

به نام خدا

برنامه نویسی مبتنی بر وب



AngularJS

منابع



<https://angularjs.org/> •

[w3schools.com](https://www.w3schools.com)

<https://www.w3schools.com> •



<http://www.tutorialsteacher.com> •

AngularJS چیست؟

- AngularJS یک زبانی برای ساخت اپلیکیشن های تک صفحه ای (Single Page Application) می باشد. Angular یک فریم ورک جاوااسکریپت است و بر مبنای مدل MVC می باشد. این فریمورک توسط گوگل عرضه شده است.

<https://angularjs.org>

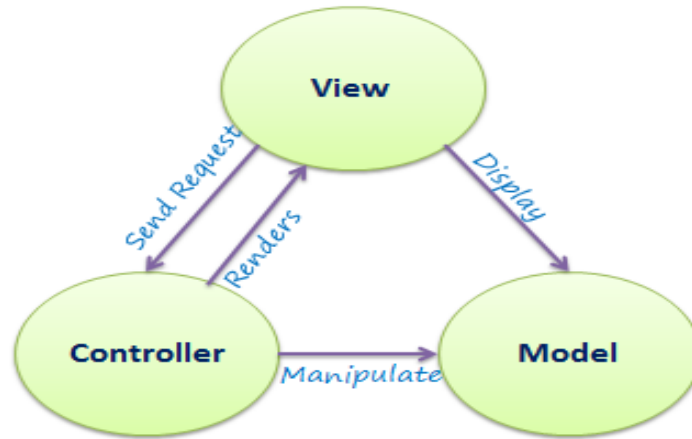


What is AngularJS?

- AngularJS is a **client side JavaScript MVC** framework to develop a dynamic web application. AngularJS was **originally started as a project in Google** but now, it is **open source framework**.

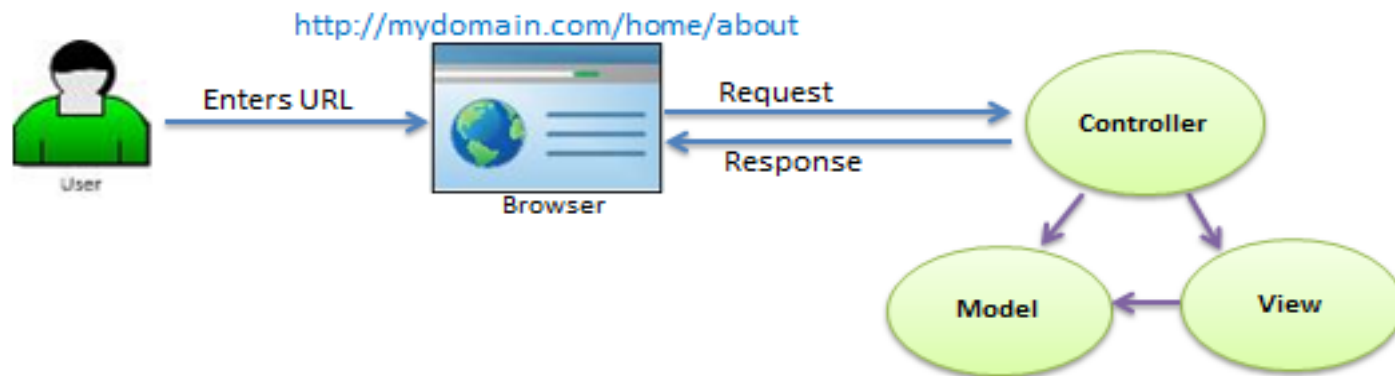
معماری MVC

- **Model** is a data and business logic.
- **View** is a User Interface.
- **Controller** is a request handler.



MVC Architecture

معماری MVC



Request/Response in MVC Architecture

نحوه استفاده از Angular

- برای استفاده از Angular کافی است این کتابخانه را دانلود کنید و با تگ `<script>` به فایل HTML اضافه کرد.
- `<script src="~/Scripts/angular.js"></script>`

```
<!DOCTYPE html>

<html>
<head>
    <script src="~/Scripts/angular.js"></script>
</head>
```

نحوه استفاده از Angular

- همچنین می توان را از CDN Angular های موجود به فایل HTML اضافه نمود.

```
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">  
</script>
```


نمونه کد

The diagram illustrates the components of an AngularJS application. Three blue ovals with labels 'Template', 'Directives', and 'Expression' are connected to specific parts of the HTML code by dotted arrows. 'Template' points to the opening <!DOCTYPE html> tag. 'Directives' points to the ng-app attribute on the <body> tag, the ng-model attribute on the first input field, and the {{ }} expression syntax within the span. 'Expression' points to the {{Num1 * Num2}} expression inside the span.

```
<!DOCTYPE html>
<html>
<head>
  <title>First AngularJS Application</title>
  <script src="/Scripts/angular.js"></script>
</head>
<body ng-app>
  <h1>First AngularJS Application</h1>
  Enter Numbers to Multiply:
  <input type="text" ng-model="Num1" /> x <input type="text" ng-model="Num2" />
  = <span>{{Num1 * Num2}}</span>
</body>
</html>
```

First AngularJS Application

First AngularJS Application

Enter Numbers to Multiply: x = 60

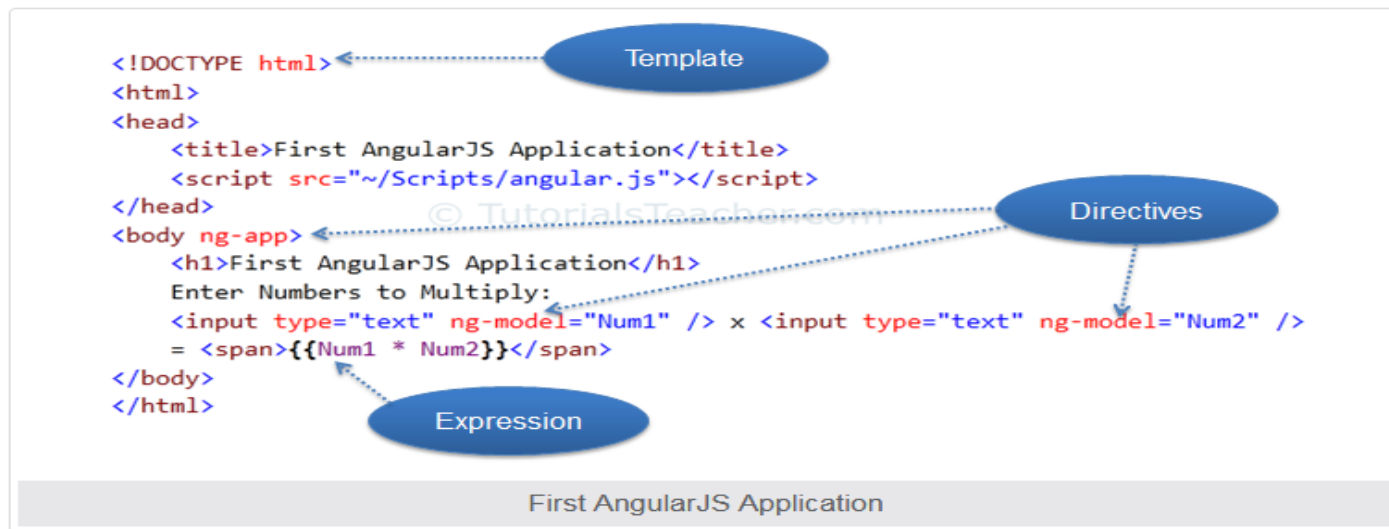
First AngularJS Application

مفاهیم مهم در Angular

Concept	Description
Template	HTML with additional markup
Directives	Extends the HTML with custom attributes and elements
Model	The data shown to the user in the view and with which the user interacts
Scope	A context where the model is stored so that controllers, directives and expressions can access it
Expressions	Executes JavaScript code inside brackets <code>{{ }}</code> .
Compiler	Parses the template and instantiates directives and expressions
Filter	Formats the value of an expression for display to the user
View	what the user sees (the DOM)
Data Binding	Sync data between the model and the view
Controller	Maintains the application data and business logic
Module	a container for different parts of an app including controllers, services, filters, directives which configure the Injector
Service	Reusable business logic, independent of views
Dependency Injection	Creates and wires objects and functions
Injector	Dependency injection container

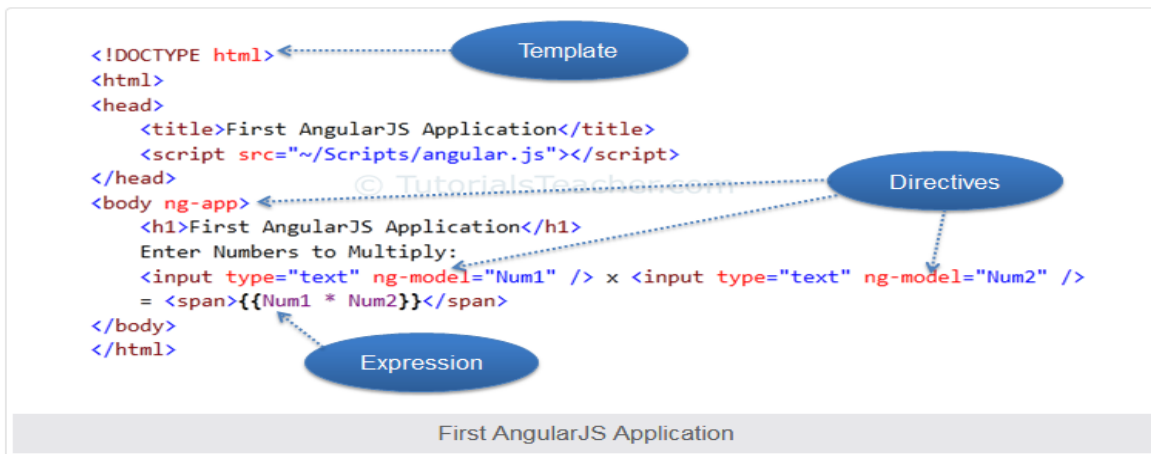
Template

- template سند html می باشد که خصوصیت های ویژه ای به آن اضافه شده است.



Directive

- **Directive** ها خصوصیت هایی (attributes یا markers) هستند که به **Angular** JS می گویند تا یک رفتار خاصی را به عنصر html مورد نظر اضافه نماید. اکثر Directive با پیشوند **ng** شروع می شوند که مخفف angular است.



code 1.txt

Directive

- دستور `ng-app` یک اپلیکیشن انگولار را مقدار دهی اولیه می کند.
- دستور `ng-init` داده های اپلیکیشن را مقدار دهی اولیه می کند.
- دستور `ng-model` مقدار کنترل های `html` از قبیل `input` و `select` و `textarea` را به داده های اپلیکیشن پیوند می دهد.

The `ng-app` directive initializes an AngularJS application.

The `ng-init` directive initializes application data.

The `ng-model` directive binds the value of HTML controls (input, select, textarea) to application data.

ng-app

- دستور **ng-app** یک اپلیکیشن انگولار را مشخص می کند. به طور معمول ng-app باید در ریشه یک سند html همانند تگ `<html>` یا `<body>` قرار بگیرد تا کنترل کل سند DOM را داشته باشد. هر چند که در هر عنصر دیگر DOM نیز می تواند قرار گیرد.

Note that multiple ng-app directives are **NOT** allowed in a single HTML document.

```
<!DOCTYPE html>
<html>
<head>
  <title>Angular Bootstrap</title>
  <script src="~/Scripts/angular.js"></script>
</head>
<body>
  <div>
    {{2/2}}
  </div>
  <div ng-app id="myDiv">
    {{5/2}}
    <div>
      {{10/2}}
    </div>
  </div>
  <div>
    {{2/2}}
  </div>
</body>
</html>
```

© TutorialsTeacher.com

Angular
features not
supported out
of ng-app

Angular
features
supported only
inside ng-app



Code 2.txt

ng-app همراه با نام ماژول

- در دایرکتیو ng-app همچنین می توانید نام یک ماژول را تعیین کنید.

ng-app with Module name:

The ng-app directive can also specify an application **module** name. This application module separates different parts of your application such as controllers, services, filters etc.

Example: ng-app with App Module

```
<!DOCTYPE html>
<html>
<head>
  <title>ng-app Directive</title>
  <script src="/Scripts/angular.js"></script>
</head>
<body ng-app="myAngularApp">
  <div>
    {{2/2}}
  </div>
  <div>
    {{5/2}}
    <div>
      {{10/2}}
    </div>
  </div>
  <script>
    var app = angular.module('myAngularApp', []);
  </script>
</body>
</html>
```

ng-init

- دایرکتیو **ng-init** برای تعیین مقدار اولیه (**initial values**) در انگولار بکار می رود. به طور متداول ما از دستور ng-init استفاده نمی کنیم بلکه از یک کنترلر یا ماژول به جای آن استفاده می کنیم.

Example

```
<div ng-app="" ng-init="firstName='John'">  
  
  <p>Name: <input type="text" ng-model="firstName"></p>  
  <p>You wrote: {{ firstName }}</p>  
  
</div>
```

ng-model

- دایرکتیو ng-model عناصر html را به خصوصیت در شی Scope متصل (bind) می کند.
- دستور ng-model همچنین کارهای زیر را انجام می دهد:
 - ✓ اعتبار سنجی را برای داده های اپلیکیشن از قبیل عدد و ایمیل و داده های ضروری، انجام می دهد.
 - ✓ وضعیت داده های اپلیکیشن را مشخص می کند مانند: نامعتبر، زشت، خطا و ...
 - ✓ کلاس های CSS را برای عناصر HTML ارائه می دهد.
 - ✓ عناصر HTML را به فرم های HTML پیوند می دهد.

پیوند دادن داده ها Data Binding

- پیوند دادن داده ها در انگولار در واقع عبارات انگولار را با داده های انگولار منطبق می کنند.

Example

```
<div ng-app="" ng-init="firstName='John'">

  <p>Name: <input type="text" ng-model="firstName"></p>
  <p>You wrote: {{ firstName }}</p>

</div>
```

The `{{ firstName }}` expression, in the example above, is an AngularJS data binding expression.

Data binding in AngularJS binds AngularJS expressions with AngularJS data.

`{{ firstName }}` is bound with `ng-model="firstName"`.

Example

```
<div ng-app="" ng-init="quantity=1;price=5">  
  
Quantity: <input type="number" ng-model="quantity">  
Costs:    <input type="number" ng-model="price">  
  
Total in dollar: {{ quantity * price }}  
  
</div>
```



Code 4.txt

تکرار کردن عناصر HTML

- دستور **ng-repeat** یک عنصر HTML را تکرار می کند.

Example

```
<div ng-app="" ng-init="names=['Jani','Hege','Kai']">
  <ul>
    <li ng-repeat="x in names">
      {{ x }}
    </li>
  </ul>
</div>
```

Looping with ng-repeat:

- Jani
- Hege
- Kai



Code 5.txt

مثال

- در مثال زیر از دستور ng-repeat بر روی یک آرایه از اشیاء استفاده شده است.

Example

```
<div ng-app="" ng-init="names=[
  {name: 'Jani', country: 'Norway'},
  {name: 'Hege', country: 'Sweden'},
  {name: 'Kai', country: 'Denmark'}]">

  <ul>
    <li ng-repeat="x in names">
      {{ x.name + ', ' + x.country }}
    </li>
  </ul>

</div>
```



Code 6.txt

مثال

```
<!DOCTYPE html>
<html>
<head>
  <script src="~/Scripts/angular.js"></script>
  <style>
    div {
      border: 1px solid green;
      width: 100%;
      height: 50px;
      display: block;
      margin-bottom: 10px;
      text-align:center;
      background-color:yellow;
    }
  </style>
</head>
<body ng-app="" ng-init="students=['Bill','Steve','Ram']">
  <ol>
    <li ng-repeat="name in students">
      {{name}}
    </li>
  </ol>
  <div ng-repeat="name in students">
    {{name}}
  </div>
</body>
</html>
```

Result:

1. Bill
2. Steve
3. Ram

Bill

Steve

Ram



Code 7.txt

ایجاد دایرکتیو های جدید Create New Directives

- علاوه بر directive های built-in در انگولار، می توان دایرکتیوهای جدید را نیز ایجاد نمائیم. برای این منظور از تابع **directive**. باید استفاده کرد. برای فراخوانی دایرکتیو جدید، یک عنصر HTML با نام همان دایرکتیو را ایجاد می کنیم. در نام گذاری باید نام دایرکتیور را به صورت camel case و در هنگام فراخوانی با dash استفاده کرد.

When naming a directive, you must use a camel case name, `w3TestDirective`, but when invoking it, you must use - separated name, `w3-test-directive`:

Example

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

<w3-test-directive></w3-test-directive>

<script>
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
    return {
        template : "<h1>Made by a directive!</h1>"
    };
});
</script>

</body>
```

Made by a directive!



Code &txt

نکته

- فراخوانی دایرکتیو جدید ایجاد شده می تواند با موارد زیر انجام شود:

Element name ✓

Attribute ✓

Class ✓

Comment ✓

مثال

Element name

```
<w3-test-directive></w3-test-directive>
```

Class

```
<div class="w3-test-directive"></div>
```

Attribute

```
<div w3-test-directive></div>
```

Comment

```
<!-- directive: w3-test-directive -->
```

محدود سازی Restrictions

- با ایجاد محدودیت به دایرکتیوها می توان تنها بعضی از متدهای آن را فراخوانی کرد.

- E for Element name
- A for Attribute
- C for Class
- M for Comment

By default the value is `EA`, meaning that both Element names and attribute names can invoke the directive.

Example

By adding a `restrict` property with the value `"A"`, the directive can only be invoked by attributes:

```
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
    return {
        restrict : "A",
        template : "<h1>Made by a directive!</h1>"
    };
});
```

Made by a directive!



Code 9.txt

عبارت AngularJS Expression

- عبارات انگولار، بین دابل آکولاد قرار می گیرند مانند `{{ expression }}`. عبارات انگولار، دقیقا مانند دستور `ng-bind`، داده ها را به HTML پیوند می دهند. عبارات انگولار، بسیار شبیه به عبارات جاوا اسکریپت هستند: این عبارات می توانند شامل **لیترال** (literals) و **اپراتور** (operators) و **متغیرها** (variables) باشند.

Example: Expression

```
<!DOCTYPE html>
<html >
<head>
  <script src="/Scripts/angular.js"></script>
</head>
<body >
  <h1>AngularJS Expression Demo:</h1>
  <div ng-app>
    2 + 2 = {{2 + 2}} <br />
    2 - 2 = {{2 - 2}} <br />
    2 * 2 = {{2 * 2}} <br />
    2 / 2 = {{2 / 2}}
  </div>
</body>
</html>
```

Result:

2 + 2 = 4

2 - 2 = 0

2 * 2 = 4

2 / 2 = 1

مثال

Example: Let AngularJS change the value of CSS properties.

Change the color of the input box below, by changing its value:

Example

```
<div ng-app="" ng-init="myCol='lightblue'">  
  
  <input style="background-color:{{myCol}}" ng-model="myCol">  
  
</div>
```

Change the value of the input field:

Change the value of the input field:



Code 10.txt

ماژول ها AngularJS Modules

- ماژول در Angular یک container برای قسمت های مختلف برنامه می باشد.

An AngularJS module defines an application.

The module is a container for the different parts of an application.

The module is a container for the application controllers.

Controllers always belong to a module.

نحوه ایجاد ماژول Creating a Module

- برای ایجاد ماژول از تابع `angular.module` استفاده می شود.

A module is created by using the AngularJS function `angular.module`

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

<script>

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

</script>
```

The "myApp" parameter refers to an HTML element in which the application will run.

Now you can add controllers, directives, filters, and more, to your AngularJS application.

نحوه ایجاد ماژول Creating a Module

Example: Create Application Module

```
<!DOCTYPE html>
<html >
<head>
  <script src="/Scripts/angular.js"></script>
</head>
<body ng-app="myApp">
  @* HTML content *@
  <script>
    var myApp = angular.module('myApp', []);

  </script>
</body>
</html>
```

Create Controller Module

Add a controller to your application, and refer to the controller with the `ng-controller` directive:

Example

```
<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>
```

John Doe



Code 11.txt

ایجاد ماژول در فایل جداگانه Modules in separate files

Example

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/aj
<body>

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

<script src="myApp.js"></script>
<script src="myCtrl.js"></script>

</body>
</html>
```

myApp.js

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

myCtrl.js

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

نحوه لود کتابخانه Angular

- اکثر اسکریپت ها (jQuery, Bootstrap, ...) را بهتر است در انتهای تگ `<body>` فراخوانی کرد، اما در مورد AngularJS توصیه می شود که این کتابخانه را **در تگ `<head>`** یا **در ابتدای تگ `<body>`** برای لود شدن قرار داد. این کار به این دلیل توصیه می شود که فراخوانی های `angular.module` تنها می تواند بعد از اینکه کتابخانه AngularJS لود شد، کامپایل شود.

مثال



Code 12.txt

Example

```
<!DOCTYPE html>
<html>
<body>
<script src="https://ajax.googleapis.com/ajax/lib

<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>
```

```
<!DOCTYPE html>
<html>
<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>

<p>It is recommended that you load the AngularJS
library either in the HEAD or at the start of the
BODY.</p>
<script src="https://ajax.googleapis.com/ajax/libs
/angularjs/1.6.9/angular.min.js"></script>
</body>
</html>
```



John Doe

```
{{ firstName + " " + lastName }}
```

ng-model Directive

- با دایرکتیو **ng-model** می توانید مقدار یک فیلد ورودی (input field) را به یک متغیری که در AngularJS ایجاد کرده اید، متقید (bind) نمائید.

Example

```
<div ng-app="myApp" ng-controller="myCtrl">  
  Name: <input ng-model="name">  
</div>
```

```
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
  $scope.name = "John Doe";  
});  
</script>
```

Name:



Code 13.txt

Two-Way Binding

- binding به صورت دو طرفه است. اگر کاربر مقدار داخل فیلد ورودی را تغییر دهد، مقدار خاصیت انگولار نیز تغییر می کند.

Example

```
<div ng-app="myApp" ng-controller="myCtrl">  
  Name: <input ng-model="name">  
  <h1>You entered: {{name}}</h1>  
</div>
```

Name:

You entered: John Doe



Code 14.txt

ادامه دارد...