>我的备忘录</h2>

<form ng-submit="todoAdd()">
  <input type="text" ng-model="todoInput" size="50" placeholder="新增">
  <input type="submit" value="新增">
</form>

<br>

<div ng-repeat="x in todoList">
  <input type="checkbox" ng-model="x.done"> <span ng-bind="x.todoText"></span>
</div>

<p><button ng-click="remove()">删除记录</button></p>

<script>
  var app = angular.module('myApp', []);
  app.controller('todoCtrl', function($scope) {
    $scope.todoList = [{todoText:'Clean House', done:false}];

    $scope.todoAdd = function() {
      $scope.todoList.push({todoText:$scope.todoInput, done:false});
      $scope.todoInput = "";
    };

    $scope.remove = function() {
      var oldList = $scope.todoList;
      $scope.todoList = [];
      angular.forEach(oldList, function(x) {
        if (!x.done) $scope.todoList.push(x);
      });
    };
  });
</script>
```

我的备忘录

☐ Sean

☐ Fiona

点击新增增加备忘录，选中备忘录点击删除进行删除
todoList 存放备忘录，标记为显示 (done=false)
todoAdd 和 remove 函数分别用来添加和删除

Node.js 安装配置

Node.js安装包及源码下载地址为: <https://nodejs.org/en/download/>

```
var http = require('http');

http.createServer(function (request, response) {

    // 发送 HTTP 头部
    // HTTP 状态值: 200 : OK
    // 内容类型: text/plain
    response.writeHead(200, {'Content-Type': 'text/plain'});

    // 发送响应数据 "Hello World"
    response.end('Hello World\n');
}).listen(8888);

// 终端打印如下信息
console.log('Server running at http://127.0.0.1:8888/');
```

```
<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
    $http.get("http://www.runoob.com/try/angularjs/data/Customers_JSON.php")
        .success(function(response) {$scope.names = response.records;});
});
</script>
```

分析Node.js 的 HTTP 服务器:

- 第一行请求 (require) Node.js 自带的 http 模块, 并且把它赋值给 http 变量。
- 接下来我们调用 http 模块提供的函数: createServer。这个函数会返回 一个对象, 这个对象有一个叫做 listen 的方法, 这个方法有一个数值参数, 指定这个 HTTP 服务器监听的端口号。

AngularJS 通过\$http.get()方法与NodeJS进行交互

总结比较（AngularJS vs JQuery）：

- jQuery在DOM上做得很好，可以根据用户交互，添加修改DOM元素。而AngularJS更关注数据展示本身，更新时会尽可能减少对DOM的破坏和重构。当然，整个操作不可能完全的避免DOM，但使用AngularJS就比jQuery对于DOM的工作少多了。
- AngularJS，尽管它的学习曲线没有那么平缓，但它强调开发效率，AngularJS中很多特点的设计都是出于提高开发者效率的目的。比如在框架内它就支持单元测试，这样的设计可以让程序员更集中精力在写出可靠的代码上，简化了测试上的麻烦。另外，降低模块间的耦合度也有利于程序员梳理项目逻辑，可以更专注于“为什么”，而不是“怎么做”。这样带来的好处就是代码会更简洁和易读，有利于调试和维护。
- jQuery的一个常见问题就是它是命令式编程，就意味着你要告诉计算机如何达成某项目的，其实你想要的就是程序运行的结果。而声明式编程则转移了重点，它只是告诉机器你想要的结果，让机器自己实现这个结果。JavaScript就是一种命令式语言，但HTML，和它指导的AngularJS就是声明式的，只需要告诉计算机你需要的展现形式，至于细节就由机器处理了。这样，完成一个任务的代码就大幅缩减了，代码质量也更高了。

所以，AngularJS和jQuery之间的比较就如同苹果和橘子的比较，两者是出于不同的目的被创建的，解决的也是不同的问题。当一个项目的重点是数据展示和执行，而不是分析，此时可能AngularJS就会更胜一筹。对于框架的选择，你要考虑到很多因素，需要整体的构思。AngularJS总体上还是一个不错的选择，可以提高程序员的效率，相对减少Web开发中的维护成本。