



www.MicroDev.ir

آموزش برنامه نویسی بدون ترس

آموزش AngularJS

برای آنهایی که وقت ندارند.



ANGULARJS

by Google

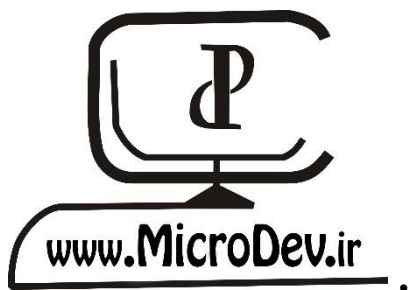
آموزش در کوتاهترین زمان

آموزش کاربردی برنامه نویسی

مولفین:

زهرا بیات قلی لاله

علی بیات قلی لاله



آموزش AngularJS برای آن‌هایی که وقت ندارند.

زهرا بیات قلی‌لاله

علی بیات قلی‌لاله

AngularJS آموزش

برای آن‌هایی که وقت ندارند.

مولفین : زهرا بیات قلی لاله – علی بیات قلی لاله

طراح جلد : زهرا بیات قلی لاله

ناشر : سایت کتاب راه

مشخصات ظاهری :

سال انتشار: فروردین ۹۵

قیمت : رایگان

فهرست:

تقدیم به:

مقدمه:

مراحل ایجاد AngularJS

قدم اول : درست کردن یک صفحه HTML ساده

قدم دوم : اضافه کردن AngularJS به پروژه

قدم سوم : اضافه کردن عبارت های `ng-controller` و `ng-model`

قدم چهارم : `ng-app` و مفهوم `scope`

قدم پنجم : راه اندازی پروژه و دیدن نتایج

قدم ششم : درست کردن `Function` و اضافه کردن `event`

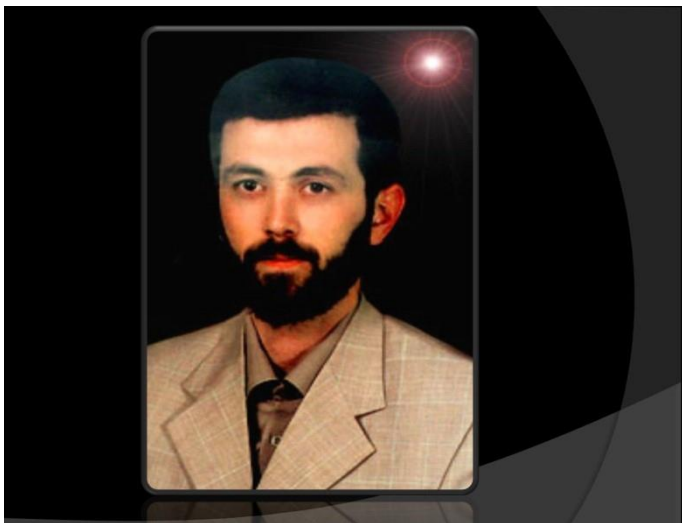
قدم هفتم : اضافه کردن `Validator` های HTML

قدم هشتم : استفاده از خاصیت `$valid` در `Angular`

قدم نهم : اجرای برنامه

قدم دهم : کد کامل شده

تقدیم به:



استاد مرحوم، جناب آقای

مهندس مسعود شکری پور

او که فکر و ذهنم را با ایده‌های جدیدش پرورش داد و راه رسیدن به تعالی را به من آموخت.

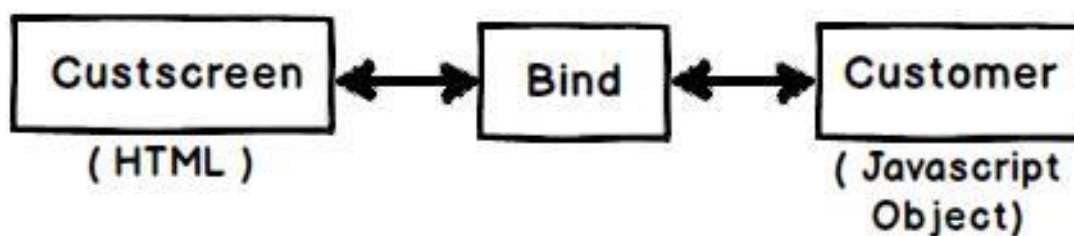
امیدوارم این کتاب روحش را شاد کند.

از شما عزیزانی که این کتاب را می‌خوانید خواهش می‌کنم برای شادی روحش یک فاتحه

بفرستید.

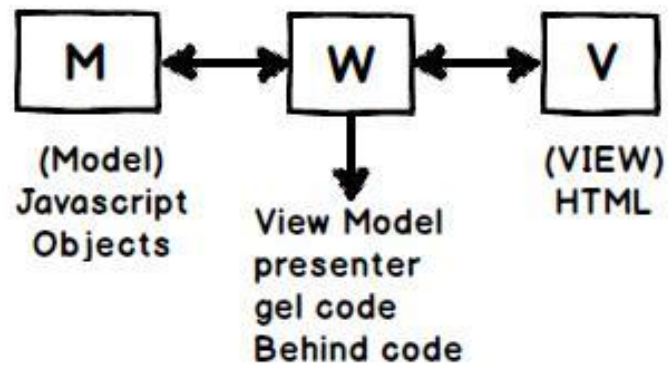


AngularJS (انگيولرجی اس) یکی از مشهورترین و بهترین فریم‌ورک‌های جاوا اسکریپت است. ویژگی قدرتمند **Data Binding** در **AngularJS** کار کردن با اشیای **JavaScript** و **HTML** را بسیار ساده کرده است. دلیل اصلی پیدایش **AngularJS** بایندینگ و بایندینگ بوده است. **AngularJS** یک بایندینگ فریم‌ورک است که رابط کاربری **HTML** را به اشیای **JavaScript** بایند می‌کند.



با انجام این کار، حجم زیادی از کدهایی که برای ایجاد ارتباط بین **HTML** و اشیای **JavaScript** نوشته می‌شد کاسته شده است.

از آنجا که از **AngularJS** از الگوهای طراحی **MVC** و **MVVM**، به طور همزمان پشتیبانی می‌کند، آن را به عنوان **MVW (Model-View-Whatever)** و یا "مدل، ویو و هر چیز دیگری" می‌شناسند.

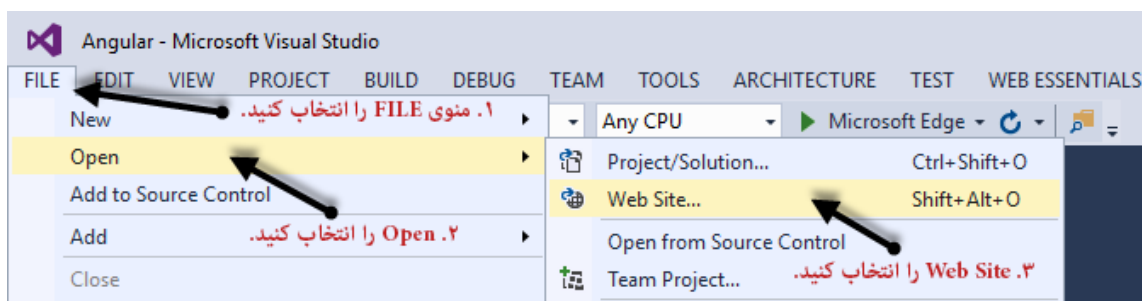


مراحل ایجاد AngularJS:

قدم اول : درست کردن یک صفحه HTML ساده

می‌خواهیم با هم یک پروژه ساده که فقط شامل HTML، CSS و JavaScript می‌باشد را ایجاد نماییم.

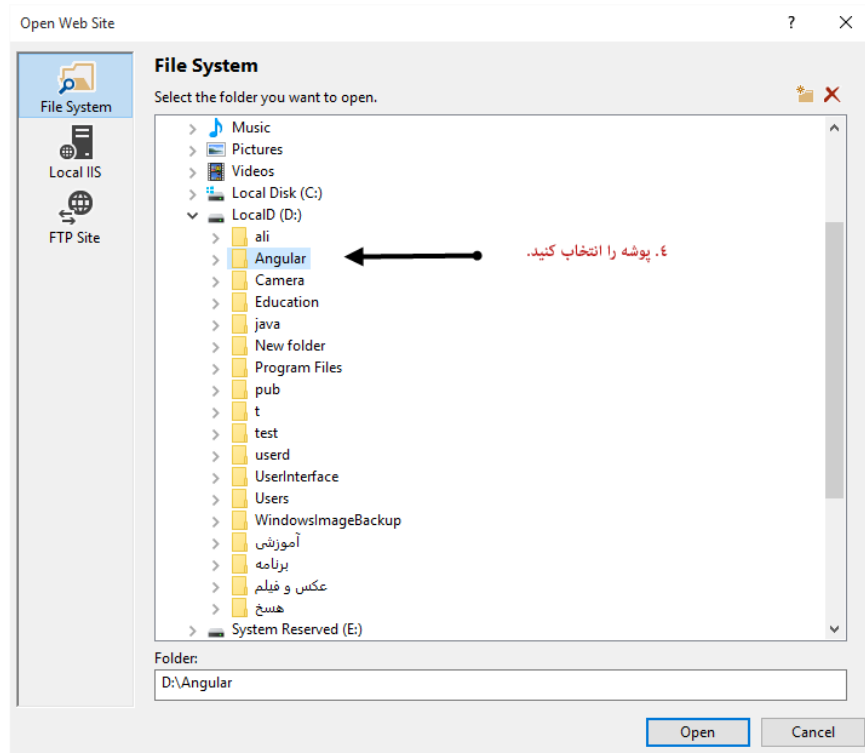
ابتدا یک پوشه در یک مسیر دلخواه درست کرده، سپس **Visual Studio** را باز کنید و مانند تصویر زیر بر روی **File->Open->Website** کلیک نمایید.



نکته!!

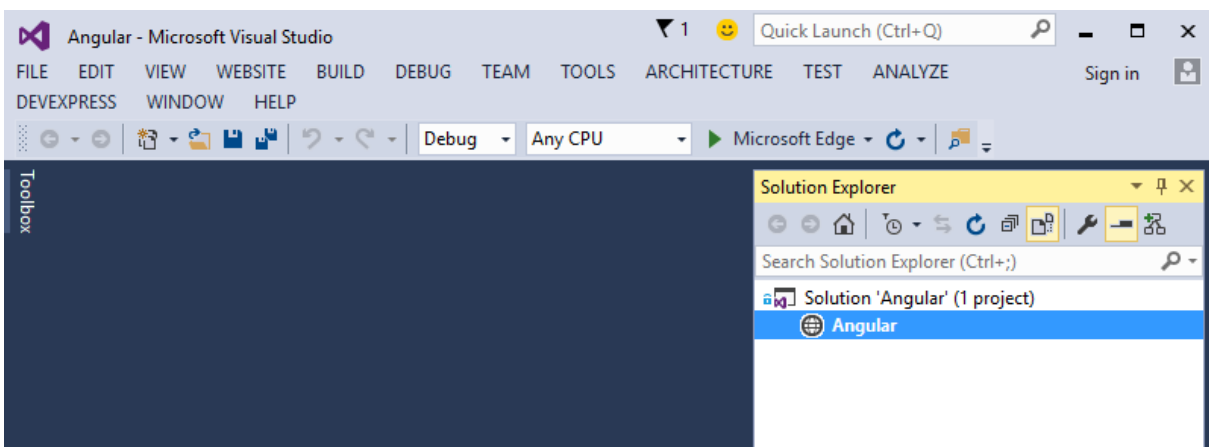
وقتی که شما یک پوشه را به عنوان Website باز می‌کنید، Visual Studio فایل‌های اضافی مانند web.config و global.asax و ... را به صورت خودکار به پروژه اضافه نمی‌کند.

حالا مانند تصویر زیر مسیر پوشه‌ای که درست ایجاد کرده اید را به آن دهید.

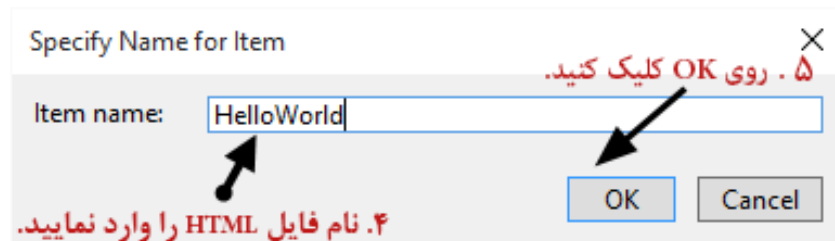
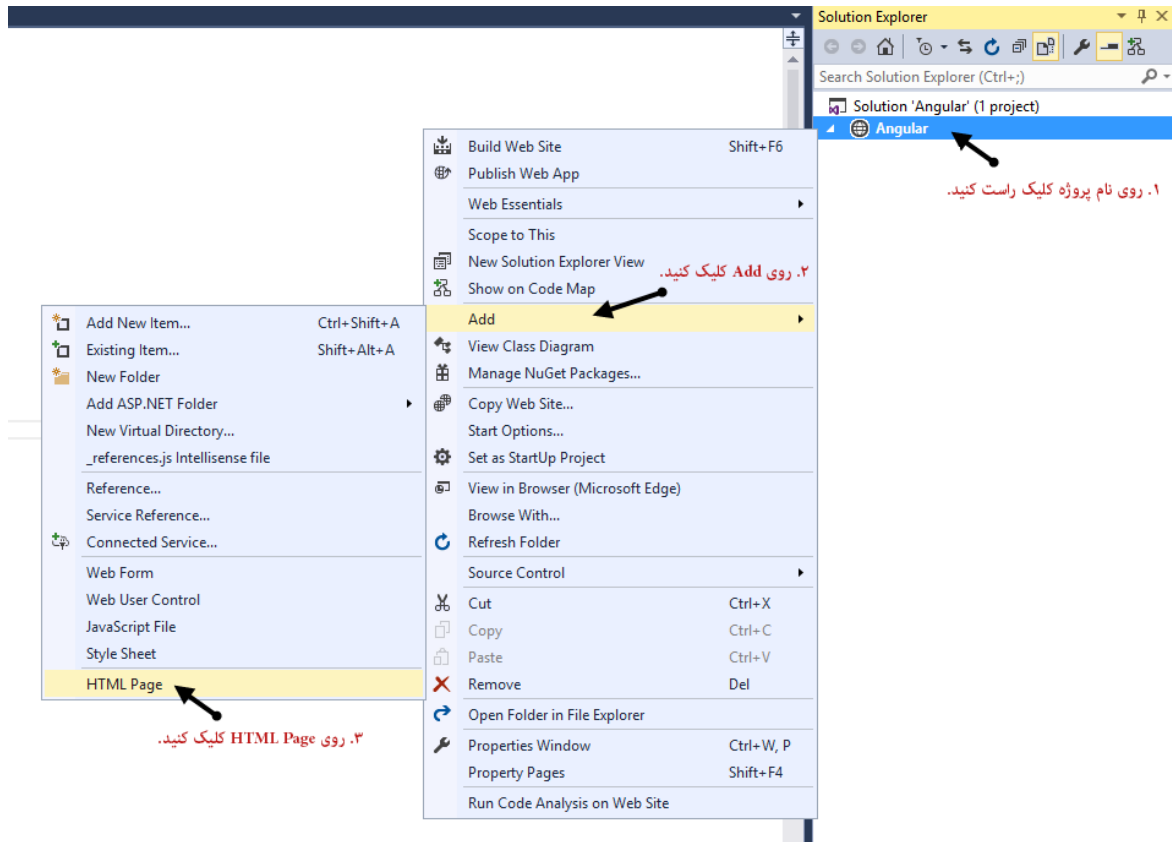


کار شما به با موفقیت به پایان رسید.

شما توانستید یک پروژه بسیار ساده ایجاد کنید، که هیچ چیزی در آن وجود ندارد. این فقط یک پروژه خالی است که فایل های مخصوص مایکروسافت هم در آن دیده نمی شود.



بیا یک صفحه **HTML** ساده درست کنیم و اسم آن را **HelloWorld.html** بگذاریم.



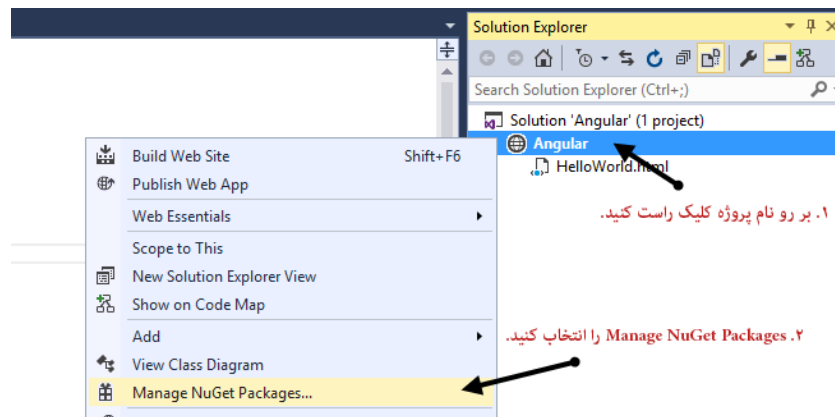
در این صفحه **HTML** یک **TextBox** ایجاد می‌کنیم و نام آن را **txtHello** قرار می‌دهیم. سپس این **TextBox** را به تابع جاوا اسکریپتی به نام **Hello** بایند خواهیم کرد.

و حالا می‌خواهیم از AngularJS برای بایند کردن این دو موجودیت استفاده نماییم.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script>
function Hello() {
this.txtHello = "HELLO WORLD";
}
  </script>
</head>
<body>
  <div id=" HelloWorldScreen">
    <input id="txtHello" type="text" />
  </div>
</body>
</html>
```

قدم دوم : اضافه کردن AngularJS به پروژه

حالا که رابط کاربری HTML و کلاس های JavaScript را در جای خود قرار دادیم، بیاید اول AngularJS را دانلود کنیم. برای این کار مانند تصویر روی پروژه راست کلیک کرده و گزینه **Manage NuGet Packages** را انتخاب کنید.



کلمه **Angular** را در کادر جستجو وارد کنید، سپس **AngularJS Core** را انتخاب

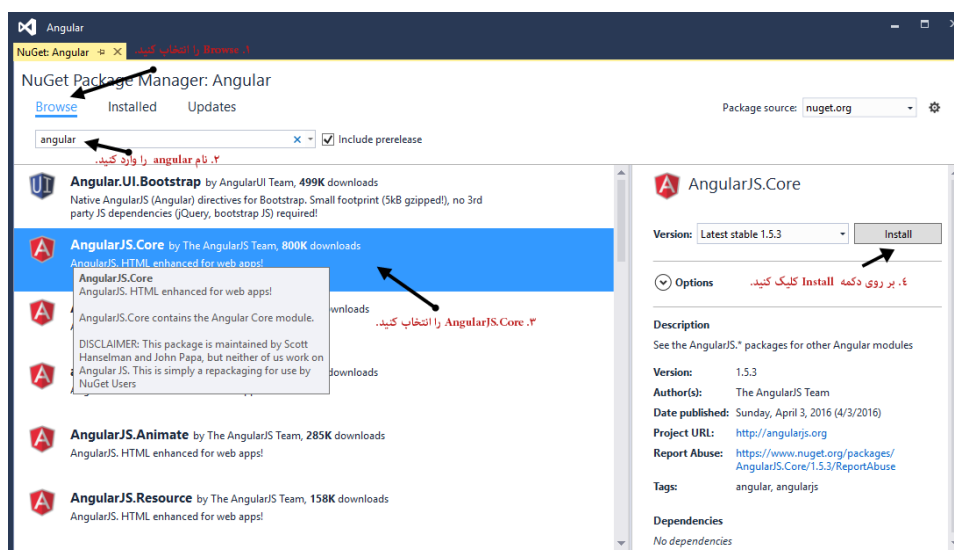
نمایید و بر روی دکمه **Install** کلیک کنید.

در لیست ظاهر شده چندین **AngularJS** وجود دارد که **Extension** های

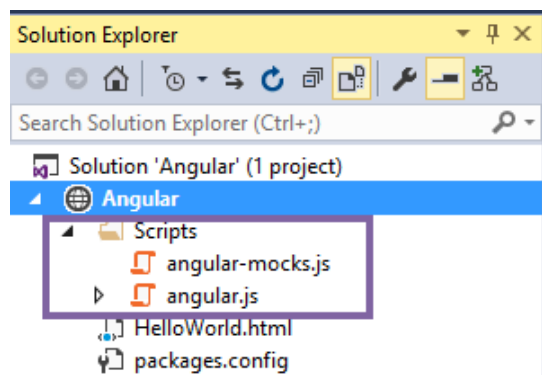
AngularJS می باشند. که به تدریج در قسمت های بعدی راجع با آنها صحبت خواهیم

کرد. و مهم ترین آن ها **Angular core** است، روی آن کلیک کنید تا به پروژه شما اضافه

گردد.



پس از نصب، شما می توانید فایل های **Angular** را در پوشه **Scripts** ببینید.



حالا مانند تصویر زیر با استفاده از تگ **script** به فایل **Angular** (به خط **Bold** شده نگاه کنید) **Reference** دهید.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<script src="Scripts/angular.js"></script>
<!--..... Code deleted for simplification-->
```

قدم سوم : اضافه کردن عبارت های **ng-model** و **ng-controller**

حالا برای اتصال کلاس **HelloWorld** با کادر متنی **HTML** به دو چیز نیاز دارید :

- درست کردن یک نمونه از کلاس **HelloWorld**

- دوم، تنظیم خاصیت شی **HelloWorld** به **TextBox**

برای این کار ما به **ng-controller** و **ng-model** نیاز داریم.

به **ng-controller** و **ng-model** در اینگیولرجی اس **directive** گفته می شود.

به یاد داشته باشید که اینگیولرجی اس یک زبان برنامه نویسی اعلانی است.

برنامه نویسی اعلانی یک پارادایم برنامه نویسی است که در آن منطق و هدف محاسبات بدون

شرح چگونگی انجام آنها بیان می شود. در **AngularJS** شما فقط **directive** ها را اضافه

می کنید، و بدون اینکه نیاز باشد چگونگی انجام آن کار را به آن بگویید کاری که می خواهید

را برای شما انجام می دهد.

دایرکتیو **ng-controller** یک شی از کنترلر **HelloWorld** درست کرده و آن را به تگ **div** بایند می‌کند، و **ng-model** مقدار فیلد **input** را به پراپرتی **txtHello** بایند می‌کند.

همانطور که در تصویر زیر نشان داده شده، برای نمایش پراپرتی **txtHello** روی صفحه نمایش، به **{ { عبارت مورد نظر } }** نیاز داریم. بنابراین وقتی چیزی را در **TextBox** تایپ می‌کنیم، در تگ **div** نمایش داده خواهد شد.

```
<div id=" HelloWorldScreen" ng-controller="HelloWorld">
  <input ng-model="txthello" id="txthello" type="text" />
  <div id="div1">{{txthello}}</div>
</div>
```

قدم چهارم : ng-app و مفهوم scope

حالا اگر دایرکتیو **ng-controller** نمونه‌ها (**instance**) را درست می‌کند. اگر مانند کد پایین، چندین دایرکتیو **ng-controller** در صفحه **HTML** خود داشته باشید چه اتفاقی خواهد افتاد؟؟

یک **instance** درست می‌کند یا چند **instance** ؟

در باره این موضوع فکر کنید...

```
<div id=" HelloWorldScreen" ng-controller="HelloWorld">
  <input ng-model="txthello" id="txthello" type="text" />
  <div id="div1">{{txthello}}</div>
</div>
<div id=" HelloWorldScreen" ng-controller="HelloWorld">
  <input ng-model="txthello" id="txthello" type="text" />
  <div id="div1">{{txthello}}</div>
</div>
```

بدیهی است که شما به عنوان یک برنامه‌نویس، دوست دارید **instance** های مختلف از **HelloWorld** داشته باشید و بتوانید به هر دو **div** موجود در صفحه **HTML** بایند کنید. این دقیقاً همان کاری است که **AngularJS** برای ما انجام می‌دهد.

اما حالا که دو **instance** متفاوت در یک صفحه **HTML** داریم، محدوده استفاده (یا **scope**) این دو **instance** چگونه تعیین می‌شود. آیا در تمام صفحه‌ی **HTML** قابل استفاده است؟

محدوده استفاده (یا **scope**) این دو **instance** با تگ **<div>** آغاز شده و با تگ **</div>** به پایان می‌رسد. **AngularJS** وقتی به دایرکتیو **ng-controller** برخورد می‌کند، این **instance** ها را به صورت خودکار می‌سازد.

حالا اگر بخواهیم این **instance** ها با هم ارتباط داشته باشند و داده هایشان را با هم به اشتراک بگذارند چه باید کرد؟؟

ما چیزی به نام **ng-app** داریم که این کار را برای ما انجام می‌دهد. **ng-app** یک شی درست می‌کند که از طریق آن تمام **controller** ها را کنترل کرد و به آنها این امکان را می‌دهد تا بتوانند داده‌هایشان را با هم به اشتراک بگذارند.

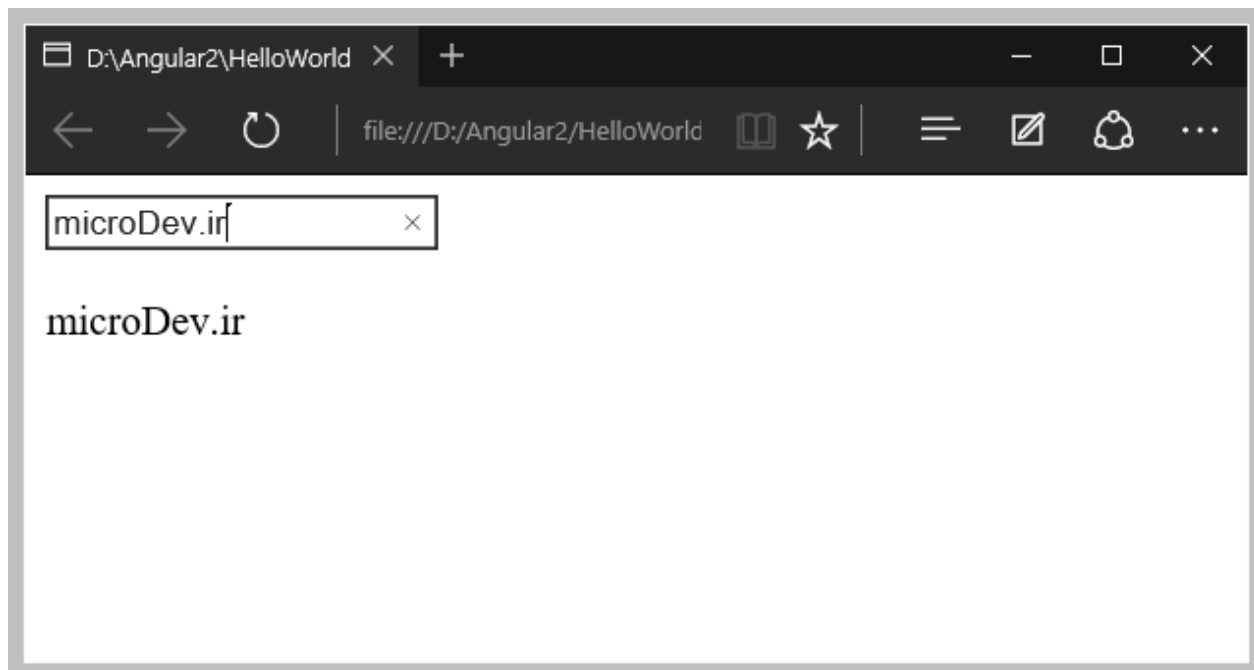
قدم پنجم : راه اندازی پروژه و دیدن نتایج

خوب، حالا که همه تنظیمات را انجام دادیم، برای شروع کار پروژه **Angular** به مقداری کد احتیاج داریم. ما می‌خواهیم **Angular** اشیا **controller** و **ng-app** را بسازد و بایندینگ را برای ما انجام دهد.

بنابراین به کد هایی که در پایین می‌بینید احتیاج داریم. اول یک **instance** از **app** می‌سازیم، سپس یک شی **controller** به **app** اضافه می‌کنیم.

تمام کار همین بود.

کلید **Ctrl+F5** را فشار دهید و از بایندینگ خودکار لذت ببرید. حالا وقتی چیزی را در **TextBox** وارد می‌کنید، عبارت پایینش به صورت خودکار **Update** می‌شود.



کد نهایی به صورت کامل

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script src="Scripts/angular.js"></script>
  <script>
    function Hello($scope) {
      $scope.txthello = "HELLO WORLD";
    }
    var myApp = angular.module("myApp", []);
    myApp.controller("HelloWorld", Hello);
  </script>
</head>
<body bgcolor="red" ng-app="myApp">
  <div ng-controller="HelloWorld">
    <input ng-model="txthello" type="text" /><br /><br />
    <div id="div1">{{txthello}}</div>
  </div>
</body>
</html>
```

قدم ششم : درست کردن Function و اضافه کردن event

بیا یک **function** ساده به نام **Func1** را به کلاس **Hello** که در گام قبلی ایجاد کردیم، اضافه کنیم. **Func1** هر چیزی را که در متغیر **txthello** وارد کنید، از طریق **alert** نمایش خواهد داد.

```
function Hello($scope) {
  $scope.txthello = "";
  $scope.Func1 = function () {
    alert($scope.txthello);
  }
}
```

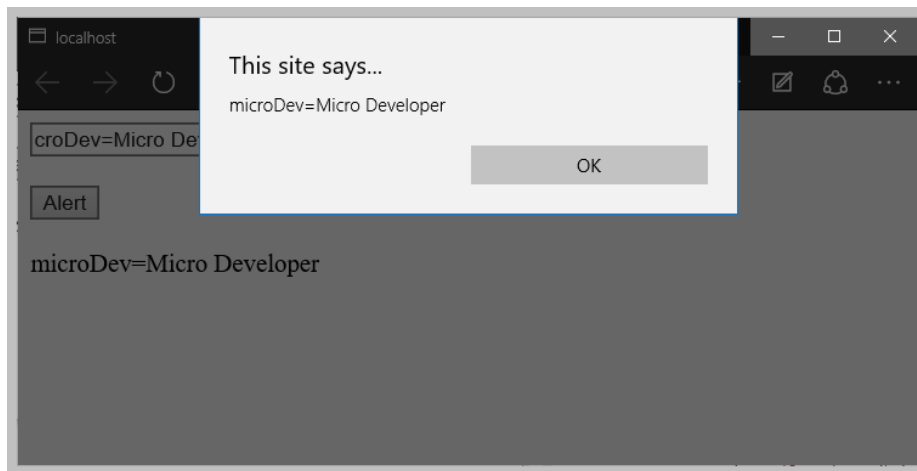
حالا اگر فکر می کنید که متد بالا (**Func1**) می تواند از طریق یک دستور ساده جاوااسکریپت فراخوانی شود، سخت در اشتباهید. **Instance** هایی که توسط **Angular**

ساخته می‌شوند، فقط از طریق **event** های **Angular** قابل دستیابی هستند. پس کدهای زیر کار نخواهند کرد.

برای فراخوانی **function** ما به **event** های **Angular** احتیاج داریم. بنابراین اگر می‌خواهیم **Func1** را صدا بزنیم، به **ng-click** در **Angular** نیاز داریم.

```
<input type="button" onclick="Fun1()" name="name" value="Alert" />
```

پس از انجام این کار باید بتوانید هنگام کلیک بر روی دکمه ی **Alert** در عمل رویداد کلیک را ببینید. (مانند تصویر زیر)



در پایین کد کامل شده **event** ها در **Angular** که توضیح داده شد، آورده شده است.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script src="Scripts/angular.js"></script>
  <script>
    function Hello($scope) {
      $scope.txtHello = "microDev.ir";
      $scope.Func1 = function () {
```

```

        alert($scope.txthello);
    }
}
var myApp = angular.module("myApp", []);
myApp.controller("HelloWorld", Hello);
</script>
</head>
<body ng-app="myApp">
    <div id=" HelloWorldScreen" ng-controller="HelloWorld">
        <input ng-model="txthello" id="txthello" type="text" /><br /><br />
        <input type="button" ng-click="Fun1()" name="name" value="Alert"
/><br /><br />
        <div id="div1">{{txthello}}</div>
    </div>
</body>
</html>

```

قدم هفتم : اضافه کردن Validator های HTML

Validation های **Angular** به صورت توکار از **Validator** های **HTML5** استفاده

می کنند. بنابراین بگذارید بگوییم ما می خواهیم مطمئن شویم که **txthello** نمی تواند

مقدار **null** بگیرد. برای این منظور نیاز داریم که کارهای زیر را انجام دهیم:

آراستن **input** با **Validator** های **HTML**.

برای مثال، در کد زیر ما خاصیت **required** را اضافه کرده ایم. در **HTML5**،

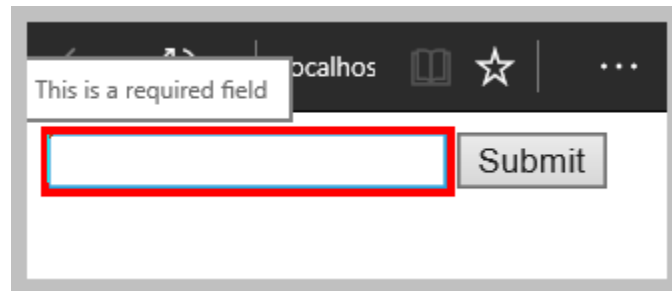
Validator های زیادی وجود دارند، که می توانید از هر کدام از آنها استفاده کنید.

این **input text** درون تگ **form** قرار داده شده و باید خاصیت **name** آن با

texthello پر شود. دقت داشته باشید که **id** در **Angular** اهمیتی ندارد و فقط

name برای آن مهم است.

حالا اعتبارسنجی **HTML** توسط **Browser** به صورت خودکار انجام می پذیرد. وقتی کاربر بر روی دکمه **submit** کلیک می کند، باید با پیغام خطای زیر مواجه شود. اما ما نمی خواهیم که پیغام خطای **Internet Explorer** یا **chrome** را ببینیم. ما می خواهیم **Angular** کنترل پیغام خطاها را در دست گیرد.



کد زیر حاوی تغییراتی است که در سه مرحله بالا اعمال کردیم.

```
<form name="frm1">  
  <input name="texthello" ng-model="txthello" id="txthello" type="text" required />  
</form>
```

قدم هشتم: استفاده از خاصیت **\$valid** در **Angular**

حالا که همه ی تنظیمات را انجام دادیم و **Validator** های توکار **HTML** را وارد کردیم، می خواهیم وقتی کاربر چیزی را تایپ می کند، دو کار انجام شود:

- فعال و غیر فعال کردن دکمه در هنگام اعتبارسنجی
- نمایش دادن و پنهان کردن پیغام خطا در تک **div**

برای این منظور، به دو دایرکتیو **ng-disabled** و **ng-show** در **Angular** احتیاج داریم. وقتی هر کدام از **Validator** ها انجام شود، ارزیابی صورت می گیرد و خاصیت **\$valid** بسته به داده های وارد شده **true** یا **false** می شود.

```
<form name="frm1" id="frm1" novalidate>
  <input name="texthello" ng-model="texthello" id="texthello"
type="text" required /><br />
  <div ng-show="!(frm1.texthello.$valid)"> این قسمت را پر کنید</div>
  <br />
  <input type="button" ng-click="Fun1()" name="name" value="Alert"
ng-disabled="!(frm1.$valid)" /><br /><br />
</form>
```

قدم نهم : اجرای برنامه

حالا که همه ی کارها با موفقیت انجام شد. کلید **Ctrl+F5** را فشار داده و از نتیجه کار سخت خود لذت ببرید.

در پایین تصویری را می بینید که برنامه را در حال اجرا به شما نشان می دهد.



قدم دهم : کد کامل شده

در زیر کد کامل شده تا به اینجا را، که شامل **event** ها و **Validation** هاست نشان

داده شده است.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <script src="Scripts/angular.js"></script>
  <script>
    function Hello($scope) {
      $scope.txthello = "";
      $scope.Fun1 = function () {
        {
          alert($scope.txthello);
        }
      }
    }
    var myApp = angular.module("myApp", []);
    myApp.controller("HelloWorld", Hello);
  </script>
</head>
<body ng-app="myApp">
  <div id=" HelloWorldScreen" ng-controller="HelloWorld">
    <form name="frm1" id="frm1" novalidate>
      <input name="texthello" ng-model="txthello" id="txthello"
type="text" required /><br />
      <div ng-show="!(frm1.texthello.$valid)"> کنید پر را قسمت این </div>
      <br />
      <input type="button" ng-click="Fun1()" name="name" value="Alert"
ng-disabled="!(frm1.$valid)" /><br /><br />
    </form>
    <div id="div1">{{txthello}}</div>
  </div>
</body>
</html>
```

لیست کتاب های نوشته شده:

<http://www.ketabrah.ir> - زهرا بیات - فلی

[/author/۱۱۱۵](#)

آدرس کتاب ها:





C++ بهتر است یا C#؟

 آموزش برنامه نویسی بدون ترس



ایده پرداز

- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی برنامه نویسی

مولفین:
 زهرا بیات قلی لاله
 علی بیات قلی لاله





C#

 معمای شی گرا

آموزش برنامه نویسی بدون ترس

ایده پرداز

- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی برنامه نویسی

مولف:
 زهرا بیات قلی لاله





 آموزش برنامه نویسی بدون ترس

سی شارپ برای بچه ها

جلد دوم



- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی برنامه نویسی

مولفین:
 زهرا بیات قلی لاله
 علی بیات قلی لاله





 آموزش برنامه نویسی بدون ترس

سی شارپ برای بچه ها

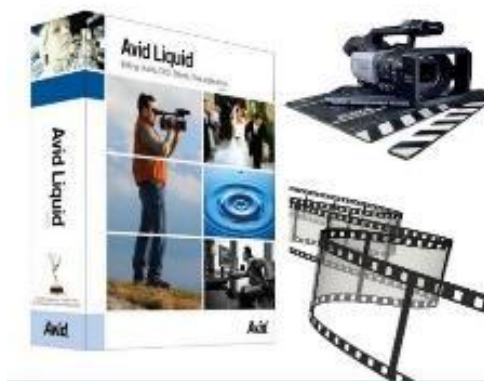
جلد اول



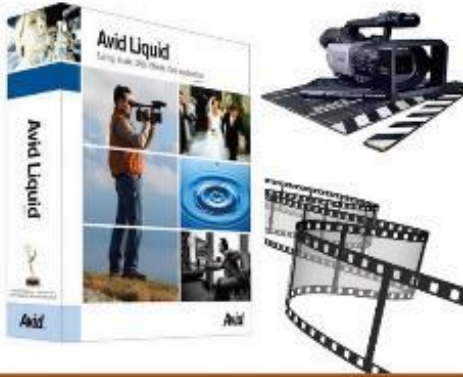
- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی برنامه نویسی

مولفین:
 زهرا بیات قلی لاله
 علی بیات قلی لاله



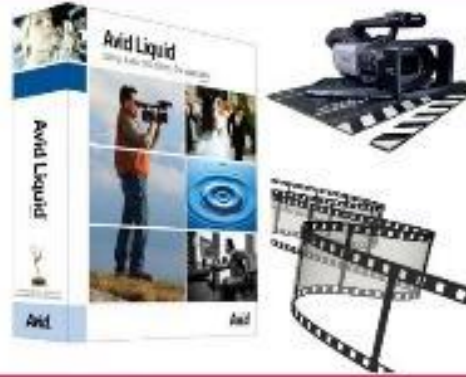


آموزش کاربردی
Avid Liquid 7.2
جلد پایانی و کاربردی



مؤلفین:
زهرا بیات قلی لاله
علی بیات قلی لاله

آموزش کاربردی
Avid Liquid 7.2
جلد سوم



مؤلفین:
زهرا بیات قلی لاله
علی بیات قلی لاله

شبکه
ساده و کاربردی

- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی و سریع



مؤلف:
زهرا بیات قلی لاله
علی بیات قلی لاله

تکنیک های کاربردی
Adobe Illustrator CC

- ✓ آموزش در کوتاهترین زمان
- ✓ آموزش کاربردی و سریع



مؤلف:
زهرا بیات قلی لاله
علی بیات قلی لاله

Adobe Illustrator CC

آموزش کامپیوتر مخصوص خانم ها در آموزشگاه ایده پرداز

با تشکر از شما دوست عزیز که هم اکنون این کتاب را برای خواندن انتخاب کردید.

من زهرا بیات، موسس آموزشگاه ایده پرداز (در گلستان بهارستان) هستم.

هر کس در زندگی هدفی دارد، اما بهترین هدفها، متعالی ترین آنهاست.

تاسیس ایده پرداز، ۲ دلیل داشت:

۱- کمک به خانمها و کودکانی که با توجه به سن یا محدودیت مالی نمی توانند در کلاسهای

آموزشگاهها شرکت کنند .

۲- آموزش نرم افزارهای به روز، که باز هم به دلیل کمبود امکانات، در اکثر آموزشگاهها یا

تدریس نمی شود و یا با هزینه گزاف تدریس می شود. و عده ای که طالب این آموزشها

هستند از علم کامپیوتری عقب می مانند. من در این آموزشگاه با به روزترین تجهیزات، به

روزترین نرم افزارها را با قیمت مناسب آموزش می دهم.

تخصص من در انواع زمینه های:

• نرم افزارهای اداری (Word, Excel, Access, PowerPoint, OneNote)

• نرم افزارهای گرافیکی (فتوشاپ، کورل، فری هند، میکس و مونتاز، فلش و تری دی

مکس..)

• برنامه نویسی (C#, ASP.Net MVC و C++)

دوستان عزیز، فقط برای ثبت نام می‌توانید با شماره‌های زیر ارتباط برقرار کنید. و سوالات برنامه‌نویسی خود را به ایمیل من بفرستید تا در اسرع وقت پاسخگو باشم.

آدرس : جاده ساوه، گلستان ، فلکه اول، ۲۴متری ارغوان غربی، خیابان ولایت، پلاک ۳۲

Email: BytMohandes@yahoo.com

Tel: ۰۹۳۷۳۷۱۹۴۰۵ - ۰۹۳۹۳۹۳۹۹۲۴