

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 是一个 JavaScript 框架

AngularJS 是一个 JavaScript 框架。它是一个以 JavaScript 编写的库。

AngularJS 是以一个 JavaScript 文件形式发布的，可通过 `script` 标签添加到网页中：

```
<script src="https://cdn.staticfile.org/angular.js/1.4.6/angular.min.js"></script>
```



我们建议把脚本放在 `<body>` 元素的底部。
这会提高网页加载速度，因为 HTML 加载不受制于脚本加载。

各个 angular.js 版本下载：<https://github.com/angular/angular.js/releases>

AngularJS 扩展了 HTML

AngularJS 通过 **ng-directives** 扩展了 HTML。

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。

ng-bind 指令把应用程序数据绑定到 HTML 视图。

AngularJS 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script src="https://cdn.staticfile.org/angular.js/1.4.6/angular.min.js"></script>
</head>
<body>
<div ng-app="">
<p>名字：<input type="text" ng-model="name"></p>
<h1>Hello {{name}}</h1>
</div>
</body>
</html>
```

尝试一下 »

实例讲解：

当网页加载完毕，AngularJS 自动开启。

ng-app 指令告诉 AngularJS，`<div>` 元素是 AngularJS **应用程序** 的"所有者"。

ng-model 指令把输入域的值绑定到应用程序变量 **name**。

ng-bind 指令把应用程序变量 **name** 绑定到某个段落的 **innerHTML**。



如果您移除了 **ng-app** 指令，HTML 将直接把表达式显示出来，不会去计算表达式的结果。

什么是 AngularJS ?

AngularJS 使得开发现代的单一页面应用程序（SPAs：Single Page Applications）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。

AngularJS 指令

正如您所看到的，AngularJS 指令是以 **ng** 作为前缀的 HTML 属性。

ng-init 指令初始化 AngularJS 应用程序变量。

AngularJS 实例

```
<div ng-app="" ng-init="firstName='John'">
<p>姓名为 <span ng-bind="firstName"></span></p>
</div>
```

尝试一下 »



HTML5 允许扩展的（自制的）属性，以 **data-** 开头。
AngularJS 属性以 **ng-** 开头，但是您可以使用 **data-ng-** 来让网页对 HTML5 有效。

带有有效的 HTML5：

AngularJS 实例

```
<div data-ng-app="" data-ng-init="firstName='John'">
<p>姓名为 <span data-ng-bind="firstName"></span></p>
</div>
```

尝试一下 »

AngularJS 表达式

AngularJS 表达式写在双大括号内：**{{ expression }}**。

AngularJS 表达式把数据绑定到 HTML，这与 **ng-bind** 指令有异曲同工之妙。

AngularJS 将在表达式书写的位置"输出"数据。

AngularJS 表达式 很像 **JavaScript 表达式**：它们可以包含文字、运算符和变量。

实例 {{ 5 + 5 }} 或 {{ firstName + " " + lastName }}

AngularJS 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<script src="https://cdn.staticfile.org/angular.js/1.4.6/angular.min.js"></script>
</head>
<body>
<div ng-app="">
<p>我的第一个表达式:  {{ 5 + 5 }}</p>
</div>
</body>
</html>
```

尝试一下 »

AngularJS 应用

AngularJS **模块 (Module)** 定义了 AngularJS 应用。

AngularJS **控制器 (Controller)** 用于控制 AngularJS 应用。

ng-app指令指明了应用, **ng-controller** 指明了控制器。

AngularJS 实例

```
<div ng-app="myApp" ng-controller="myCtrl">
名: <input type="text" ng-model="firstName"><br>
姓: <input type="text" ng-model="lastName"><br>
<br>
姓名: {{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 »

AngularJS 模块定义应用:

AngularJS 模块

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

AngularJS 控制器控制应用:

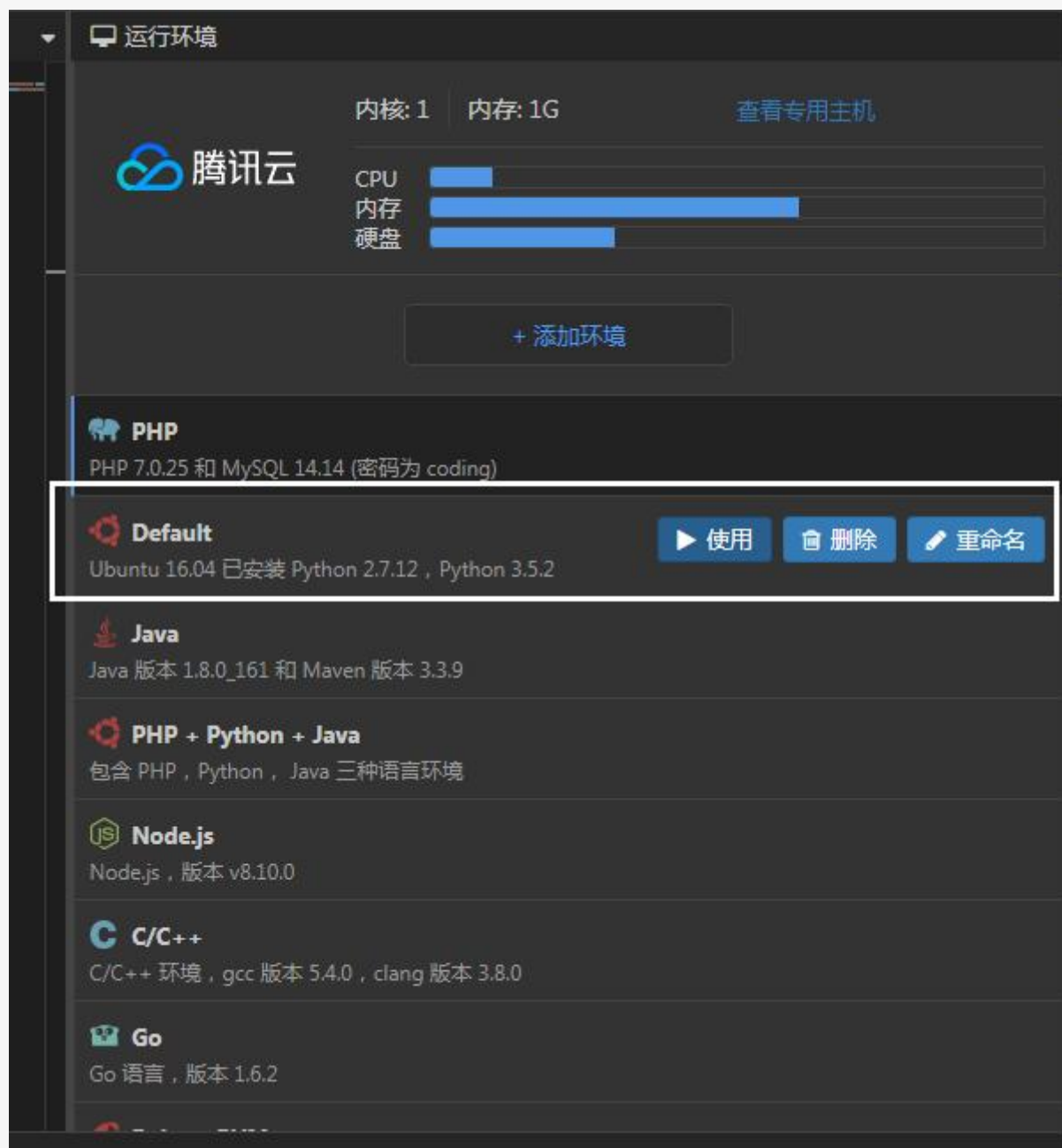
AngularJS 控制器

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

在接下来的教程中你将学习到更多的应用和模块的知识。

在 Cloud Studio 中使用 AngularJS

- step1 : 访问 [腾讯云开发者平台](#)，注册/登录账户。
- step2 : 在右侧的运行环境菜单选择："ubuntu"



- step3 : 在左侧代码目录中新建 html 目录，编写你的 HTML 代码，例如 index.html

- step4 : 在 index.html 文件的中粘贴如下代码、来引入 AngularJS :

```
<script src="https://cdn.staticfile.org/angular.js/1.4.6/angular.min.js"></script>
```

- step5 : 在终端中输入命令 `sudo vim /etc/nginx/site-enabled/default`。将配置文件红框部分修改为如下图所示, 然后输入命令: `sudo nginx restart` 重启 nginx 服务 (nginx 安装完成并启动后默认会监听80端口。我们需要将 nginx 的站点目录以及监听的端口号改为我们需要的)

```
server {  
    listen 8080 default_server;  
    listen [::]:8080 default_server;  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    # listen [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
    #  
    # Read up on ssl_ciphers to ensure a secure configuration.  
    # See: https://bugs.debian.org/765782  
    #  
    # Self signed certs generated by the ssl-cert package  
    # Don't use them in a production server!  
    #  
    # include snippets/snakeoil.conf;  
  
    root /home/coding/workspace/html;  
  
    # Add index.php to the list if you are using PHP  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name _;  
  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        try_files $uri $uri/ =404;  
    }  
}
```

- step6 : 点击最右侧的【访问链接】选项卡, 在访问链接面板中填写端口号为: 8080 (和刚才 nginx 配置文件中的端口号一致), 点击创建链接, 即可点击生成的链接访问我们刚刚编写的代码, 查看 AngularJS 效果。



现在 CODING 正在举办一场基于 Cloud Studio 工作空间的【我最喜爱的 Cloud Studio 插件评选大赛】。进入活动官网：<https://studio.qcloud.coding.net/campaign/favorite-plugins/index>，了解更多活动信息。

[← AngularJS 教程](#)[AngularJS 表达式 →](#)[✎ 点我分享笔记](#)