# 15.AngularJS 模块

模块定义了一个应用程序。

模块是应用程序中不同部分的容器。

模块是应用控制器的容器。

控制器通常属于一个模块。

## 创建模块

你可以通过 AngularJS 的 **angular.module** 函数来创建模块：

<div ng-app="myApp">...</div>  
  
<script>

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

</script>

"myApp" 参数对应执行应用的 HTML 元素。

现在你可以在 AngularJS 应用中添加控制器，指令，过滤器等。

## 添加控制器

你可以使用 **ng-controller** 指令来添加应用的控制器:

## AngularJS 实例

<div ng-app="**myApp**" ng-controller=**"myCtrl"**>  
{{ firstName + " " + lastName }}  
</div>  
  
<script>

var app = angular.module(**"myApp"**, []);  
  
app.controller(**"myCtrl"**, function($scope) {  
    $scope.firstName = "John";  
    $scope.lastName = "Doe";  
});

</script>

**[尝试一下 »](https://www.runoob.com/try/try.php?filename=try_ng_module" \t "https://www.runoob.com/angularjs/_blank)**

你可以在 [AngularJS 控制器](https://www.runoob.com/angularjs/angularjs-controllers.html" \o "AngularJS 控制器" \t "https://www.runoob.com/angularjs/_top)章节学到更多关于控制器的知识。

## 添加指令

AngularJS 提供了很多内置的指令，你可以使用它们来为你的应用添加功能。

完整的指令内容可以参阅 [AngularJS 参考手册](https://www.runoob.com/angularjs/angularjs-reference.html)。

此外，你可以使用模块来为你应用添加自己的指令：

## AngularJS 实例

<div ng-app="myApp" runoob-directive></div>  
  
<script>

var app = angular.module("myApp", []);  
  
app.directive("runoobDirective", function() {  
    return {  
        template : "我在指令构造器中创建!"  
    };  
});

</script>

**[尝试一下 »](https://www.runoob.com/try/try.php?filename=try_ng_directive" \t "https://www.runoob.com/angularjs/_blank)**

你可以在 [AngularJS 指令](https://www.runoob.com/angularjs/angularjs-directives.html" \o "AngularJS 指令" \t "https://www.runoob.com/angularjs/_top)章节学到更多关于指令的知识。

## 模块和控制器包含在 JS 文件中

通常 AngularJS 应用程序将模块和控制器包含在 JavaScript 文件中。

在以下实例中， "myApp.js" 包含了应用模块的定义程序， "myCtrl.js" 文件包含了控制器：

## AngularJS 实例

<!DOCTYPE html>  
<html>  
<script src="http://apps.bdimg.com/libs/angular.js/1.4.6/angular.min.js"></script>  
<body>  
  
<div ng-app="**myApp**" ng-controller="**myCtrl**">  
{{ firstName + " " + lastName }}  
</div>  
  
<script src="**myApp.js**"></script>  
<script src="**myCtrl.js**"></script>  
  
</body>  
</html>

**[尝试一下 »](https://www.runoob.com/try/tryit.php?filename=try_ng_module_files" \t "https://www.runoob.com/angularjs/_blank)**

## myApp.js

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

|  |  |
| --- | --- |
| **IMG_256** | 在模块定义中 [] 参数用于定义模块的依赖关系。 中括号[]表示该模块没有依赖，如果有依赖的话会在中括号写上依赖的模块名字。 |

## myCtrl.js

app.controller(**"myCtrl"**, function($scope) {  
    $scope.firstName = "John";  
    $scope.lastName= "Doe";  
});

## 函数会影响到全局命名空间

JavaScript 中应避免使用全局函数。因为他们很容易被其他脚本文件覆盖。

AngularJS 模块让所有函数的作用域在该模块下，避免了该问题。

## 什么时候载入库?

|  |  |
| --- | --- |
| **IMG_257** | 在我们的实例中，所有 AngularJS 库都在 HTML 文档的头部载入。 |

对于 HTML 应用程序，通常建议把所有的脚本都放置在 <body> 元素的最底部。

这会提高网页加载速度，因为 HTML 加载不受制于脚本加载。

在我们的多个 AngularJS 实例中，您将看到 AngularJS 库是在文档的 <head> 区域被加载。

在我们的实例中，AngularJS 在 <head> 元素中被加载，因为对 angular.module 的调用只能在库加载完成后才能进行。

另一个解决方案是在 <body> 元素中加载 AngularJS 库，但是必须放置在您的 AngularJS 脚本前面：

## AngularJS 实例

<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
<script src="http://apps.bdimg.com/libs/angular.js/1.4.6/angular.min.js"></script>  
</head>  
<body>  
  
<div ng-app="myApp" ng-controller="myCtrl">  
{{ firstName + " " + lastName }}  
</div>  
<script>  
var app = angular.module("myApp", []);  
app.controller("myCtrl", function($scope) {  
    $scope.firstName = "John";  
    $scope.lastName = "Doe";  
});  
</script>  
  
</body>  
</html>

**[尝试一下 »](https://www.runoob.com/try/tryit.php?filename=try_ng_module_body" \t "https://www.runoob.com/angularjs/_blank)**