

بسم الله الرحمن الرحيم

دورة AngularJS

تقديم: أنيس حكمت أبو حميد.

البريد الإلكتروني: aneeshikmat@2nees.com

الموقع الإلكتروني: www.2nees.com

الفهرس

- المقدمة (مهمة جدا بالإضافة الى الأساسيةات لذكر الكثير من النقاط التي لن نعود لشرحها .. مثل تعريف المصطلحات الرئيسية)
- أساسيات ال Angular
- Expressions
- Directives
- Modules
- Controller
- Filters
- إنشاء angular tabs من خلال ال tabs
- التعامل مع ال FORMS

الفهرس

- Custom Directive وال ng-include
- \$http
- التعامل مع الجداول في AngularJS Table
- AngularJS Create New Module
- الخاتمة

(الدورة مع الأمثلة يمكنك تحميلها من 2nees.com/ar/courses)

المقدمة

الحمد لله رب العالمين، أعطى اللسان، وعلم البيان، وخلق الإنسان، فبأي
الأاء ربكمَا تكذبان.. لك الحمد يا من هو للحمد أهل، أهل الثناء والمجد،
أحق ما قال العبد وكلنا لك عبد.

لك الحمد.. من ضعيف يطلب نصرتك..

لك الحمد.. من فقير يطلب غناك..

لك الحمد.. من ذليلٍ يطلب عزك..

لك الحمد.. ما دعوناك إلا حسن ظنِّك.. وما رجوناك إلا ثقةً فيك، وما
خفناك إلا تصدقًا بوعدك ووعيدك.. فلائك الحمد حتى ترضى، فالحمد لله
رب العالمين والصلاه والسلام على علم الأعلام، وإمام كل إمام، محمد
بن عبد الله صل الله عليه وسلم وعلى آله وصحبه أجمعين، وبعد:

المقدمة

نعود لكماليوم بدورة جديدة ورائعة في مجال تطوير المواقع، وهي دروة AngularJS ^_ ^، وقبل الحديث عن بيئة العمل هذه، يجب عليك أن تعرف متطلبات هذه الدورة وهي:

- JS - CSS – HTML ..
- (ليست متطلبات ضرورية) jQuery, PHP, Database ..
- بالغالب سيكون عملنا مقترباً على جانب الـ FRONT-END لكن سنحتاج في بعض الأمثلة الى وجود سيرفر محلي ويتمكنك الحصول عليه من خلال تحميل بيئة تطوير مواقع مثل wamp أو غيره حسب نظام التشغيل الخاص بك .. وذلك لتنفيذ بعض الأمثلة على جلب بيانات عن طريق الـ ajax أو json .. الخ

المقدمة

روابط مهمة لل AngularJS (مراجعة هذه الدورة المتواضعة)

- [AngularJS API Docs](#)

- [Guide to AngularJS Documentation](#)

- [W3schools Angular](#)

- كتاب AngularJS Up & Running

PDF has a nice description) Shaping up with angular
•
(& examples

جميع هذه المواقع أو الكتب أو الشروحات التي ذكرت أو لم تذكر تم استخدامها بما يلبي الحاجة .. فمثلاً كان الشرح من كتاب AngualarJS .. Up & Running و الدوال وغيرها من روابط الموقع الرسمي، والأمثلة من w3schools و Shaping up with angular ... بالإضافة إلى الملخصات المختلفة، ومن خلال التعامل مع بيئة العمل هذه ..

المقدمة

والآن قبل البدء بهذه الدورة .. أحب أن أنقل لكم قوله سمعته من "بريج بالك" يقول فيه أن ال js و مكاتبها التي نعرفها مثل ال jQuery نستخدمها لجعل صفحات الويب Dynamics بناها على التلاعب بالعناصر والمواقع وغيرها .. لكن ال AngularJS هو فعليا يقوم على جعل ال HTML dynamic !! .. وشخصيا .. استهجنت هذه الفكرة في بداية الأمر .. لكن وبعد تعلمي وتجربتي لل Angular فإني أقول أن هذا أقل ما يمكن قوله لشدة روعتها ... وهذا ما ستشاهده أنت ..



المقدمة

أول أمر يخطر ببالنا الآن ..لماذا يجب أن أتعلم الـ Angular ؟ الجواب هو ..إذا كنت تستخدم الجافا سكريبت لجعل من موقعك فال الخيار الأمثل هو angular لأنه يتيح لك الآتي:

١. يساعدك على تنظيم الشيفرة البرمجة للمشروع.

٢. يساعدك على إنشاء مواقع متجاوبة مع الهاتف Responsive
٣. يمكنك من خلال الـ angular التعامل مع الـ * _ * jQuery
٤. سهلة جدا في الفحص واكتشاف الأخطاء.

إذا ما هو الـ Angular ؟

المقدمة

ال Angular هي بيئة عمل (Framework) مصممة لتضييف نوعاً من التفاعلية على ال HTML، ويكتفي من مميزاتها أنك لتعامل معها يكتفي أن تتعلم الجافا سكريبت (pure)، ويقوم مبدأ التقسيم البناءي لها على مبدأ ال MVVM أو MVC ...

مفهوم ال MVC: يجب أن تعرف ما هو ال MVC قبل البدء، لأنك ستتعامل مع بيئة عمل مصممة نوعاً ما لتكون قريبة من ال MVC، ولهذا فهم هذا النوع .. سيسهل عليك فهم ال angular كما أنه مهم لك في الأمام اذا أحببت تعلم بيئات عمل أخرى ..

المقدمة

MVC هي اختصار ل Model-View-Controller، وهو أحد الـ Design Pattern الأكثر شهرة، وتقوم فكرته على تقسيم المشروع (وخصوصاً الكبير منها) إلى وحدات منطقية صغيرة يتعارف على نسقها بين المبرمجين والمطوريين، وتكون هذه الوحدات هي نقاط الإنطلاق للبحث عن المشاكل والتطوير للأنظمة وبنائها وتحديثها، وتكون هذه الوحدات مقسمة إلى ٣ أجزاء أو قطع وهي:

١) model: فعلياً يمكننا اعتبارها مركز أي مشروع أو تطبيق، وعادة تحتوي على البيانات التي يمكن جلبها من الخادم (server) (عادة)، وأي بيانات يراها أي مستخدم بعد ذلك ستكون قادمة من جزء من هذا الـ model أو من أكثر من ... model

المقدمة

٢) view: هي الواجهة التي يتعامل معها المستخدم، وهي Dynamic وتفاعلية، وتتحدى معلوماتها بحسب البيانات القادمة من ال model، لذلك فهي تصمم إعتماداً على ما يحتوي ال model من بيانات.

٣) controller: هي الوحدة المنطقية التي تبني وتصل بين البيانات المختلفة وطريقة العرض المناسبة لها، أي أنها المسؤولة عن الربط بين ال View وال Model^٨، بالإضافة إلى ذلك، فهي التي تقوم بفصل وتقسيم البيانات القادمة من ال model وإجراء التغييرات اللازمة وغيرها من الأمور، وتعطيها لل view حسب احتياجها..

هذا ال pattern قدم فائدة كبيرة لنا، فمن خلاله أيضاً أمكننا أن نستخدم الشيفرة البرمجية في أكثر من مكان دون الحاجة إلى إعادة بناء العجلة

٨

المقدمة

الآن مميزات ال Angular :

١. AngularJS is a Single Page Application (SPA) . وإن كنت لا تعرف ما هو ال SPA فهو الموقع أو التطبيق الذي يقوم بتحميل جميع المصادر التي يحتاجها في أول عملية تحميل الصفحة.
٢. من مميزاتها أيضا أنها تحتاج إلى عدد أقل من الأسطر البرمجية مقارنة مع الجافا سكريبت أو ال jQuery .. أيضا فالشيفرة البرمجية الخاصة بها تعد شيفرة clean ^ * لأنك تسير على النمط المحدد لبيئة العمل هذه، ويمكن استخدامها في أماكن أخرى بما أن الشيفرة البرمجية قليلة وواضحة بشكل سهل.

المقدمة

٣. بوجود ال angular أصبح من السهل فهم أي تطبيق فقط من النظر الى ال HTML وال Controller المستخدم، وبالنسبة لمطوري ال angular فإن هذه الميزة وبمعنى آخر تسمح لك بإنشاء HTMLX لأن المسميات والتصنيفات واضحة من خلالها دون حتى الحاجة الى الاعتماد على ال html5 أو انتظار ال ... html6
٤. يعمل ال angular مع ال pure html، لذلك لن يجد المصمم صعوبة في تصميم وتطوير هذه الصفحات.
٥. نحن لسنا هنا بصدّد التخلّي عن ال UI jQuery أو حتى ال Bootstrap، وإنما ال Angular يندمج كطرف ثالث يقدم ما لديه من خدمات ليتم دمج هذه المكاتب والخصائص معاً ^ ^

أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ

فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ ۖ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ وَحْيُهُ ۚ
وَقُلْ رَبِّ رَزْنِي عِلْمًا

صدق الله العظيم

سورة طه [الآية: ١١٤]

أسسیات ال Angular

الآن .. لنبدأ فعليا بعملية تحميل وشرح ال Angular

تحميل ال Angular

قم بالدخول الى الموقع الرسمي <https://angularjs.org/> ومن ثم
قم بالضغط على تحميل، سيكون ناتج التحميل هو ملف مسمى ب
... angular.min.js

أيضا اختصارا ل الوقت .. عند تطبيق الأمثلة .. أريدك أن تقوم باستخدام ال
bootstrap ويمكنك ذلك من خلال الموقع <http://getbootstrap.com/> .. وسيكون الملف المطلوب من
التحميل هو .. bootstrap.min.css

أساسيات ال Angular

الآن سنقوم بعرض template سنقوم باستخدامه في دورة ال .. أيضا وبالإضافة الى الملفات التي قمنا بتحميلها، أريدك أن تقوم بإنشاء app.js و style.css كما في الصورة ..



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>

  <body>
    <!-- Body Code -->

    <script type="text/javascript" src="js/angular.min.js"></script>
    <script type="text/javascript" src="js/app.js"></script>
  </body>
</html>
```

أسسیات ال Angular

تعريفات مهمة قبل البدء: (تعريفات مبدئية لحين الوصول الى كل عنوان)
يوجد عدة تعريفات مهمة يجب عليك ان تعرفها قبل الخوض او البدء
بشرح أي مفهوم بال angular وهي:

(١) Template: ويقصد بها ال HTML مع الخصائص
المضافة ...

(٢) Directives: ويقصد بها عناصر ال HTML مضافا اليها
...custom attributes and elements

(٣) Model: ويقصد بها البيانات التي ستعرض للمستخدم من خلال ال
view الخاصة بها، بناءا على ما التفاعل بينها وبين المستخدم.

(٤) Scope: وهو مكان حفظ ال model والذي يحدد النطاق الذي يمكن
الوصول اليه من قبل كل من controllers, directives and
expressions

أسسیات ال Angular

٥) Expressions: تسمح لك بالوصول الى المتغيرات والدوال ضمن النطاق الخاص بال model، وذلك يعطينا القدرة على اضافة قيم .. html dynamic

٦) Filter: هذه تتيح لنا التحكم بشكل القيمة المراد طباعتها للمستخدم (بمعنى آخر ..Format for Expressions).

٧) View: وهو ما يشاهده المستخدم (DOM).

٨) Data Binding: هذا المصطلح يشير الى عملية تبادل في الوقت الحقيقي (بشكل متزامن) بين كل من ال view وال model

٩) Controller: يقوم بالتحكم بالتطبيق من خلال الربط المنطقي بين ال view وال model ..

أساسيات ال Angular

١٠) Module : هو الحاوي لأجزاء عدة مختلفة من المشروع مثل ...controllers, services, filters, directives

ملاحظة: تعني ال ng في المشروع (التفاصيل لاحقا). الآن لنبدأ على بركة الله في Angular Directive مثال بسيط في ال ..angular

```
<!DOCTYPE html>
<html ng-app=""> ← Directives
  <head>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>

  <body class="ex-1">
    <!-- Body Code -->
    Name: {{ "Anees" + " Hikmat" }} <br />
    Age: {{ 10 + 14 }} ← Expressions
    <script type="text/javascript" src="js/angular.min.js"></script>
    <script type="text/javascript" src="js/app.js"></script>
  </body>
</html>
```

Name: Anees Hikmat
Age: 24

الناتج

أسسیات ال Angular

شرح المثال السابق:

(١) ng-app: حسب التعريفات التي طرحاها ..نكتشف أن هذه الخاصية عبارة عن Directive ..وظيفة هذا ال directive هي تعريف تطبيق ال angular.. وهذا مهم لتطبيق أي Expressions، وبدونه لن يعمل، بالإضافة إلى ذلك يمكننا من اسناد قيمة، هذه القيمة تمثل اسم ال modules الرئيسي للمشروع.. وسنطرق لمثال على هذه لاحقا بإذن الله تعالى ..

(٢) ..هذه تستخدم لتعريف ال Expressions .. وهو أول موضوع سنقوم بالحديث عنه بإذن الله تعالى .. والآن لنبدأ معا حيز العمل الفعلي ..

شاهد المثال

Expressions

إن ال Angular يقوم على مزامنة البيانات من والى ال html .. ولهذا فهو بالأهمية بمكان لـ Expressions خلال

قواعد ال Expressions :

- (١) يجب عليك كتابة ال Expressions داخل ال {{ }}
- (٢) تقوم بعملية مزامنة البيانات أيضاً، ويمكن استخدامها مثل ال ng-bind Expressions
- (٣) سيتم طباعة النتائج كما هي تماماً في ال ..Expressions
- (٤) ال Expressions في ال Angular تشبه ال Expressions في الجافا سكربت من حيث القدرة على احتواء أرقام وأحرف و العمليات المختلفة ... مثل (“anees” + “hikmat”) ..

Expressions

مثال :

```
<body class="ex-2">
  <!-- Body Code -->
  Name: {{ "Anees" + " Hikmat" }} <br />
  Age: {{ 10 + 14 }}

  <div ng-bind="'Anees' + ' Hikmat'"></div>
  <div ng-bind="10 + 14"></div>

<script type="text/javascript" src="js/angular.js">
```

Name: Anees Hikmat

Age: 24

Anees Hikmat

24

Expressions

مثال ٢:

ملاحظة: ng-init تستخدم لإعطاء قيمة ابتدائية.

```
<div ng-init="faceStyle = '^_^'; id=1">
  Face: {{ faceStyle + " :P " + id }}
  <br />
  Face: <span ng-bind="faceStyle + ' :P ' + id"></span>
</div>
```

Face: ^_^ :P 1

Face: ^_^ :P 1

Expressions

مثال ٣:

ال Expressions في ال Angular يمكنها من استدعاء القيم من ال objects أو من المصفوفات .. وهذا أمر مهم .. لأننا في أغلب الأوقات سنتعامل مع JSON .. وال JSON في الغالب سيكون إما objects أو مصفوفات أو خليط بينهما .. لذلك سنورد مثلاً أيضاً على هذه الملاحظة المهمة ..

```
<div ng-init="info={sort:1, cost:50}; rank=[1,2,3,4,5]">
    Face Object: {{ info.cost }}$ object
    <br />
    Face Array: {{ rank[4] }} array
    <br />
</div>
```

Face Object: 50\$
Face Array: 5

شاهد المثال

عن أبي هريرة رضي الله عنه أن رسول الله - صلى الله عليه وسلم - قال :
ـ (قال الله : أَنْفَقَ يَا ابْنَ آدَمَ أَنْفَقَ عَلَيْكَ) .

Directives

تحدثنا فيما سبق أن أي attribute تسبق بـ ng فهذه تعني أو تدل على angular .. وبما أن الفكرة أصبحت أوضح قليلاً الآن، فإن هذه الـ ng تخبرنا بالتحديد أن ما نراه هو directive ^_^.. directive المختلفة والمهمة، وتحدثنا بكل تأكيد هناك العديد من الـ directive عن بعضها في الشرائح السابقة، وسنضيف إليها أيضاً الآن وفي الأمام وبعد هذا الموضوع العديد من الـ Directive بإذن الله تعالى... والآن سنذكر إحدى أهم الـ directive .. وهو الـ ng-model.. هذا الـ directive يقوم على عمل data bind (تزامن للبيانات) مع عناصر التحكم مثل الـ input text والـ text-area .. الخ

Directives

مثال ١: (لاحظ هنا كيف يتم أخذ القيمة وتحديثها مباشرة من خلال الـ Expression Directive)

```
<div ng-init="firstName='Anees';lastName='Abu-hmaid';ageName=24">
  <p>First Name: <input type="text" ng-model="firstName"></p>
  <p>You wrote: {{ firstName }}</p>

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

  <p>Age: <input type="text" ng-model="ageName"></p>
  <p>You wrote: {{ ageName }}</p>
</div>
```

First Name: Anees :P^_*

You wrote: Anees :P^_*

Last Name: Abu-hmaid ^_^

You wrote: Abu-hmaid ^_^

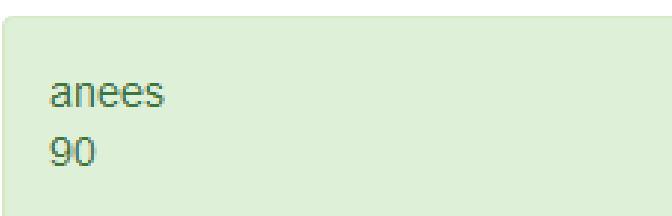
Age: 25

You wrote: 25

Directives

مثال ٢: هناك أيضا directive رائع وهو `ng-repeat`، ووظيفته هي القيام بعمل دوران ... شاهد المثال:

```
<div ng-init="students=[{name: 'anees', mark: 90}, {name: 'taher', mark: 95}]">
  <div class="alert alert-success" ng-repeat="items in students">
    <div>{{ items.name }}</div>
    <div>{{ items.mark }}</div>
  </div>
</div>
```



يمثل ال `items` هنا المعلومات المراد جلبها على شكل `object`.

شاهد المثال

Directives

ملاحظات مهمة:

- (١) قلنا أنها تستخدم لإسناد القيم الإبتدائية للعناصر، لكن هذا Directive لا يكثر من استخدامه عادة، وإنما نستخدم الـ controllers لإعطاء أو إسناد القيم الإبتدائية ..(في غالب الأمر)
- (٢) هناك العديد من الـ directive الأخرى .. يمكنك الإطلاع عليها من المراجع التي أرفقتها سابقا .. وسيتم بإذن الله تعالى ذكر عدد لا يأس به قدر الإمكان بما ينساب هذه الدورة ... وتم إدراج هذا الموضوع كمقدمة للـ module والـ controllers ..

<https://docs.angularjs.org/guide/directive>

Modules

تحدثنا عن ال module في الشرائح السابقة أيضا، كما أكرر لك ذلك لأهمية الموضوع وبأسلوب آخر، إن أول ما تفكر به عن النظر لل modules أنها طريقة تستخدم في ال angular لضم أو إحتواء مجموعة مترابطة من الشيفرات البرمجية موجودة تحت عنوان أو اسم واحد، هو الإسم الخاص بال module ^_ ^، ولذلك يشاهد الآخرين الشيفرة البرمجية الخاصة بك بسهولة، ويكون فهمها أسهل وأسرع ^_ ^ و هناك جزئيتان مهمتان حول ال module وهي

(١) أنه يمكننا تعريف ال controller وال services .. module لكل

(٢) يمكننا أيضا انشاء module أخرى تعتمد عليها ال module الذي قمنا ببنائه .. module can also depend on other modules as dependencies

Modules

أعتذر عن إطالة الكلام في الشريحة السابقة .. لكن لأهمية الموضوع وستجد أن هذا الكلام الكثير، ما هو إلى أمر بسيط جدا عند التنفيذ ^_*

والآن كيف يتم تحميل أو الاتصال بال module ؟
يكون ذلك من خلال ال ng-app .. وال فكرة هي ارسال **الإسم الخاص**
بال module إلى ال directive ng-app ...
ولتصبح الصورة أوضح .. شاهد كيف يمكننا إنشاء .. module

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

متغير جافا سكريبت

للدلالة على استخدام ال
AngularJS

تحديد أننا سنبني
module

اسم التطبيق
(اسم ال
(module

هذه ما أطلقنا عليها ال
Dependencies في الشريحة
السابقة .. هنا نضع ال
 الأخرى التي تعتمد على
 الذي ثمنا ببنائه ...

Modules

الآن .. اذا عدت الى ال template الذي قلت بأنني سأستخدمه .. ستجد هناك ملف ملف app.js .. هذا الملف سنقوم بوضع ال module بداخله على الشكل التالي:

```
[function() {
    var app = angular.module('store', []);
}]();
```

والآن .. ماذا بعد؟
يجب علينا أن نخبر ال ng-app بوجود اسمه store module حتى يستطيع الإتصال به و مناداته، و قلنا ذلك يتم عن طريق استخدام ال ng-app="store" .. شاهد المثال: module

Modules

شكل الملف الآن: (وبهذا نكون انتهينا من شرح طريقة انشاء وبناء الـ module داخل الصفحات) هل لاحظت سهولة الموضوع ..
المهم أن تقوم بالتطبيق معك الآن أولا بأول .. لأن الإضافات ستبدأ من

هنا ... ^_*

```
<!DOCTYPE html>
]<html ng-app="store">
  <head>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>

  <body class="ex-4">          HTML PAGE
    <!-- Body Code -->

    <script type="text/javascript" src="js/angular.min.js"></script>
    <script type="text/javascript" src="js/app.js"></script>
  </body>
</html>
```

```
(function() {           APP.js PAGE
  var app = angular.module('store', []);
})();
```

لا يكون السروز دائماً إلا جديداً على النفس ، ولا سروز للنفس إلا من جديد
على حالة من أحوالها؛ فلو لم يكن في كل دينار قوة جديدة غير التي في مثله لما
سرّ بالمال أحد، ولا كان له الخطر الذي هو له؛ ولو لم يكن لكل طعام جوع
يورده جديداً على المعدة لما هنأ ولا مرأ؛ ولو لم يكن الليل بعد نهار ، والنهر بعد
ليل ، والفصول كلها نقضاً على نقبيه ، و شيئاً مختلفاً على شيء مختلف - لـما كان
في السماء والأرض جمال ، ولا منظر جمال ، ولا إحساس بهما؛ والطبيعة التي لا
تُفلح في جعلك معها طفلاً تكون جديداً على نفسك - لن تُفلح في جعلك مسروراً
بها لتكون هي جديدة عليك .

من كتب وحي الله - الجزء الأول

Controller

لقد أجزلنا في الحديث عن ال controller أيضا .. لكن العجيب في الموضوع، أنني لم أتحدث أو أطرق إلى أي مثال إلى الآن، وهذا صحيح، لأنني كنت أنتظر الإنتهاء من شرح طريقة تعريف ال module حتى يتسع لنا فهم الموضوع بشكله الصحيح ^_ ^، والآن اذا لاحظت ال module السابق، فستجد أنه فارغ.... ولذلك يجب علينا ملئه حتى نستفيد من وجوده .. وسنملأه الآن .. بإحدى أهم المكونات وهو ال Controller ^_ ^ ..

فهذا ال Controller هو الذي يجلب البيانات من السيرفر وغيرها ومن ثم يفصلها ويعرضها للمستخدم عن طريق ال view، وهو الذي يحدد أي من البيانات التي يجب أن تظهر للمستخدم، كما أنه المسؤول عن أي تفاعل يحصل بين المستخدم وال view لتنفيذ أي أمر ^_ ^ ..

Controller

والآن .. لنقوم بإنشاء Controller

```
](function() {
    // Declare Angular Module
    var app = angular.module('store', []);

    // Declare Controller
    app.controller('StoreController', function() {
        // Code
    });
});
```

الإسم الخاص بـ controller
لاحظ أنني استخدمت أحد المعايير البرمجية في التسمية .. (وهناك أخرى)

app.js
قمنا الآن بإنشاء ال controller

لاحظ سهولة تعريف ال controller، ولا يلاحظ أنني عرفته باستخدام ال app وهذا ال app هو المتغير الذي يحفظ ال object الخاص بال module .. هذه الطريقة أراها أسهل وأرتب في كتابة الشيفرة البرمجية، ولذلك أيضاً تحتويت الجافا سكريبت كود بـ *_^ (function()...)

Controller

والآن سأقوم بتعريف منتج سيتم عرضه في المتجر (store)، وسنشاهد الآن، كيف دور ال controller الرائع ... شاهد المثال: (سنقوم بارجاع القيم عن طريق ال controller من خلال ال (product الذي جعلنا قيمته الراجعة هي ال function

```
// Declare Controller
app.controller('StoreController', function() {
    this.product = productItem;
});

var productItem = {
    'name': 'angularJs Course',
    'desc': '2nees.com - Free courses',
    'cost': '0$'
};
```

قمت بتعريف متغير من نوع object



Controller

الآن ..كيف سنستطيع الوصول الى القيمة السابقة وعرضها من خلال
ال html ؟ .. اذا كنت مصرا على الإجابة ..فشاهد الصورة التالية:
(لا تنسى تطبيق وكتابة ما يتم عرضه بالصور لنبقى معا أثناء التنفيذ)

```
<div ng-controller="StoreController as store">
  <div class="row">
    <label class="label label-default col-xs-1">Name</label>
    <span class="col-xs-offset-1">{{store.product.name}}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-1">Description</label>
    <span class="col-xs-offset-1">{{store.product.desc}}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-1">Cost</label>
    <span class="col-xs-offset-1">{{store.product.cost}}</span>
  </div>
</div>
```

Controller

شرح الصورة السابقة:

(١) Controller هو اسم ال StoreController الذي قمنا بانشائه داخل ال module store

(٢) هنا قمنا بإنشاء store على أنه اسم مرادف لل as store .. وبهذا يمكنني التعامل مع الإسم store وكأنه StoreController .. وهذا أسهل لتنفيذ الشيفرات البرمجية .. StoreController

ملاحظة مهمة: لن يعمل ال StoreController الا ضمن نطاق البدائية والنهائية لل div الخاص به .. لذلك لو قمنا بوضع ال {{store.product.cost}} كمثال خارج هذا ال div .. فلن يطبع شيء .. لكن ماذا لو أننا لم نضع أيضا اسم ال module ^ ^

Controller

```
<html ng-app="store">
<head>
  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
  <link rel="stylesheet" type="text/css" href="css/style.css" />
</head>

<body class="ex-4 container">
  {{store.product.name}} {{store.product.desc}} {{store.product.cost}}
  <div ng-controller="StoreController as store">
    <div class="row">
      <label class="label label-default col-xs-1">Name</label>
      <span class="col-xs-offset-1">{{store.product.name}}</span>
    </div>
    <div class="row">
      <label class="label label-default col-...>
      <span class="col-xs-offset-1">{{store.product.desc}}</span>
    </div>
    <div class="row">
      <label class="label label-default col-...>
      <span class="col-xs-offset-1">{{store.product.cost}}</span>
    </div>
  </div>
  {{store.product.name}} {{store.product.desc}} {{store.product.cost}}
<script type="text/javascript" src="js/angular.min.js"></script>
<script type="text/javascript" src="js/app.js"></script>
</body>
</html>
```

مثال في حالة وجود ال
store module in ng-app
وحاولنا جلب القيمة من خارج ال
controller

Name
Description
Cost

angularJs Course
2nees.com - Free courses
0\$

الناتج

Controller

```
<html ng-app="":>
  <head>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>
  لاحظ هنا أنتي لم أقم باستدعاء ال
  module
<body class="ex-4 container">
  الذي يحتوي على
  controller
    {{store.product.name}} {{store.product.desc}} {{store.product.cost}}
  <div ng-controller="StoreController as store">
    <div class="row">
      <label class="label label-default col-xs-1">Name</label>
      <span class="col-xs-offset-1">{{store.product.name}}</span>
    </div>
    <div class="row">
      <label class="label label-default col-xs-1">Description</label>
      <span class="col-xs-offset-1">{{store.product.desc}}</span>
    </div>
    <div class="row">
      <label class="label label-default col-xs-1">Cost</label>
      <span class="col-xs-offset-1">{{store.product.cost}}</span>
    </div>
  </div>
  {{store.product.name}} {{store.product.desc}} {{store.product.cost}}
<script type="text/javascript" src="js/angular.min.js"></script>
<script type="text/javascript" src="js/app.js"></script>
</body>
</html>
```

الذي يحتوي على
controller

انتبه جيدا للنتيجة ^_^

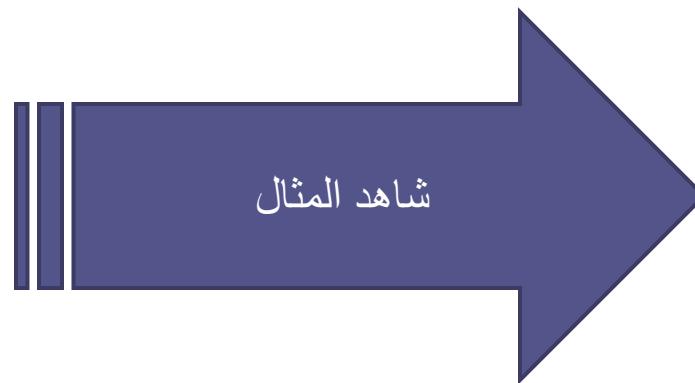
Name	{{store.product.name}}
Description	{{store.product.desc}}
Cost	{{store.product.cost}}

{{store.product.name}} {{store.product.desc}} {{store.product.cost}}

Controller

بكل .. تأكيد يمكنك تجربة العديد من الإحتمالات .. لكنني تطرقـت
لإحتمالات قد يكثر الخطأ في الواقع بها ^_^ ..

شاهد المثال: (وتأكد من أنك تستطيع تطبيق وكتابة الشيفرة البرمجية
قبل الإنقال للمستويات التي تليها ..)



وبعد ذلك انتقل للشريحة التالية لشرح نقطة خطيرة ^_^ ...

Controller

شاهد

هذه

الملاحظة

الخطيرة:

انتبه لطريقة تعريف
الرسائل داخل الـ
Controller

```
<div ng-controller="aneesMessageController as aneesMsg">
  <div class="row">
    <label class="label label-default col-xs-2">Welcome Message</label>
    <span class="col-xs-offset-1">{{aneesMsg.hi}}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-2">Good By Message</label>
    <span class="col-xs-offset-1">{{aneesMsg.bye}}</span>
  </div>
</div>
```

HTML

```
app.controller('aneesMessageController', function(){
  this.hi = "Hi Anees in Angular";
  var bye = "Good by Anees!";
});
```

app.js



Hi Anees in Angular

outPut
لاحظ عدم وجود أي خطأ ^_^

ولاحظ عدم ظهور الـ
bye

Controller

لماذا لم تظهر الرسالة السابقة؟

لنعيد تسلسل الأحداث حتى نصل الى النتيجة المطلوبة ^ ^

(١) قمنا بوضع module store وتحتوي على اسم ال ng-app (لم يتم اظهارها بالصفحة السابقة لغايات استغلال حجم الصفحة فقط)

(٢) قمنا بإنشاء controller بعنوان .. aneesMessageController

(٣) ثم قمنا بوضع رسالتين، الأولى عرفت من خلال استخدام ال this والآخر من خلال استخدام ال var على أنها local variable

(٤) قمنا بجلب هذه القيم من خلال ال html عن طريق ال (bye).. var local controller لكن ال لم يطبع

Controller

اذن ... من خلال التتبع السابق.. نصل لنتيجة مهمة، وهي أن ال inner variable أو ال Local variable لا يمكن الوصول إليها بالطريقة هذه، ولذلك، ولنستطيع الوصول إلى المتغيرات أو القيم، يجب أن نستخدم الكلمة المحوزة this، وهذا هي تعلم عمل ال var من حيث المبدأ، أي أنها تقوم بتعريف متغير، الا أنها تعطي صلاحية للوصول إليها ^ ^

: ^ ^ والآن شاهد المثال عمليا ^

شاهد المثال

لحظة تفكير

ذو العقل يشقى بالنعم بعقله وأخوه الجهمة في الشقاوة ينعم

أبو الطيب المتنبي

Controller

الآن .. ماذا لو أردنا استرجاع القيم من مصفوفة تحتوي مجموعة من ال objects

طبعا، وبكل تأكيد يمكن ذلك من خلال الشكل التالي ^_ ^
 (قم بتنفيذ الشيفرة البرمجية وانظر النتائج قبل استكمالك للشرايح ^_ ^)

```

<div ng-controller="aneesCoursesController as aneesCourses">
  <div class="row">
    <label class="label label-default col-xs-2">Courses Name</label>
    <span class="col-xs-offset-1">{{ aneesCourses.courses[0].name }}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-2">Courses Name</label>
    <span class="col-xs-offset-1">{{ aneesCourses.courses[1].name }}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-2">Courses Name</label>
    <span class="col-xs-offset-1">{{ aneesCourses.courses[2].name }}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-2">Courses Name</label>
    <span class="col-xs-offset-1">{{ aneesCourses.courses[3].name }}</span>
  </div>
</div>

```

Courses Name	AngularJS
Courses Name	jQuery
Courses Name	PHP Advance
Courses Name	CSS3

```

app.controller('aneesCoursesController', function(){
  this.courses = coursesItem;
});

var coursesItem = [
  {id:1, name: 'AngularJS', price: 'Free'},
  {id:2, name: 'jQuery', price: 'Free'},
  {id:3, name: 'PHP Advance', price: 'Free'},
  {id:4, name: 'CSS3', price: 'Free'}
];

```

Controller

هل لاحظت الشريحة السابقة والنتائج الخاصة بها ^_ ^ ؟
ما رأيك بطريقة الحل تلك؟

بكل تأكيد الحل السابق سيعمل معك بشكل صحيح، لكن هل من المنطق أن أقوم بتكرار أو استدعاء القيم من المصفوفة واحدة واحدة ؟
طبعا لا، وهنا يظهر أيضا جمال آخر وجديد من ال Angular وهو وجود خاصية تسمح لك باستدعاء جميع القيم عن طريق الدوران ^_ ^ وإن سألتني كيف ذلك ^_ ^ .. سأخبرك أنه عليك استخدام directive واحد وبسيط وهو ^_ ^ * .. ng-repeat لشاهد مثلا ^_ ^

Controller

مثال: لاحظ هنا ”...“ اسم المتغير الخاص بالقيمة عند الدوران..
ال course Loop هذه تمثل اسم ال Alias in aneesCourse.courses
بالإضافة الى اسم المتغير الذي يحتوي مجموعة
القيم في ملف ال app.js .. (this.courses)..
(هذا نفس المثال السابق.. لكن باستخدام ال ng-repeat)

```
<div ng-controller="aneesCoursesController as aneesCourses">
  <div class="row" ng-repeat="courseLoop in aneesCourses.courses">
    <label class="label label-default col-xs-2">Courses Name</label>
    <span class="col-xs-offset-1">{{courseLoop.name}}</span>
  </div>
</div>
```

شاهد المثال

Controller

الآن .. ما رأيك أن نقوم بإضافة زر لتحميل الدورات السابقة ..
طبعا سيكون الزر -وهي P- لغایات الشرح:
ما أريده أيضا بالضبط هو .. اظهار زر التحميل للدورات المسموح لي
بتحميلها فقط .. أي أن الزر لن يكون ظاهرا لبعض الدورات ..
والسؤال الآن .. كيف هذا؟ والجواب هو * * ng-show

```
<div ng-controller="aneesCoursesController as aneesCourses">
  <div class="row" ng-repeat="courseLoop in aneesCourses.courses">
    <label class="label label-default col-xs-2">Courses Name</label>
    <button class="col-xs-offset-1" ng-show="courseLoop.here">Download</button>
  </div>
</div>
```

شاهد المثال:

```
var coursesItem = [
  {id:1, name: 'AngularJS', price: 'Free', here: true},
  {id:2, name: 'jQuery', price: 'Free', here: false},
  {id:3, name: 'PHP Advance', price: 'Free', here: false},
  {id:4, name: 'CSS3', price: 'Free', here: true},
];
```

Controller

في المقابل للخاصية السابقة، يوجد لدينا `ng-hide` .. وبنفس الأسلوب في حالة كانت الـ `ng-hide` تساوي `true` فهذا يعني أنها ستكون مخفية .. والآن لاستعراض مثلاً سريعاً: (قم بتطبيقه لتصل لنتيجة)

```
<div class="row" ng-repeat="courseLoop in aneesCourses.courses">
    <label class="label label-default col-xs-2">Courses Name</label>
    <button class="col-xs-offset-1" ng-hide="courseLoop.here">Downlaod
</div>
<hr />
<div class="row" ng-repeat="courseLoop in aneesCourses.courses">
    <label class="label label-default col-xs-2">Courses Name</label>
    <button class="col-xs-offset-1" ng-show="!courseLoop.here">Downlaod
</div>
<hr />
<div class="row" ng-repeat="courseLoop in aneesCourses.courses">
    <label class="label label-default col-xs-2">Courses Name</label>
    <button class="col-xs-offset-1" ng-hide="!courseLoop.here">Downlaod
</div>
```

Controller

٨.. اذا قمت بتطبيق المثال السابق .. ما رأيك الان أن تقوم باستخدام بعض المتغيرات الموجودة والتي يمكننا استخدامها مع ال ng-repeat

٨_٨
...
booleann و \$middle و \$first هذه ٣ متغيرات نوعها boolean first لتحديد اذا كان العنصر أول العناصر وال middle جميع العناصر بين البداية والنهاية وال last في نهاية العناصر *

٨_٨
\$index يقوم على ارجاع موقع القيمة ...
\$odd و \$even تقوم على فحص هل العنصر فردي أم زوجي ..
٨_٨
تخيل مقدار الفائدة المتحققة
٨٨
بكل تأكيد أن true تعني يحقق الشرط و false أنه لا يحقق الشرط

Controller

مثال: (قم بتطبيقه ثم شاهد المثال)

```
<div class="row" ng-repeat="courseLoop in aneesCourses.courses">
  <div>First Element: {{ $first }}</div>
  <div>Middle Element: {{ $middle }}</div>
  <div>Last Element: {{ $last }}</div>
  <div>Index of Element: {{ $index }}</div>
  <div>At Even Position: {{ $even }}</div>
  <div>At Odd Position: {{ $odd }}</div>
  <hr />
</div>
```



شاهد المثال

Controller

الآن .. هناك أيضا حركة جميلة يمكننا من التعامل معها واستخدامها من خلال ال repeat .. وتكون هذه الخاصية في امكانية تكرار عناصر ال html أثناء الدوران ..
ويكون هذا باستخدام ال ng-repeat-start="" وال ... ng-repeat-end

```
<table class="table table-striped">
<tr ng-repeat-start="courseLoop in aneesCourses.courses">
  <td>{{courseLoop.id}}</td>
  <td>{{courseLoop.name}}</td>
  <td>{{courseLoop.price}}</td>
</tr>
<tr ng-repeat-end>
  <td>{{$odd}}</td>
  <td>{{$last}}</td>
  <td>{{$index}}</td>
</tr>
</table>
```

1	AngularJS	Free
false	false	0
2	jQuery	Free
true	false	1
3	PHP Advance	Free
false	false	2
4	CSS3	Free

Controller

مم...مماذا بعد `^_^`، هناك فكرة رائعة ما رأيك أن نقوم بتطبيقها معا؟
هناك `event` اسمه `directive` .. وهذا يمثل `^_^ event`
الآن ..لنشاهد مثلاً على هذا ال

```
<button class="btn btn-info" ng-init="count=1;" ng-click="count = count + 1">
    Increase Count {{ count }}
</button>
```

هل لاحظت سهولة الفكرة `^_^` ..
لكن هذا التنفيذ المباشر لل `ng-click` .. لكن ماذا لو أردنا إرسال أو
استقبال قيم من خلال ال `app.js` كما كنا نتعامل مع المتغيرات ؟

Controller

وللإجابة على السؤال السابق.. شاهد الصورة التالية:
لكي نأخذ القيمة من function .. قمنا باستخدام ال this وذلك لأجل
ال ال scope .. وبهذا أستطيع الوصول للقيم وارسالها واستقبالها ..

```
app.controller('aneesCoursesController', function() {
    this.courses = coursesItem;

    this.toggleCourse = function() {
        for(var i = 0; i < coursesItem.length; i++){
            this.courses[i].here = !coursesItem[i].here;
        }
    };
});

var coursesItem = [
    {id:1, name: 'AngularJS', price: 'Free', here: true},
    {id:2, name: 'jQuery', price: 'Free', here: false},
    {id:3, name: 'PHP Advance', price: 'Free', here: false},
    {id:4, name: 'CSS3', price: 'Free', here: true},
];
```

فكرة الفانكتشن هي عملية قلب القيمة من
true to false & false to true

والآن لنشاهد المثال
بشكل عملي وانتبه
لكل جزئية ^_^

شاهد المثال

عَنْ أَبِي هُرَيْرَةَ رَضِيَ اللَّهُ عَنْهُ، قَالَ : قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ :
كَلِمَتَانِ خَفِيفَتَانِ عَلَى الْلِّسَانِ ثَقِيلَتَانِ فِي الْمِيزَانِ
حَبِيبَتَانِ إِلَى الرَّحْمَنِ : سُبْحَانَ اللَّهِ وَبِحَمْدِهِ سُبْحَانَ اللَّهِ الْعَظِيمِ " .

كتابٌ صحيحٌ مُلْكٌ حَلَّيْهِ

Filters

السؤال البديهي .. ^_٨.. ما هو ال AngularJS Filters بكل بساطة هي مزية تستخدم لمعالجة البيانات أو شكل البيانات ليتم عرضها للمستخدم بشكل سهل القراءة والفهم، ويتم تطبيق هذه ال filters داخل ال Expressions or controller ... وعادة ما تكون هي المرحلة النهائية لشكل البيانات الناتجة للمستخدم.. ومن الأمثلة عليها الفلتر الخاص بصيغة الوقت أو العملة .. الخ

أيضا من مزايا الفلتر أنه يوفر علينا تخزين البيانات بأشكال متعددة، فلو كان لدينا محتوى dynamic .. فإنه يكفي أن نضع الفلتر داخل ال view وسيتم عرض البيانات هذه كما تم بالفلتر .. وهذا يجعل أمر تعديل شكل البيانات أسهل أيضا .. ^_٨ .. *

Filters

ملاحظة: يمكننا استخدام أحد الـ Filters الموجودة أو المبنية أصلاً في الـ AngularJS أو يمكننا بناء الـ Filter الخاص بك ^_^

الـ Filters الموجودة هي:

(1) Currency: يقوم هذا الفلتر على جعل الرقم على شكل عملة (أي كأننا نتعامل مع العملة مثل \$1,234.56) ويكون الرمز الخاص بالعملة هو الرمز الخاص باللغة المستخدمة محلياً (الحالة الإفتراضية) ...

والآن لنشاهد مثلاً على هذا الفلتر

Filters

الصيغة العامة:

```
 {{ currency_expression | currency : symbol : fractionSize}}
```

شاهد المثال: (ال symbol اختياري وال fractionSize اختياري..)

```
<div ng-controller="aneesFilters as filter">
  <div ng-repeat="item in filter.filterItem">
    <div>{{item.price | currency}}</div>
    <div>{{item.price | currency:"JOD "}}</div>
    <div>{{item.price | currency:"JOD ":9}}</div>
    <hr />
  </div>
</div>
```

القيمة التي ستغير نوع الفلتر رمز العملة عدد الخانات

```
app.controller('aneesFilters', function(){
  this.filterItem = filterItems;
});

var filterItems = [
  {price: 1254},
  {price: 154},
  {price: 963},
  {price: 1880},
];

$1,254.00
JOD 1,254.00
JOD 1,254.0000000000

$154.00
JOD 154.00
JOD 154.0000000000

$963.00
JOD 963.00
JOD 963.0000000000

$1,880.00
JOD 1,880.00
JOD 1,880.0000000000
```

Filters

(٢) number: يقوم هذا الفلتر على جعل النص الرقمي بصيغة رقم يحتوي فوائل تفصل بين الفئات، كما يمكن تحديد عدد الخانات الخاصة بالأعشار .. و اذا لم تكن القيمة المسندة للفلتر رقما ..سيتم ارجاع (أي لا شيء..وستصبح المخرجات فارغة) .. و إذا استخدمنا كلمة Infinity أو -Infinity فسيتم ارجاع الإشارة الخاصة بالمالانهاية ..

الصيغة العامة:

```
 {{ number_expression | number : fractionSize }}
```

Filters

```
<div ng-repeat="item in filter.filterItem">
  <div>{{item.count | number}}</div>
  <div>{{item.count | number:2}}</div>
  <div>{{-item.count | number:6}}</div>
  <hr />
</div>
```

سلب القيمة

100
100.00
-100.000000
-2,112
-2,112.00
2,112.000000
0
0.00
0.000000
66,655,412
66,655,412.00
-66,655,412.000000

```
var filterItems = [
  {price: 1254, count: 100},
  {price: 963, count: -2112},
  {price: 1880, count: 0},
  {price: 153, count: 66655412},
];
```

شاهد المثال:

بخصوص الملاينهية .. فإنها لن تعمل على ال local الى في حال قمت ببعض التغييرات .. يمكنك الإطلاع على المزيد [من هنا](#) أو يمكنك رؤية المثال وتطبيقه مباشرة [من هنا](#)...

Filters

أسهل فلتر هههه يقوم هذا ^_ ^ :uppercase & lowercase (³
الفلتر على تغيير حالة الأحرف إما الى uppercase أو الى
*_^ ... lowercase

الصيغة العامة:

```
 {{ uppercase_expression | uppercase}}}
```

```
 {{ lowercase_expression | lowercase}}}
```

UpperCase

ANEES

HIJAWI

```
<h2>UpperCase</h2>
<div ng-repeat="item in filter.filterItem">
    <div>{{item.name | uppercase}}</div>
    <hr />
</div>
```

MALEK

SADDAM

LowerCase

anees

```
<h2>LowerCase</h2>
<div ng-repeat="item in filter.filterItem">
    <div>{{item.name | lowercase}}</div>
    <hr />
</div>
```

hijawi

malek

saddam

```
var filterItems = [
    {price: 1254, count: 100, name: "Anees"} ,
    {price: 963, count: -2112, name: "Hijiawi"} ,
    {price: 1880, count: 0, name: "Malek"} ,
    {price: 153, count: 66655412, name: "Saddam"} ,
];
```

Filters

مثال:

Filters

٤) date : سهلة أيضا .. نستخدم هذا الفلتر لتغيير الشكل الخاص بالتاريخ .. والصيغة العامة له هي:

```
{[ date_expression | date : format : timezone ]}
```

الحالة الإفتراضية هي mediumDate في حال عدم تحديد شكل التاريخ المراد ..

Date تمثل نص أو رقم يعبر عن أجزاء الثانية أو عن تاريخ معين، بمعنى آخر، إما أن يعتمد أحد الأنظمة الخاصة بالوقت مثل ال ISO أو يتم وضع التاريخ المراد بنائا على عدد أجزاء الثواني ..

Filters

Oct 29, 2010

Friday, October 29, 2010

29/10/2010 at 5:40AM

Oct 26, 2010

Tuesday, October 26, 2010

26/10/2010 at 11:07PM

Nov 10, 1991

Sunday, November 10, 1991

10/11/1991 at 10:11AM

Dec 5, 1992

Saturday, December 5, 1992

05/12/1992 at 6:11PM

مثال:

```
<div ng-repeat="item in filter.filterItem">
  <div>{{item.date | date}}</div>
  <div>{{item.date | date : 'fullDate'}}</div>
  <div>{{item.date | date : "dd/MM/yyyy 'at' h:mma"}}</div>
  <hr />
</div>
```

```
var filterItems = [
  {price: 1254, count: 100, name: "Anees", date: '1288323623006'},
  {price: 963, count: -2112, name: "Hijawi", date: '1288323623006'},
  {price: 1880, count: 0, name: "Malek", date: '1991-11-10T10:11'},
  {price: 153, count: 66655412, name: "Saddam", date: '1992-12-05T18:11'},
];
```

يمكنك مشاهدة صيغ الوقت المختلفة من هنا.

إلهي أنت حسبي وأنت ربى
وما للقلب من عشق سواك
ايها من فطرتني وملكت نفسي
وجعلت جوارحي تهوى لقاك
ارحم فؤاد عشيق بات يبكي
ذنوبها تحول دون رضاك
ولا تجعل هموم الدنيا همي
وقربني بذكرك من حماك
وبيوم حشرك يا ربى أجرنى
واسدل على ثوب رضاك
وما دعوت وما طلبت الا
مقام الأقربين بأن اراك

لحد أحضنه مجموعة ملائكة

Filters

: تستخدم هذه المزية لترتيب قيم مصفوفة معينة إن كانت هذه المصفوفة ذات قيم نصية أو رقمية .. يكون الترتيب النصي أبجدياً .. ^{*} والصيغة العامة هي:

```
 {{ orderBy_expression | orderBy : expression : reverse}}}
```

ال reverse هي مزية اختيارية تسمح لنا من اعادة الترتيب بشكل عكسي، يعني ترتيب من Z - A عكسي يصبح من A - Z .. القيمة الخاصة به منطقية يعني (true or false)

Filters

مثال :

```
var filterItems = [
    {price: 1254, count: 100, name: "Anees"},
    {price: 963, count: -2112, name: "Hijawi"},
    {price: 1880, count: 0, name: "Malek"},
    {price: 153, count: 66655412, name: "Saddam"},
];
```

```
<h2>OrderBy price</h2>
<div ng-repeat="item in filter.filterItem | orderBy:'-price'" class="col-xs-3">
    <div>Name: {{item.name}}</div>
    <div>Count: {{item.count}}</div>
    <div>Price: {{item.price}}</div>
    <hr />
</div>
```

الترتيب تنازلي بسبب اشارة السالب
والترتيب يكون بناء على ال
price

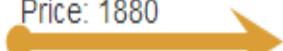
OrderBy price

Name:Malek
Count:0
Price: 1880

Name:Anees
Count:100
Price: 1254

Name:Hijawi
Count:-2112
Price: 963

Name:Saddam
Count:66655412
Price: 153



مثال ٢

Filters

OrderBy by Click

OrderBy Name (Revers Statuses false)

Name:Saddam
Count:66655412
Price: 153

OrderBy Count

Name:Hijawi
Count:-2112
Price: 963

هذا الترتيب يمثل الحالة
الافتراضية التي قمنا بوضعها ..
False & Price

Name:Anees
Count:100
Price: 1254

Name:Malek
Count:0
Price: 1880

```

<h2>OrderBy by Click</h2>
<button class="btn btn-lg" ng-click="selectedItem='name';reversValue=!reversValue">OrderBy Name (Re
<button class="btn btn-lg" ng-click="selectedItem='count'">OrderBy Count</button>
<div class="clearfix"></div>
<div ng-repeat="item in filter.filterItem | orderBy:selectedItem:reversValue" class="col-xs-3">
  <div>Name: {{item.name}}</div>
  <div>Count: {{item.count}}</div>
  <div>Price: {{item.price}}</div>
  <hr />
</div>
  
```

سيكون هنا الترتيب من خلال النقر على أحد الأزرار ^_~
لكن ماذا لو أردنا وضع ترتيب ابتدائي ؟

نعم.. صحيح.. وهناك طريقة أخرى أنظر إليها.. سنقوم بشرحها لاحقاً إن شاء الله ..

```

app.controller('aneesFilters', function($scope) {
  this.filterItem = filterItems;

  $scope.reversValue = false;
  $scope.selectedItem = 'price';
});
  
```

Filters

بهذا نكون انتهينا بإذن الله تعالى من شرح ال orderBy .. لكن كملحظة مهمة وأظنك لاحظتها .. وهي أنني عندما ذكرت الأمثلة والصيغ العامة التي سبقت لم أستخدم السكريبت ولم أطرق اليه والسبب في ذلك يعود الى محاولة تبسيط الأمر حتى الإنتهاء من فهم ال angular كيف يعمل .. وحين الإنتهاء بإذن الله تعالى من آخر جزئية بالفلتر .. سنقوم بطرح مثال يوضح طريقة بناء واستخدام فلتر من خلال الجافا سكريبت .. وسيتم شرح مفهوم مهم مثل ال \$filter وال \$scope .. وغيرها .. لذلك .. دعنا نحاول المسير كما ذكرنا ... وإن استطعت تعلمها أو تطبيقها لوحدك فهذا ممتاز جدا .. وبعد فهم الصيغة العامة وكيف يعمل ال Angular (يفترض أنك تستطيع القيام بذلك من خلال الصغيرة العامة الخالية بال스크ريبت ^_ ^ ..)

Filters

٦) limitTo: تقوم هذه المزية على انشاء نص أو مصفوفة جديدة (نسخة فقط) تحتوي عدد العناصر الذي تم تحديده للاقتطاع أو كأكبر عدد ممكن يسمح لنا بالحصول عليه ..ويمكن البدء بتحديد مكان البدء بالعد حتى يتم النسخ من بداية المصفوفة - أو النص - (موجب) أو من نهاية المصفوفة - أو النص - (اذا كانت القيمة سالبة) ... وليسهل شرح الكلام ..لاحظ الصيغة العامة:

```
 {{ limitTo_expression | limitTo : limit : begin}}}
```

Limit: الحد الأعلى للعناصر..(عدد) و begin: مزية اختيارية قيمتها الإفتراضية ، القيمة السالبة هنا تعني ابدأ من مهایة النص أو المصفوفة

Filters

مثال :

```
var filterItems = [  
  {price: 1254, count: 100, name: "Anees"},  
  {price: 963, count: -2112, name: "Hijawi"},  
  {price: 1880, count: 0, name: "Malek"},  
  {price: 153, count: 66655412, name: "Saddam"},  
];
```

```
<h2>Limit 2</h2>  
<div ng-repeat="item in filter.filterItem | limitTo:2" class="col-xs-3">  
  <div>Name: {{item.name}}</div>  
  <div>Count: {{item.count}}</div>  
  <div>Price: {{item.price}}</div>  
  <hr />  
</div>
```

Limit 2

Name:Anees

Count:100

Price: 1254

Name:Hijawi

Count:-2112

Price: 963

Limit Increase by Click

Filters

Increase Limit

Name:Anees
Count:100
Price: 1254

limit = 1
القيمة الإفتراضية التي قمت بوضعها لهذا المثال
* _____ ^

مثال ٢:

```
app.controller('aneesFilters', function($scope) {
  this.filterItem = filterItems;

  // Initialized value on orderBy click example in 9.html
  $scope.reverseValue = false;
  $scope.selectedItem = 'price';
  $scope.limitCount = 1;

});

<button class="btn btn-lg" ng-click="limitCount = limitCount + 1">Increase Limit</button>
<div class="clearfix"></div>
<div ng-repeat="item in filter.filterItem | limitTo:limitCount" class="col-xs-3">
  <div>Name: {{item.name}}</div>
  <div>Count: {{item.count}}</div>
  <div>Price: {{item.price}}</div>
  <hr />
</div>
```

Filters

(`json`): يسمح لنا هذا الفلتر من القيام بتحويل متغير الجافا سكريبت من نوع `object` الى `JSON string` ..
لماذا يمكن أن أحتاج الى هذا التحويل أصلا؟!
وهو سؤال منطقي جدا .. وإجابته سهلة .. وهي أننا نستخدم هذا الأسلوب غالبا لعمل debugging (فحص القيم وتصحيحها في حالة وجود أي مشكلة) ..
الصيغة العامة:

```
 {{ json_expression | json : spacing}}
```

Filters

مثال:

```
<h2>JSON</h2>
<div class="clearfix"></div>
<div>
    <pre class="col-xs-6">{{ filter.filterItem | json }}</pre>
    <pre class="col-xs-6">{{ filter.filterItem | json:6 }}</pre>
    <hr />
</div>
```

JSON

```
[{
  {
    "price": 1254,
    "count": 100,
    "name": "Anees",
    "date": "1288323623006"
  },
  {
    "price": 963,
    "count": -2112,
    "name": "Hijawi",
    "date": "1288123622006"
  },
  {
    "price": 1880,
    "count": 0,
    "name": "Malek",
    "date": "1991-11-10T10:11"
  },
  {
    "price": 153,
    "count": 66655412,
    "name": "Saddam",
    "date": "1992-12-05T18:11"
  }
]
```

JSON Filter Output

```
[{
  {
    "price": 1254,
    "count": 100,
    "name": "Anees",
    "date": "1288323623006"
  },
  {
    "price": 963,
    "count": -2112,
    "name": "Hijawi",
    "date": "1288123622006"
  },
  {
    "price": 1880,
    "count": 0,
    "name": "Malek",
    "date": "1991-11-10T10:11"
  },
  {
    "price": 153,
    "count": 66655412,
    "name": "Saddam",
    "date": "1992-12-05T18:11"
  }
]
```

قال الله تعالى: فَمَنْ كَانَ يَرْجُو لِقَاءَ رَبِّهِ فَلْيَعْمَلْ عَمَلاً صَالِحًا وَلَا
يُشْرِكْ بِعِبَادَةِ رَبِّهِ أَحَدًا {الكهف: ١٠}

قد يعمد المسلم العمل لله تعالى لا للناس فيمدحه الناس عليه فلا
يعتبر ذلك رباء ولا يذم عليه؛ بل يكون بشري طيبة له، كما في
حديث أبي ذر - رضي الله عنه - قال : قيل لرسول الله - صلى الله
عليه وسلم : أرأيت الرجل يعمل العمل من الخير ويحمده الناس
عليه ، وفي رواية : يحبه الناس عليه . قال : " تلك عاجل بشري
المؤمن " . رواه مسلم .

باب إخلاص العمل لله تعالى

Filters

: يوجد بالـ filter اسمه Filters ^_ ^، ويقوم هذا الفلتر على اقتطاع جزء من المصفوفة وارجاعه على أنه مصفوفة جديدة بناءً على ما تمت تصفيته .. مثل البحث عن اسم anees ضمن قائمة من taher, anees, anas, saddam .. الخ، في حال كتابة .. سيظهر لنا مصفوفة جديدة تحتوي anas و anees .. الخ

الصيغة العامة:

```
{ { filter_expression | filter : expression : comparator } }
```

Filters

شرح الصيغة العامة:

يقصد بال expression التي string, object, function هو ال يمكن استخدامها لإختيار العناصر من المصفوفة ...

أما ال comparator فيقصد بها المقارنة التي يتم من خلالها فحص التطابق الكامل بين العنصر في المصفوفة والقيمة المراد البحث عنها .. وهي إما أن تكون true أو false|undefined أو

function(actual, expected) بحيث يقوم هذا ال function(actual, expected) بإرجاع true في حالة التطابق .. وفي حال استخدمنا ال true مباشرة فإنها اختصار ل function(actual, expected) { return angular.equals(actual, expected); }

Filters

مثال ١ :

```
<h2>Filter</h2>
<label>Search: <input ng-model="searchText"></label>
<div class="clearfix"></div>
<div ng-repeat="item in filter.filterItem | filter:searchText" class="col-xs-3">
  <div>Name: {{item.name}}</div>
  <div>Count: {{item.count}}</div>
  <div>Price: {{item.price}}</div>
  <hr />
</div>
```

هذا سيتم البحث أو التصفية بناء على أي من الحقول الاسم أو العدد أو السعر ..

Filter

Search: s

Name:Anees
Count:100
Price: 1254

Name:Saddam
Count:66655412
Price: 153

Filter

Search: sa

Name:Saddam
Count:66655412
Price: 153

Filters

الشيفرة البرمجية في الشريحة التالية.. (الأهمية الشرح)

Filter

Search All:	Search name:	Search count:	Search price:	Equality <input type="checkbox"/>
anees	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Name:Anees Count:100 Price: 1254				

Filter

Search All:	Search name:	Search count:	Search price:	Equality <input checked="" type="checkbox"/>
anees	<input type="text"/>	<input type="text"/>	<input type="text"/>	
لا يوجد نتائج ..(يجب أن يتطابق الإسم)				

Filter

Search All:	Search name:	Search count:	Search price:	Equality <input checked="" type="checkbox"/>
Anees	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Name:Anees Count:100 Price: 1254				

ستعمل مع البحث عن طريق
النص ..

Filters

لاحظ أنه يمكنني أن أستخدم كل object على حدٍ ل القيام بعملية البحث ^، كما أنني يمكنني تحديد جميع العناصر لأحد مربعات البحث من خلال استخدام ال \$... شاهد الشيفرة البرمجية للمثال السابق:

```
<h2>Filter</h2>


البحث في كل الخيارات


<label class="col-xs-2">Search All: <input ng-model="searchText.$"></label>
<label class="col-xs-2">Search name: <input ng-model="searchText.name"></label>


البحث عن طريق الاسم


<label class="col-xs-2">Search count: <input ng-model="searchText.count"></label>
<label class="col-xs-2">Search price: <input ng-model="searchText.price"></label>
<label class="col-xs-2">Equality <input type="checkbox" ng-model="strict"></label><br>
<div class="clearfix"></div>
<div ng-repeat="item in filter.filterItem | filter:searchText:strict" class="col-xs-3">
  <div>Name: {{item.name}}</div>
  <div>Count: {{item.count}}</div>
  <div>Price: {{item.price}}</div>
  <hr />
</div>
```

Filters

الآن .. نكون بإذن الله تعالى .. إنتهينا من موضوع ال Filter .. لكن يبقى علينا التنويه إلى جزئية قمنا بتأجيلها سابقا .. وهي استخدام ال filter من خلال الجافا سكريبت .. أو حتى انشاء custom filter من خلال الجافا سكريبت .. وهذا ما سنطرق اليه بإذن الله تعالى، وكما ان هناك صيغة عامة تعلمنا استخدامها داخل ال html .. هناك صيغة عامة للجافا سكريبت .. وبكل تأكيد لن أشرح عنها الى مثال واحد، لأن هذه الفكرة تشبه ما تعلمناه ^٨، والإختلاف أننا استخدمنا الجافا سكريبت للقيام بالعملية ^{*} _٨ ...

قبل الإنقال إلى الشريحة التالية .. استعرض جميع الأمثلة .. وتوقف على كل جزئية منها .. فكل جزئية تمثل فكرة .. والشيفرة البرجية كذلك:

شاهد المثال

Filters

استخدام الفلتر من الجافا سكريبت: (الخيارات الإختيارية تبقى اختيارية، والإلزامية .. تبقى كذلك). كما شرحنا سابقا `^_ ^` كمثال شاهد الصيغة

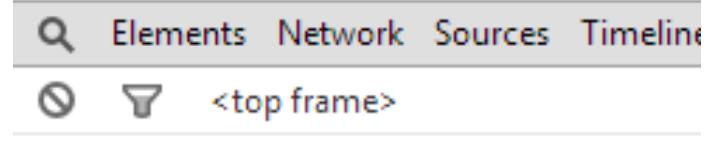
العامة للجافا سكريبت من هنا

In JavaScript

```
$filter('date')(date, format, timezone)
```

```
app.controller('aneesFiltersJS', function($filter) {
  this.filterItem = filterItems;

  console.log($filter('uppercase')("anees"));
  console.log($filter('date')('1992-12-05T16:11', "MM h 'o'clock'"));
});
```



Filters

والآن .. آخر جزئية بإذن الله تعالى وهي إنشاء custom filter

لكي نتمكن من إنشاء filter خاص بنا، هناك دالة في ال angular تمكنا من ذلك وهي `filter()`.
ويمكننا استخدامها في كل ... angular module نقوم بإنشائه داخل ال module.
الآن انظر إلى الشكل التالي:

```
app.filter(' ', function () {  
    return function () {  
        return;  
    };  
});
```

اسم الفلتر `^`

والآن ماذا يعني وجود ال `? return function`

إن ما يعنيه هو القيام باستدعاء لإرجاع قيمة ما، ويتم هذا الإرجاع في كل مرة يتم طلب ال `function` بها من خلال ال `angular`، وهذا يعني وجود حركتين لكل عملية اتصال بفلتر وهمما أخذ القيمة من جراء تفاعل المستخدم على أمر ما، ومن ثم تم الاتصال بالفلتر وتحديث التشكيل البيانات ..

Filters

والآن .. لنأخذ مثلاً عملياً، ولنقم بإنشاء فلتر بسيط يقوم بتربيع أي رقم..

```
app.filter('makeSquareNumber', function () {  
    return function (number) {  
        return Math.pow(number, 2);  
    };  
});
```

الـ

الـ makeSquareNumber اسم الفلتر، وهذا الفلتر سنجعله يربيع
الـ ... * _ ^ _ ...
الـ ... ^ _ ...

ملاحظة: لا آخذ بعين الاعتبار في الأمثلة حالات الخطأ الممكنة، مثل ادخال
نص بدل رقم وغيرها .. لأن هذه الأمور يجب أن تعرفها بالجافا سكريبت
وليس بدوره الـ angular ...

Filters

ملاحظات على المثال السابق:

(١) يتم انشاء global makeSquareNumber فلتر يمكن استخدامه، ولا يمكن استخدام هذا الفلتر الا من خلال ال app موديول .. *

(٢) سيتم في هذا المثال معالجة الأرقام التي تمثل object واحد، وليس مصفوفات وغيرها ..

والآن هنا طريقة استخدام المثال السابق: (ولاحظ كيف أنني أستخدم ال filter مع العنصر ك object داخل جملة الدوران..).

```
<div ng-repeat="item in filter.filterItem" class="col-xs-3">
  {{ item.price | makeSquareNumber }} </div>
app.controller('aneesFiltersJS', function($filter) {
  this.filterItem = filterItems;
});
```

Filters

مثال ٢: سنقوم الآن بإنشاء مثال لفلتر سيعمل مع ال repeat نفسها بدلاً من ال object ... هذا المثال سيقوم على فلترة عناصر المصفوفة التي تبدأ بالأحرف الخاصة بها بالحرف "a" ..(تملة المثال في

الشريحة التالية)

```
app.filter('startsWithA', function () {
  return function (items) {
    var filtered = [];
    قمنا بتعريف مصفوفة
    for (var i = 0; i < items.length; i++) {
      قمنا بتخزين الأويجك داخل متغير
      var item = items[i];
      if (/a/i.test(item.name.substring(0, 1))) {
        ادخال القيمة إلى المصفوفة
        filtered.push(item);
      }
    }
    هذا ارجاع المصفوفة بالقيم الجديدة بعد الفلترة ^_ ^
    return filtered;
  };
})();

var filterItems = [
  {price: 1254, count: 100, name: "Anees", date: '1288323623006'},
  {price: 963, count: -2112, name: "aHijawi", date: '1288123622006'},
  {price: 1880, count: 0, name: "Malek", date: '1991-11-10T10:11'},
  {price: 153, count: 66655412, name: "Saddam", date: '1992-12-05T18:11'},
];
```

Filters

حقيقة .. إن الفكرة من المثال السابق ليست جديدة، ففعلياً ما قمنا به هو عملية دوران لأخذ عناصر المصفوفة واجراء شرط، هذا الشرط هو الفلتر، والناتج الخاص به حفظ داخل مصفوفة أخرى، تمثل النتائج التي ستعرض .. وبهذا، يستقبل هذا الفلتر المصفوفة كاملة ..

```
<div ng-repeat="item in filter.filterItem | startsWithA" class="col-xs-3">
  <div class="alert alert-info">Name was start with "A" character: {{item.name}}</div>
</div>
```

Name was start with "A" character:
Anees

Name was start with "A" character:
aHijawi

رب اغفر لي ولوالدي رب ارحمهما كما ربياني صغيراً. ربنا آتنا في الدنيا حسنة وفي الآخرة حسنة وقنا عذاب النار ربنا لا تؤاخذنا إن نسينا أو أخطأنا ربنا ولا تحمل علينا إصراماً كما حملته على الذين من قبلنا ربنا ولا تحملنا ما لا طاقة لنا به واعف عننا واغفر لنا وارحمنا أنت مولانا فانصرنا على القوم الكافرين رب أوزعني أنأشكر نعمتك التي أنعمت عليّ وعلى والديّ وأن أعمل صالحاً ترضاه وأدخلني برحمتك في عبادك الصالحين وأصلاح لي في ذريتي إني تبت إليك وإنني من المسلمين. رب اجعلني مقيم الصلاة ومن ذريتي ربنا وتقبل دعاء ربنا اغفر لي ولوالدي وللمؤمنين يوم يقوم لحساب.

Filters

والآن .. مَاذا ؟ ^ ^

الم يت卜ادر لذهنك سؤال معين .. ؟

الم يدر في ذهنك .. كيف يمكنني استخدام متغيرات مع الفلتر، مثلاً، عندما تحدثنا عن ال JSON filter .. كنا نضع كلمة json ثم 4: أي اجعل المسافة بمقدار ٤ فراغات ..كيف يمكنني الآن مثلاً اضافة متغير مثل هذا لمثال الرقم، بحيث أحدد مقدار التضاعف .. قوة ٢ أو ٣ أو ٤ .. الخ، وكيف يمكنني أيضاً بنفس الطريقة وفي المثال الثاني، أن أحدد الحرف الذي سأقوم بالفلترة بنائاً عليه .. مثل ”b“ أو ”c“ .. الخ

هذا الأمر بسيط جداً .. ورائع، بحيث يعطيك قدرة كبيرة على التحكم بالفلتر الخاص بك .. ^ ^ .. لنشاهد مثلاً الآن:

Filters

المثال:

```
...<br />
<div ng-repeat="item in filter.filterItem" class="col-xs-3">
    {{ item.price | makeSquareNumberWithArg:powerNumber:message }}</div>
...
app.filter('makeSquareNumberWithArg', function () {
  return function (number, power, addMessage) {
    if(isNaN(power) || power < 0) {
      return 1;
    }
    if(addMessage) {
      addMessage = " This is your message with filter: " + addMessage.toUpperCase();
    } else {
      addMessage = "";
    }
    return Math.pow(number, power) + addMessage;
  };
});
```

لاحظ هنا .. لا يوجد اختلاف على الصيغة سوى أنشأ أضفنا قيماً جديدة، وهي القيم التي أرسلناها من خلال الـ **html**

1

Add your message

1254

963

2

Add your message

2

We used 2 as power!

1572516

927369

1572516 This is your message with
filter: WE USED 2 AS POWER

927369 This is your message with filter:
WE USED 2 AS POWER

Filters

لاحظ في المثال السابق.. أنني استقبلت القيم التي تم ارسالها من خلال الفلتر، وبعد ذلك جعلت أحد الخيارات الزامية وهو القوة المرفوعة للرقم، وبإرجاعي للرقم ١ فأنا أنهى الفلتر بهذه القيمة لعدم إنطاباق الشروط، ويمكنني أيضا عرض رسالة خطأ .. كما تريده يمكنك التحكم بذلك .. بينما الخيار الثاني .. قمت بجعل حقل الرسالة اختياري.. لاحظ كيف أيمكننا التعامل بكل سهولة مع الخصائص ^ *

والآن .. لنستخدم هذه الفكرة مع ال repeat أيضا * _ .. لنشاهد المثال الثاني:

Filters

مثال ٢:

```
<input class="form-control" type="text" ng-model="letter" />
<br />
<div ng-repeat="item in filter.filterItem | startsWithLetterArg:letter" class="col-xs-3">
  <div class="alert alert-danger">Name was start with "{{letter}}" character: {{item.name}}</div>
</div>
```

```
app.filter('startsWithLetterArg', function () {
  return function (items, letter) {
    var filtered = [];
    var letterMatch = new RegExp(letter, 'i');
    for (var i = 0; i < items.length; i++) {
      var item = items[i];
      if (letterMatch.test(item.name.substring(0, 1))) {
        filtered.push(item);
      }
    }
    return filtered;
  };
})
```

بنفس الفكرة والمثال الذي ذكرناه
سابقاً.. لكن باستبدال ال
"a" to letter

m

Name was start with "m" character:
Malek

Filters

بهذا نكون إنتهينا بإذن الله تعالى من حديثنا المتواضع عن الـ Filters، وكلّي رجاءً لك أنك التزمت بتطبيق جميع الأمثلة السابقة ..

الآن لمشاهدة جميع الأمثلة التي تم ذكرها والعمل بشكل عملي، يمكنك النقر هنا ... يرجى امعان النظر بالنتائج والشيفرة البرمجية وال .. console



ng-src Directive

الآن في فقرة سهلة وسريعة وكفاحل بين الدروس: P .. هل خطر
ببالك كيف يمكننا باستخدام ال angular path تحديد ال لصورة .؟

حسنا .. ما رأيك أن نجرب أو نرى هذه الطريقة:

```
<div ng-repeat="item in filter.filterImage" class="col-xs-3">
    {{ item.name }}
    
</div>
```

```
app.controller('aneesImageJS', function($filter) {
    this.filterImage = filterImages;
});
```

```
var filterImages = [
    {name: 'anees', image: '1.jpg'},
    {name: 'ali', image: '2.jpg'},
];
```

بعد تجربتك لها ..
انظر للخطأ وخمن
لماذا هذا الخطأ ؟
بعدها انتقل للشريحة
التالية ^ _

ng-src Directive

طبعاً لن تظهر الصورة، وسيظهر خطأ لديك في ال `*_*console` لماذا ؟

بكل بساطة وبالنهاية لا تنسى أننا نتعامل مع سكريبت، وهنا سيحاول المتصفح تحميل الصورة قبل تنفيذ ال `Exeprission` .. لهذا يجب علينا أن نجده حل أو طريقة أخرى لهذه المشكلة `..^..` ويكون الحل من خلال `ng-src` .. فهذه ستسمح لنا من بتحديد الصورة المطلوبة وجلبها، وهي فعلاً عملية `..^.. resource` ..
والآن لنذهب لمشاهدة المثال:

ng-src Directive

:مثال

```
<div ng-controller="aneesImageJS as filter">
  <div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
    <div class="col-xs-6">{{ item.name }}</div>
    
  </div>
</div>
```

anees



ali



شاهد المثال

الناتج

الله واحد
كتاب واحد
أمة واحدة

مسلم أحب ديني
الحمد لله الذي خلقي مسلماً مؤمناً بالله الواحد الأحد
ونسأله بعزته ورحمته ومغفرته وقدرته
أن يغفر لنا خطايña، وأن يوحد صفوفنا، وأن يرزقنا الهدایة
والثبات، وأن يجعلنا بفضله من الصالحين القانتين، وأن يجعلنا
برحتمه رحماء فيما بيننا ..
اللهم آمين يا رب العالمين.

إنشاء tabs من خلال ال angular

الآن.. لنتحدث عن المزيد من ال directive ... ولنشاهد تطبيقا عمليا جميلا، وهو إنشاء tabs للتنقل بين المحتوى .. (طبعا لا ننسى أننا نستخدم bootstrap كما ذكرنا أول الدورة اختصارا للوقت ..).

طبعا هذا تطبيق عملي وتفاعلية باستخدام ال angular .. وهو بسيط وسيشرح فكرة مهمة جدا .. قبل الإنقال الى الموضوع التالي ^
لذلك إنتبه وسر معي " حبة حبة ^ " .. شاهد المثال:

```
<section>
  <ul class="nav nav-pills">
    <li> <a href ng-click="tab = 1">2nees.com</a> </li>
    <li> <a href ng-click="tab = 2">2nees courses</a> </li>
    <li> <a href ng-click="tab = 3">2nees article</a> </li>
  </ul>
  {{tab}}
</section>
```

bootstrap class



إنشاء tabs من خلال ال angular

المثال السابق بسيط جداً.. عند النقر على أحد الروابط الموجودة ..سيقوم بتغيير قيمة ال tab وستشاهدها مطبوعة أسفل الحاوي الخاص بال شاهد النتيجة: Tab



إذا عدت للصورة السابقة ستلاحظ أن قيمة ال tab هي ٢ عند النقر عليها ^..وفعلياً ما حصل هو أن ال Expristion قام بعمل تحديث (updated) للقيمة الخاصة به مباشرة عند النقر باستخدام ال Ng-click ...والآن ..لنستبدل الرقم ولنعرض المحتوى الخاص بكل tab ...لكن كيف يتم ذلك ؟ تذكر ؟ حاول ..ثم انتقل للشريحة التالية *

إنشاء tabs من خلال ال angular

الآن .. سنقوم باستخدام ال ng-show التي ذكرناها سابقا .. !!
كيف ذلك؟! .. بكل بساطة من خلال أخذ القيمة من ال tab عند النقر
ومقارنتها داخل ال ng-show .. شاهد المثال:

```
<div class="cpanel" ng-show="tab == 1">  
    This is 1  
</div>  
  
<div class="cpanel" ng-show="tab == 2"> This is 3 ←  
    This is 2  
</div>  
  
<div class="cpanel" ng-show="tab == 3">  
    This is 3  
</div>
```

لاحظ أننا قمنا بعرض
المحتوى بنائياً على ال tab

2nees.com

2nees courses

2nees article



إنشاء tabs من خلال الـ angular

لكن هناك مشكلة في التطبيق السابق .. هل لاحظتها؟
المشكلة هي أنه لم يتم عرض أي محتوى ل tab الا عند قيمنا
بالنقر.. وهذه مشكلة تحل بكل بساطة من خلال تحديد قيمة ابتدائية لل
باستخدام `ng-init` ... لاحظ المثال:

```
<section ng-init="tab1 = 1">
  <ul class="nav nav-pills">
    <li> <a href ng-click="tab1 = 1">2nees
    <li> <a href ng-click="tab1 = 2">2nees
    <li> <a href ng-click="tab1 = 3">2nees
  </ul>

  <div class="tabpanel" ng-show="tab1 === 1">
    This is 1
  </div>

  <div class="tabpanel" ng-show="tab1 === 2">
    This is 2
  </div>

  <div class="tabpanel" ng-show="tab1 === 3">
    This is 3
  </div>
</section>
```

	2nees.com	2nees courses	2nees article	بدون init
--	-----------	---------------	---------------	-----------

	2nees.com	2nees courses	2nees article	وجود init = 1
--	-----------	---------------	---------------	---------------

إنشاء tabs من خلال الـ angular

* والآن .. ما رأيك أن نعطي المثال نوحا من التفاعلية مع المستخدم .. مثل اضافة الكلاس active لـ tab التي قام باختيارها .. اليك من المنشق القيام بذلك .. شاهد المثال:

يستخدم لإضافة كلاس من خلال ال directive هو (ng-class)

* angular



The screenshot shows a browser interface with three tabs at the top: '2nees.com', '2nees courses', and '2nees article'. The '2nees courses' tab is highlighted with a blue background and white text. Below the tabs, the text 'This is 2' is visible. A hand cursor is positioned over the '2nees courses' tab. At the bottom of the screen, the browser's developer tools are open, specifically the 'Elements' tab. The DOM tree is displayed, showing the HTML structure of the page. An arrow points from the text 'active' in the previous slide to the 'active' class in the highlighted line of the DOM tree.

```
Elements Network Sources Timeline Profiles Resources Audits Console
▶ <section ng-init="tab1 = 1">...</section>
  <hr>
▼ <section ng-init="tab2 = 1">
  ▼ <ul class="nav nav-pills">
    ::before
    ▼ <li ng-class="{active: tab2 === 1}" class="active">
      <a href ng-click="tab2 = 1">2nees.com</a>
    </li>
    ▶ <li ng-class="{active: tab2 === 2}" class="active">...</li>
    ▶ <li ng-class="{active: tab2 === 3}">...</li>
    ...<li>
```

إنشاء tabs من خلال الـ angular

الآن .. ما رأيك في التسلسل السابق ؟
أعتقد أن المثال إكتمل .. وهو سهل وواضح، طبعا ليس لأنني أنا من
كتبت المثال .. بل لأن الـ angular تبسط علينا المهام : P:
الآن ما أرغب بسؤال عنه ... هل يعجبك الشكل السابق؟
وهل تشعر بوجود خطأ منطقي في طريقة العمل هذه ؟
وهل تشعر بأن الشيفرة البرمجية نظيفة "Clean" ؟
طبعا يجب أن تكون إجابتك ب لا أو ربما ثم نعم أو ربما ثم لا أو ربما
... طبعا هي ليست " حزيرة " .. وإنما هي تخيل منطقي للشيفرة
فالشكل السابق قد يكون بسيطا .. لكننا بذات الوقت إعتمدنا إعتمادا كليا
على الـ html .. وهذا يسبب لنا خطأ منطقي .. أين الـ controller ؟!
وبهذا لن تعتبر الشيفرة البرمجية clean ! .. وبما أنه html سيتوجب عليه تكرار
هذه العملية في كل html .. لماذا لا أستخدم الـ controller وأنفذ العملية من
خلاله في كل صفحة html .. هذا هو ما يجب أن يسأل أو يجاب عليه & _

إنشاء tabs من خلال الـ angular

ولتحويل الصيغة الى صيغة مثالية أو صيغة أفضل اتبع الخطوات
التالية:

```
app.controller('tabsController', function() {
    this.tab3 = 1;           انشاء كونترولر، ووضع
}) ;                      القيمة الإفتراضية من خلاله
```

(1)

```
<section ng-controller="tabsController as tabCtrl">
    <ul class="nav nav-pills">
        <li ng-class="{active: tab3 === 1}"> <a href ng-click="tab3 =
        <li ng-class="{active: tab3 === 2}"> <a href ng-click="tab3 =
        <li ng-class="{active: tab3 === 3}"> <a href ng-click="tab3 =
    </ul>
```

```
<div class="tabpanel" ng-show="tab3 === 1">
    This is 1
</div>
```

سيتم التعامل الآن من خلال الكونترولر، وأول ما قمنا بفعله هو إنشاء الكونترولر، ومن ثم قمت بنقل القيمة الإفتراضية للجافا سكريبت

```
<div class="tabpanel" ng-show="tab3 === 2">
```

إنشاء tabs من خلال ال angular

٢) في هذه الخطوة، سيكون المطلوب منا بكل بساطة هو انشاء متغير من خلاله يمكننا الوصول وتعديل قيمة ال tab التي قمنا بالنقر عليها، لذلك تلاحظ أن المتغير tab3 هو من تغيرت قيمته بناء على setTab التي تم استقبالها من ال .. html

```
this.selectTab = function(setTab) {  
    this.tab3 = setTab;  
};
```

```
<a href ng-click="tabCtrl.selectTab(1)">2nees .com</a> </li>  
<a href ng-click="tabCtrl.selectTab(2)">2nees courses</a> <,  
<a href ng-click="tabCtrl.selectTab(3)">2nees article</a> <,
```

^_^ angular tabs من خلال إنشاء

```
this.isSelected = function(checkTab) {  
    return this.tab3 === checkTab;  
};  
  
<li ng-class="{active: tabCtrl.isSelected(1)}">  
<li ng-class="{active: tabCtrl.isSelected(2)}">  
<li ng-class="{active: tabCtrl.isSelected(3)}">  
  
<div class="cpanel" ng-show="tabCtrl.isSelected(1)">  
    This is 1  
</div>  
  
<div class="cpanel" ng-show="tabCtrl.isSelected(2)">  
    This is 2  
</div>  
  
<div class="cpanel" ng-show="tabCtrl.isSelected(3)">  
    This is 3  
</div>
```

٣) لاحظ هنا، فعليا عند التحقق من أي عملية فإننا نرجع قيمة ١ أو ٠ صحيح أو خطأ وبناءً عليها يكون الناتج المنطقي .. وهذا كذلك.. سيتم التتحقق هل ما تم النقر عليه يساوي الحالية .. اذا نعم.. أضف ال وأظهر المحتوى المطلوب

^_^ angular tabs من خلال إنشاء

والآن شاهد الشكل الكامل للمثال:

```
<section ng-controller="tabsController as tabCtrl">
  <ul class="nav nav-pills">
    <li ng-class="{active: tabCtrl.isSelected(1)}">
      <a href ng-click="tabCtrl.selectTab(1)">2nees .com</a>
    </li>
    <li ng-class="{active: tabCtrl.isSelected(2)}">
      <a href ng-click="tabCtrl.selectTab(2)">2nees courses</a>
    </li>
    <li ng-class="{active: tabCtrl.isSelected(3)}">
      <a href ng-click="tabCtrl.selectTab(3)">2nees article</a>
    </li>
  </ul>

  <div class="tabpanel" ng-show="tabCtrl.isSelected(1)">
    This is 1
  </div>

  <div class="tabpanel" ng-show="tabCtrl.isSelected(2)">
    This is 2
  </div>

  <div class="tabpanel" ng-show="tabCtrl.isSelected(3)">
    This is 3
  </div>
</section>
```

```
app.controller('tabsController', function() {
  this.tab3 = 2;

  this.selectTab = function(setTab) {
    this.tab3 = setTab;
  };

  this.isSelected = function(checkTab) {
    return this.tab3 == checkTab;
  };
});
```

شاهد المثال

أصبح من يتقن (الكلام) ربما يحترمه الجمّهور (الجاهل) أكثر بكثير
من أهل العلم الحقيقى

منتشر عن الأمانة المركبة من صفحة فن الحياة لـ الدكتور عبد الرحمن ناصر خطّفه الله

التعامل مع ال FORMS

والآن.. وللبدء بهذه الجزئية المهمة، علينا أن نستذكر Directive مهم جداً، وهو ال ng-model، وبكل تأكيد تحدثنا عنه سابقاً .. لكن الآن سنزيدكم من الشعر بيتاً ^_ ^، ونشاهد كيف يمكننا استخدامه مع ال ... * _ ^ Forms، وأهمية دوره مع ال Controller

- تذكير: تحدثنا سابقاً عن ال ng-model، ووظيفتها القيام بعمل data-bind مع عناصر التحكم ..

والآن لنشاهد مثلاً سريعاً، على طريقة التفاعل أو استخدام ال * _ ^.. Controller مع ال Ng-model

التعامل مع ال FORMS

مثال: الآن أنظر إلى المثال ..لا شيء جديد سوى أنني استخدمت ال ng-model الخاص بال controller داخل ال Alias name .. formCtrl.powerNumber فأصبح الإسم إطلاق التسمية ..الآن طبق المثال (مهم) قبل الإنقال للشريحة التالية: P

```
<section ng-controller="formsController as formCtrl">
  <form name="formReview" class="form">
    <input type="text" class="form-control" ng-model="formCtrl.powerNumber">
  </form>
  <div class="panel panel-info col-xs-6">
    {{formCtrl.powerNumber}}
  </div>
app.controller('formsController', function() {
  this.powerNumber = 55;
}) ;
```

التعامل مع ال FORMS

الآن .. ستلاحظ أمراً، وهو أن ال {{formCtrl.powerNumber}} هو أن ال صفحة، ثم تختفي وتعوض القيمة الفعلية مكانها! ستظهر أثناء تحميل الصفحة، ثم تختفي وتعوض القيمة الفعلية مكانها!

والسبب في هذا هو وضع ال js angular أو js عموماً في نهاية الصفحة أو قبل إغلاق ال body .. وبهذا فإنه لن يتم تحميل هذا الملف إلا بعد تنفيذ ما قبله من أوامر.. لذلك سيظهر ال Expression السابق ومن ثم يختفي .. ولحل هذه المسألة .. يمكنك أن تقوم بوضع ملفات ال js في ال header .. أو يمكنك استخدام ال ng-clock .. ^ _ * ..

وبكل بساطة فكرة ال ng-clock هي !! css هي وبالتحديد display بقيمة none .. لكن مع وجود تفاعل طبعاً أو عمليات من خلال ال angular .. لكن ما يهمك أنت .. هو طريقة استخدامه، شاهد مثلاً: ^ _ *

التعامل مع ال FORMS

```
[ng\:cloak], [ng-cloak], [data-ng-cloak], [x-ng-cloak], .ng-cloak, .x-ng-cloak {  
    display: none !important;  
}  
- )
```

style.css

```
<section ng-controller="formsController as formCtrl">  
    <form name="formReview" class="form">  
        <input type="text" class="form-control" ng-model="formCtrl.powerNumber">  
    </form>  
    <div class="panel panel-info col-xs-4">  
        without ng-cloak: {{formCtrl.powerNumber}}  
    </div>  
    <div class="panel panel-info ng-cloak col-xs-4">  
        with ng-cloak class: {{formCtrl.powerNumber}}  
    </div>  
    <div class="panel panel-info col-xs-4" ng-cloak>  
        with ng-cloak directive: {{formCtrl.powerNumber}}  
    </div>
```

ستشاهد الفرق أثناء تحميل الصفحة.. استمر بالضغط على Ctrl + F5 وستشاهد النتيجة أيضا ..

قبل تحميل الصفحة

without ng-cloak:
{{formCtrl.powerNumber}}

أثناء تحميل الصفحة ...
لاحظ الآن عدم ظهور ال
^
الـ divs
ng-clock _ ^

55

without ng-cloak: 55

بعد تحميل الصفحة

with ng-cloak class: 55

with ng-cloak directive: 55

التعامل مع ال FORMS

والآن ^_^ .. لنبدأ بال Forms بشكل فعلي

```
<form name="formReview" class="form" ng-submit="formCtrl.submitFunction()>
  <input type="text" class="form-control" ng-model="formCtrl.username">
  <input type="submit" class="btn btn-lg" value="Submit Form">
</form>
```

```
<div ng-cloak>{{ formCtrl.submitResult }} </div>
```

```
app.controller('formsController', function(){
  this.submitFunction = function() {
    this.submitResult = "You Are Submit Form with: " + this.username;
  };
});
```

شاهد هذا المثال البسيط .. أول فكرة وأهمها في ng-submit، والتي من خلالها يتم عمل submit لمحفوبيات هذا الفورم .. ولاحظ انني قمت بإجراة العملية التي أر غب بها، ولاحظ كيف بدأ يظهر دور ال ng-model

التعامل مع ال FORMS

والآن ..سيخطر ببالنا سؤال ..لماذا لم نستخدم ال ng-click هنا ؟!
اليس من الممكن استخدامها ؟.. لا يوجد فرق عما تعلمناه سابقا وقمنا به
من خلال ال ?ng-click

جميع ما قلته صحيح يا صديقي ^_^. لكن سأشرح لك الآن بعض النقاط المهمة التي توضح لماذا استخدمت ال ng-submit والتي تمثل فروقات بين ال ng-click وال ng-submit

١) نستخدم ال ng-submit مع ال form element تحديدا، بينما نستخدم ال ng-click مع أي عنصر آخر ..والسبب في ذلك أن ال يكون لجميع محتويات ال ..form submit (^_^.. Form is Submitted, Not Button

التعامل مع ال FORMS

٢) ينطلق ال trigger الخاص بال ng-submit من خلال ال click او من خلال ال enter فوق أي عنصر .. بينما ال ng-click اذا تم وضعها لل button فلن ينطلق الحدث الى عند النقر على ال .. btn

والآن .. لننتقل الى جزئية مهمة وهي:

Form Validation and States

وهنا سنتحدث عن الأخطاء الممكن حدوثها، وكيفية التحكم والتحقق .. ^_^ angular باستخدام ال

التعامل مع ال FORMS

والآن .. شاهد هذه الشيفرة البرمجية، وانتبه لكل خاصية ونقطة بها: ^_^

```
<form name="formReview" class="form" ng-submit="formCtrl.submitFunction2()" novalidate>
<input type="text" class="form-control" ng-model="formCtrl.courseName" required>
<input type="text" class="form-control" ng-model="formCtrl.coursePrice" required>
<input type="text" class="form-control" ng-model="formCtrl.courseLink" required>
<input type="submit" class="btn btn-lg" value="Submit Form">
</form>
<div ng-cloak>{{formReview.$valid}}</div>
```

لاحظ هنا.. أولاً قمنا بإطفاء التحقق الإعتيادي بالفورم من خلال استخدام ال novalidate الموجودة في ال HTML5 ..
بعد ذلك قمنا باستخدام ال required كمثال على خاصية ستحقق من تتحققها في الحقول الموجودة..

وبعد ذلك استخدمنا اسم الفورم ثم . ثم \$valid وهذا المتغير مهم، فهو المسؤول عن ارجاع النتيجة النهائية الخاصة بالتحقق للفورم ..-->

التعامل مع الـ FORMS

والآن .. شاهد تنفيذ الشيفرة البرمجية السابقة:

^ _ ^

✓

X

X

Submit Form

false نتيجة التحقق..
(وهذا يعني حسب مثالنا.. أن
الحقول لم يتم تعبئتها كاملاً..)

قم بتنفيذ المثال للضرورة
حتى تتابع معنا التغيرات
أولا بأول ^ _ ^ ...

AngularJS

✓

Free

✓

2nees.com

✓

Submit Form

true النتيجة الآن
صحيحة..

التعامل مع ال FORMS

الآن .. اذا لاحظت .. فإن النتيجة الخاصة بال \$valid تظهر مباشرة إما true وإما false حسب المعطيات التي تم التعامل معها .. والسؤال الآن - _ - ؟ .. كيف يمكنني استخدام هذه القيمة لمنع عملية الـ submitting الخاصة بالفورم ؟

يكون ذلك بوضعها كشرط ^ ... بالشكل التالي:

```
        ng-submit="formReview.$valid && formCtrl.submitFunction3() " no
```

والآن .. لنذهب سوياً لمشاهدة مثلاً عملياً على هذه الجزئية ^ *

التعامل مع ال FORMS

مثال:

```
this.submitFunction3 = function() {  
    this.submitResult3 = "Valid Form ^_^";  
    this.flag = true;  
};
```

app.js (in formsController)

```
<form name="formReview" class="form" ng-submit="formReview.$valid && formCtrl.submitFunction3()" novalidate>  
    <input type="text" class="form-control" ng-model="formCtrl.courseName" required>  
    <input type="text" class="form-control" ng-model="formCtrl.coursePrice" required>  
    <input type="text" class="form-control" ng-model="formCtrl.courseLink" required>  
    <input type="submit" class="btn btn-lg" value="Submit Form">  
</form>  
<div ng-cloak ng-show="formCtrl.flag" class="alert alert-success">{{ formCtrl.submitResult3 }} </div>
```

asd	✓
XCV	✓
	✗

Submit Form

AngularJS	✓
Free	✓
2nees.com ^_^	✓
Submit Form	Mouse Clicked
Valid Form ^_^	

إن دولة الخلافة الإسلامية، دولة تحتوي في ثناياها وجدورها خلفاء يقومون بامتصاص ما وهبهم الله، وبالصبر على ما هم فيه، وبالعمل لتحقيق هذه الخلافة. ويمك أفرادها إيمان وتقوى وعبادة ليست بمفهومها المختزل، وليس بشعائرها فقط .. بل بمفهومها الشامل .. هنا مجموع هؤلاء الأفراد سيشكل بكل تأكيد دولة الخلافة، ولن يهمنا من هو الخليفة أو القائد القادم .. فهو بالنسبة أحد هؤلاء الأفراد، ووظيفته من أكثر الوظائف دقة وصعوبة وأمانة، لكنه ليس بمعزل عن الجذور - مجموع الأفراد بفكر الخليفة - .. في هذه المرحلة بكل تأكيد سيمك المجتمع " فكرا جمعيا " يبني هذه الأمة، ويعيد مجدها وحضارتها ..

معاً نظمته من كتاب سيرة خليفة قلم الدكتور محمد هاشمي العربي

التعامل مع ال FORMS

والآن ..^{*} .. إلى جزئية مهمة وهي ال class التي تتغير في الحقول أثناء عملية التفاعل بين المستخدم والحقل ..^{*}، وتكون أهميتها في تمييز الحقول التي حصل بها الخطأ .. والحقول التي تم التحقق منها بنجاح .. وعمل هذا أمر سهل جدا ..^{*} .. كل ما عليك معرفته هو الكلاس الذي يتم تغييرها لحظة التفاعل والحالة المطلوبة .. ومن ثم تحديد اللون عن طريق ال CSS ..^{*}

والآن .. لنشاهد معاً مجموعة الكلاس التي يمكننا التعامل معها ..
الكلاس التي تحدث عنها تتعلق بعمليات ال input .. لذلك تسمى .. (input class

التعامل مع ال FORMS

ال Input Class التي سنتحدث عنها هي:

الحقل الإفتراضي ..

```
<input type="email" class="form-control" ng-model="formCtrl.email5" required>
```

هذه الكلاس يتم تحميلها قبل البدء بالكتابة من قبل المستخدم

```
<input type="email" class="" ng-pristine ng-invalid ..etc" ...>
```

هذه الكلاس يتم تحميلها عند كتابة المستخدم المعلومات ولم يتحقق الشرط (كانت المعلومات المدخلة خاطئة ..)

```
<input type="email" class="" ng-dirty ng-invalid ..etc" ...>
```

يتم تنفيذ هذه الكلاس في حالة الكتابة وتحقق الشرط ..(المعلومات صحيحة)

```
<input type="email" class="" ng-dirty ng-valid ..etc" ...>
```

قبل أي ادخال ..(برتقالى)

ادخل صحيح (أخضر)

حقل بريد الكتروني (تحقق خاطئ)

Submit Form

Submit Form

anees

anees@2nees.com|

Submit Form

صحيح
اطار أخضر

```
.form5-ex .ng-pristine.ng-invalid {  
    border-color: orange;  
}  
  
.form5-ex .ng-valid.ng-dirty{  
    border-color: green;  
}  
  
.form5-ex .ng-invalid.ng-dirty{  
    border-color: red;  
}
```

التعامل مع ال FORMS

مثال:

```
<form name="formReview5" class="form form5-ex" ng-submit="formReview5.$valid && formCtrl.submitFunction5()" novalidate>
    <input type="text" class="form-control" ng-model="formCtrl.name5" required>
    <input type="email" class="form-control" ng-model="formCtrl.email5" required>
    <input type="submit" class="btn btn-lg" value="Submit Form">
</form>
<div ng-cloak ng-show="formCtrl.flag5" class="alert alert-success">{{ formCtrl.submitResult5 }} </div>
.
.
this.flag5 = false;
this.submitFunction5 = function() {
    this.submitResult5 = "Valid Form 5 With Email rule & required *_*";
    this.flag5 = true;
};

    ].form5-ex .ng-pristine.ng-invalid {
        border-color: orange;
    }

    ].form5-ex .ng-valid.ng-dirty{
        border-color: green;
    }

    ].form5-ex .ng-invalid.ng-dirty{
        border-color: red;
    }
```

هو كلاس وضعته لكي يتم تنفيذ المثال هنا
فقط ..

التعامل مع ال FORMS

ملاحظة:

بهذه الجزئية وبما تم شرحه الى الان، يمكنك المتابعة والبحث عن المزيد من الخصائص، واستخدام ال angular dir المختلفة والبحث عنها مثل ال pattern .. مثل ال ng-minlength .. مثل ال checkbox .. الأمور سهلة جدا ولا يوجد اختلاف سوى أنك ستقرأ قليلا ..

مثلا ال checkbox من خلال ال angular سترجع true أو false بناءا على الشكل التالي ctrl.user.agree .. ويمكننا أيضا إسناد قيمة أخرى مثل Angular أو C++ أو P.. وذلك من خلال الشكل التالي:

ng-model="ctrl.user.agree"
ng-true-value="Angular"
ng-false-value="C++"

..... والآن لنشاهد ما تم ذكره من أمثلة

شاهد المثال

Custom Directive وال ng-include

والآن .. الى موضوع جديد ^_ ^ وهو ال !! ng-include

هل جاء في خاطرك يوماً أن تتطلع الى استخدام نفس الشيفرة الخاصة بال HTML في أكثر من صفحة دون تكرارها ؟
اذا كانت الإجابة نعم، فيمكنك صنع ذلك الان دون الحاجة الى استخدام لغة PHP, ASP مثل Server Side .. الخ
كيف ذلك ؟ ^_ ^

بكل بساطة من خلال ng-include والموجودة داخل ال * ^ AngularJS

Custom Directive وال ng-include

والآن..لنعد قليلا لمشاهدة مثال ذكرناه سابقا:

```
<div ng-controller="aneesImageJS as filter">
<div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
    <div class="col-xs-6">{{ item.name }}</div>
    

```

ما أريده هو تضمين هذه الشيفرة البرمجية في أي صفحة أخرى دون إعادة تكراره ..لذلك قم بنسخ هذه

الشيفرة البرمجية الآن ..ودعنا نتابع ^

والآن ..شاهد كيف سأقوم بتضمين هذه الشيفرة البرمجية واستدعائها عند الحاجة ^، ولاحظ أننا سنأخذها مع ال Expressions التي تم استخدامها ^ . قمة الروعة ^ .(تابع معي أولا بأول هنا)

```
<span ng-include="'14-include.html'"></span>
```

Custom Directive وال ng-include

في طريقة التمixin السابقة نلاحظ أهمية وجود "، وبداخلها يتم وضع الصفحة التي يجب تضمينها ^..
والي الان لنشاهد شكل الصفحة التي سنقوم بتضمينها .. *



```
1 <div class="col-xs-6">{{ item.name }}</div>
2 
```

14-include.html

هذه هي الشيفرة التي قمنا بنسخها ونهدف الى تضمينها ..

وهذا هو الشكل الخاص بالصفحة التي سنقوم باستخدام التمixin بها
... (14.html)

```
----- .
<div ng-controller="aneesImageJS as filter">!-- This code copy from ex-11.html to ex
  <div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
    <span ng-include="'14-include.html'"></span>
  </div>
</div>
```

Custom Directive وال ng-include

والآن لنعيد ترتيب أوراقنا ونشاهد المثال كاملاً، ومن ثم سنقوم بتنفيذه

* ^

The screenshot shows a code editor with the file **14.html** open. The code includes an AngularJS application setup with controllers and a custom directive implementation. A specific section of the code is highlighted in yellow, showing the use of `ng-include` to include the content of **14-include.html**. Below the editor, a preview window shows the resulting HTML structure with the included content. The status bar at the bottom displays several tabs including **14.html**, **style.css**, **app.js**, **11.html**, and **14-include.html**.

```
<!DOCTYPE html>
<html ng-app="store">
  <head>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>

  <body class="ex-14 container">
    <div ng-controller="aneesImageJS as filter">!-- This code copy from ex-11.html to execute
      <div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
        <span ng-include="'14-include.html'"></span>
      </div>
    </div>

    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
    <script type="text/javascript" src="js/angular.min.js"></script>
    <script type="text/javascript" src="js/app.js"></script>
  </body>
</html>
```

14.html

14-include.html

14-include.html

هذه هي الشيفرة التي قمنا بنسخها ونهدف الى تصميمها ..

الآن .. قم بتنفيذ المثال – ومن ثم انتقل للشريحة التالية .. مهمـا كانت النتيجة:

انيس حكمت أبو حميد (دورـة مجـانية) 2nees.com

Custom Directive وال ng-include

عند تطبيقنا للمثال السابق، سيقوم بإعطائنا خطأ جميل ^.. وهو:

الشكل الناتج من التنفيذ

الخطأ الحاصل

```
Elements Network Sources Timeline Profiles Resources Audits Console
<top frame>
XMLHttpRequest cannot load file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/14-include.html. Cross origin requests are only supported for HTTP.
Error: Failed to execute 'send' on 'XMLHttpRequest': Failed to load 'file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/14-include.html'.
    at Error (native)
    at file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:87:37
    at n (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:82:442)
    at f (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:80:174)
    at file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:112:113
    at n.$eval (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:126:15)
    at n.$digest (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:123:106)
    at n.$apply (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:126:293)
    at file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:17:479
    at Object.e [as invoke] (file:///C:/Users/interactive3.ALROWAD/Desktop/Anees%20File/angular/js/angular.min.js:36:315)
angular.min.js:87
angular.min.js:102
```

سأقوم الآن بالبحث عن المشكلة بجوجل ..(طبعاً الحل والمشكلة بسيطة ومن البديهيات .. لكن خطر بيالي التطرق لمحاولة حل المشاكل التي نتعرض لها من خلال البحث .. لترسيخ هذا المفهوم ..)



Cross origin requests are only supported for HTTP



Google

أدوات البحث

المزيد

الأخبار

صور

فيديو

ويب

دائماً ابحث في جوجل .. وستجد الحل
لكن هذه المرة سأقدم لكل الحل على طبق من ذهب ^_^

حوالي ٤٢٦٠٠٠٠ من النتائج (عدد النتائج: ٥٧٠٠٠)

[Cross origin requests are only supported for HTTP but it's ...](#)
[ترجم هذه الصفحة](#)

You need to actually run a webserver, and make the get request to a URI on that server, rather than making the get request to a file; e.g. change the ...

["Cross origin requests are only supported for HTTP." error ...](#)
[ترجم هذه الصفحة](#)

I'm trying to load a 3D model into Three.js with JSONLoader , - 2012/05/25
and that ... My crystal ball says that you are loading the model using either
file:// or ...

XMLHttpRequest cannot load file. Cross origin requests are

الصفحة

in ... Cross
your bro

5 Answers

ng-include

لاحظ حتى أنني لم أحدد المعلومات الكافية .. فقط بحث سريع .. نسخ ولصق وكانت النتيجة أنني وجدت الحل من الرابط الأول .. وحتى من الرابط الرابع..P:

active oldest votes

Angular
هذه الصفحة

43

You need to actually run a webserver, and make the get request to a URI on that server, rather than making the get request to a file; e.g. change the line:

`$.get("C:/xampp/htdocs/webname/resources/templates/signup.php",`

7 Answers

active oldest votes

أنيس حكمت أبو حميد (دورة مجانية) 2nees.com

هل عرفت أين

المشكلة؟

نعم يا صديقي، إنها باستخدام المتصفح التقليدي لاستعراض هذه الصفحة.. والحل يكون هو تشغيل الصفحة من خلال السيرفر مثل أباتشي



42

This error is happening because you are just opening html documents directly from the browser. To fix this you will need to serve your code from a webserver and access it on localhost. If you have apache setup, use it to serve your files. Some IDE's have built in web servers, like jetbrains IDE's, Eclipse...



If you have node setup then you can use `http-server`. Just run `npm install http-server -g` and you will be able to use it in terminal like `http-server C:\location\to\app`.

Custom Directive وال ng-include

والآن .. لنقوم بتطبيق المثال على السيرفر (ما لدی هو سيرفر أباتشي في بيئة تطوير wamp) وبكل تأكيد لن يهمني نوع بيئة التطوير التي تعمل عليها أو نوع السيرفر .. المهم هو وجوده:

The screenshot shows a web browser window with two user profiles. The left profile, 'anees', contains a shopping cart icon and a red message in Arabic: 'ناتج تطبيق المثال'. The right profile, 'ali', contains a book icon. The browser's address bar shows the URL '192.168.0.147/angularCourse/14.html'. Below the browser is a developer tools console with the following text:
الرابط الخاص بك يجب أن يكون بهذا الشكل
localhost/folderName/pageName.html

يقول الرافعي رحمه الله: "عَلِمَ اللَّهُ! مَا فَتَنَ الْمُغْرُورِينَ مِنْ شَبَابِنَا إِلَّا
مَا أَخْذَهُمْ مِنْ هَذِهِ الْحَضَارَةِ، فَإِنَّ لَهَا فِي زِينَتِهَا وَرُونَقِهَا أَخْذَةً
كَالسُّحْرِ، فَلَا يَمِيزُونَ بَيْنَ خَيْرِهَا وَشَرِهَا، وَلَا يَفْرَقُونَ بَيْنَ مَبَادِئِهَا
وَعَوَاقِبِهَا، ثُمَّ لَا يَقْتَنُونَ مِنْهَا إِلَّا مَا يَدْعُوهُمْ إِلَى مَا يَمِيتُ وَيَصْدِهِمْ
عَمَّا يَحْيِي وَمَا يَحُولُ بَيْنَهُمْ وَبَيْنَ قُلُوبِهِمْ، فَلَيْسَ إِلَّا الْمُتَابِعَةُ
وَالتَّقْلِيدُ".

Custom Directive وال ng-include

..الآن نكون انتهينا من هذه الجزئية (ال ng-include) والآن
إلى الجزئية الأخرى وهي ال Custom Directive ..ولنبدأ بها
على بركة الله

ال Custom Directive الذي سنقوم ببنائه، يشابه فكرة العمل
الخاصة بال ng-include ..ومنها سنتعلم إنشائه، ومنها سنشاهد لماذا
يوجد خيار مثل هذا، وما هي أهميته ..

والسؤال الأول والأهم الآن، لماذا يجب أن أقوم بإنشاء Custom Directive
ليقوم بنفس هذه الوظيفة ؟

Custom Directive وال ng-include

وهذا السؤال منطقي جدا .. وجوابه يبدأ من نقطة مهمة وهي أن ال يمكننا من كتابة شيفرة HTML من تصميمي وصيغة Directive السلوك الخاص بها من خلال التطبيق ^ ^ .. مثلا شاهد الشيفرة التالية:

```
<aside class="col-sm-3">
  <book-cover></book-cover>
  <h4><book-rating></book-rating></h4>
</aside>

<div class="col-sm-9">
  <h3><book-title></book-title></h3>

  <book-authors></book-authors>

  <book-review-text></book-review-text>

  <book-genres></book-genres>
</div>
```

اذن !! ماذا بعد !!

انتقل للشريحة التالية:

Custom Directive وـ ng-include

- وهذا يعني
- (١) أننا يمكننا إنشاء واستخدام Custom Tag أو Attribute والتي يمكن استخدامها أو تعديلها والعمل عليها ... (وراثتها من أجل التطبيق)
 - (٢) يمكننا تضمين ال Controller Logic واستعماله من خلال هذا البناء اذا أردنا ذلك.
 - (٣) يمكن أن يتم استخدامه لبناء UI المعقدة
 - (٤) يمكنني من استخدام بعض المكونات أو الأوامر المجودة سلفاً أو التي تم استخدامها ..

Custom Directive وال ng-include

والآن كيف يمكننا بناء Custom Directive ؟ قبل شرح الآلية لذلك .. يتوجب علينا شرح المكونات التي يمكننا استخدامها هنا، ومع أنّس سنكتفي بصورة مبسطة و مختصرة عن الموضوع لوضوح الفكرة .. إلا أنه من الأهمية بمكان شرح مثل هذه الخصائص ^ *

Property	Description	2nees.com
restrict	وتسخدم هذه المزية لتحديد المكان الذي يمكننا من خلاله استخدام ال Directive .. هل هو من خلال أن يكون Element or attribute or css class or comment ...etc	
scope	ويستخدم لإنشاء نطاق جديد للأبناء ل Controller معين، أي إمكانية الوصول المباشر (Shared) للعناصر الموجودة داخله أم فقط من خلال استخدام ال scope ... وسيُسمى هنا isolate scope	
template	وهذا يمكننا استخدامها لوضع المحتوى الذي سيمثل الناتج الخاص بال Directive ويمكن من تضمين HTML Data Binding Expressions, other directives ...etc	
templateUrl	وهذا يمكننا وضع الرابط الخاص بالملف الذي سنقوم بتضمينه والذي سيستخدمه ال directive ويمكن اختياريا استخدام ال ... Dom Element id	
controller	ويمكننا استخدام هذه المزية لتحديد ال Controller المناسب ليعمل مع ال Template Directive	
link	وهذا تقوم باستخدامه لتنفيذ أكتر من مهمة function Dom Manipulation tasks	

Custom Directive وال ng-include

مثال على إنشاء Custom Directive

```
<div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
    <course-title></course-title>
</div>
```

بنفس فكرة المثال السابق.. لكن استبدلنا هنا ال **ng-include** ب **<course-title></course-title>** HTML
Custom TAG ^_^

ال tag الذي قمنا باستخدامه.. ال - تتحول الى صيغة **camelCase**

```
// Example In 14.html
app.directive('courseTitle', function() {
    return {
        restrict: 'E',
        templateUrl: '14-include.html',
    };
});
```

اختصار ل **Element**

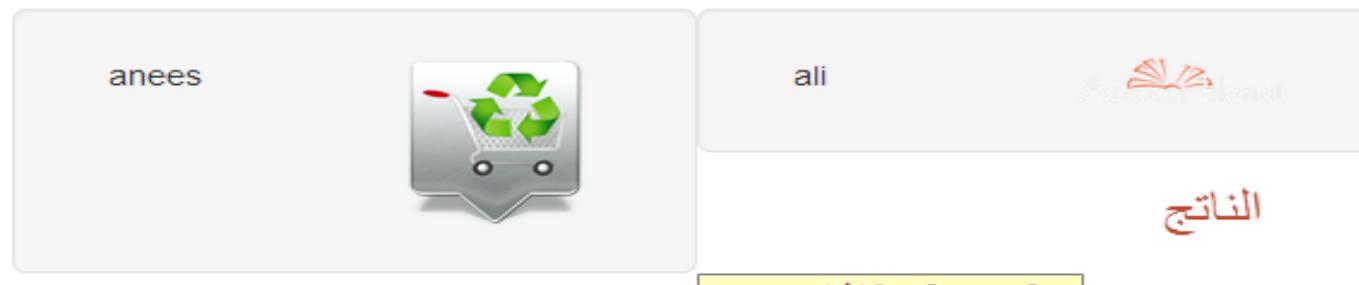
رابط الملف الذي سنقوم بتضمينه

ملف الجافا سكربت مهم جدا هنا .

14-include.html
هي ذات الصفحة في المثال السابق

Custom Directive وال ng-include

والآن لنقوم بتطبيق المثال ولنشاهد النتائج (Run on Server)



Elements Network Sources Timeline Profiles Resources Audits Console

```

</span>
</div>
<!-- end ngRepeat: item in filter.filterImage --&gt;
► &lt;div class="clearfix"&gt;...&lt;/div&gt;
&lt;hr&gt;
<!-- ngRepeat: item in filter.filterImage --&gt;
▼ &lt;div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg ng-scope"&gt;
  ▼ &lt;course-title&gt;
    &lt;div class="col-xs-6 ng-binding"&gt;anees&lt;/div&gt;
    &lt;img ng-src="1.png" alt="Image" class="col-xs-6 image-responsive" src="1.png"&gt;
  &lt;/course-title&gt;
&lt;/div&gt;
<!-- end ngRepeat: item in filter.filterImage --&gt;
▼ &lt;div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg ng-scope"&gt;
  ▼ &lt;course-title&gt;
    &lt;div class="col-xs-6 ng-binding"&gt;ali&lt;/div&gt;
    &lt;img ng-src="2.png" alt="Image" class="col-xs-6 image-responsive" src="2.png"&gt;
  &lt;/course-title&gt;
&lt;/div&gt;
<!-- end ngRepeat: item in filter.filterImage --&gt;</pre>
```

Custom Directive وال ng-include

ملاحظة: في المثال السابق قمنا باستخدام

<course-title></course-title>

ولم نقم باستخدام <course-title/> ..
XHTML ليست مدعومة في كل مكان .. فمثلا هي تقرأ من خلال ال
شكّلها <course-title></course-title> .. لكن في ال
HTML5 تقرأ على <course-title> .. لذلك وجب التنبيه ..

والآن.. لنكمل سلسلة الأمثلة بإذن الله تعالى ^_ ^ ..

طبعا تحدثنا عن ال restrict ، وفي مثالنا السابق استخدمنا الحرف E
والذي رمّزنا به إلى Element .. والآن ما رأيك بمشاهدة ال A ^_ ^

Custom Directive وال ng-include

كما شرحتنا في الشرائح السابقة، فإن ال Directive يمكن أن يكون Attribute .. و يمكن أن يكون Element (ذكر من هنا) الخ

* * Attribute على أنه Directive على أنه Directive

```
<div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg">
  <div course-title-attr></div>      HTML
</div>
```

anees



ali

ناتج التطبيق

```
app.directive('courseTitleAttr', function() {
  return {
    restrict: 'A',
    templateUrl: '14-include.html',
  };
});
```

App.js

Elements Network Sources Timeline Profiles Resources Audits Console

```
<div ng-repeat="item in filter.filterImage" class="col-xs-3 well well-lg ng-scope">
  <div course-title-attr>
    <div class="col-xs-6 ng-binding">anees</div>
    
  </div>
</div>
```

Custom Directive وال ng-include

والآن .. هنا سننهي حديثنا بإذن الله تعالى عن موضوع ال Directive ، وسننتقل الى موضوع جديد وشيق بإذن الله تعالى لكن وقبل ذلك، عليك القيام بتطبيق الأمثلة التي تم ذكرها ..

وبعد ذلك يمكنك مشاهدة جميع الأمثلة التي تم ذكرها في هذا الموضوع من هنا ^_ ^، تتبع الشيفرة البرمجية بانتباه، الموضوع سهل جدا، كما أنيوه الى أهمية تطبيق هذه الأمثلة على ال server، وقد شرحتنا ذلك في بداية هذا الموضوع ..(الملفات المطلوب نسخها للسيفر ليعمل المثال هي: js folder, css folder, 14.html, 14-include.html, 1.png, 2.png) ليصبح الشكل كما في الشريحة التالية:

Custom Directive وـ ng-include

Local Disk (C:) ▶ wamp ▶ www ▶ angularCourse ▶

Category	Share with	Burn	New folder
Name	Date modified	Type	Size
css	17-Jun-15 10:23 AM	File folder	
js	17-Jun-15 10:23 AM	File folder	
1.png	16-Feb-15 12:33 PM	PNG image	76 KB
2.png	22-Oct-14 11:26 AM	PNG image	12 KB
14.html	23-Jun-15 11:53 AM	Chrome HTML Do...	2 KB
14-include.html	17-Jun-15 11:17 AM	Chrome HTML Do...	1 KB

هذه الملفات التي تم نسخها لتطبيق المثال 14 .. (انتبه، لا
تحذف هذه الملفات من مجلد الأمثلة -_-)

شاهد المثال ك HTML

أَعُوذُ بِاللَّهِ مِنَ الشَّيْطَانِ الرَّجِيمِ
الْحَجَّ أَشْهُرٌ مَعْلُومَاتٌ حَفَّمَنْ فَرَضَ فِيهِنَّ الْحَجَّ فَلَا رَفَثَ وَلَا فُسُوقَ وَلَا جَدَالَ
فِي الْحَجَّ قَوْمًا تَفْعَلُوا مِنْ خَيْرٍ يَعْلَمُهُ اللَّهُ وَتَزَوَّدُوا فَإِنَّ خَيْرَ الزَّادِ التَّقْوَى حَجَّ
وَاتَّقُونَ يَا أُولَئِي الْأَلْبَابِ (١٩٧) لَنْسَ عَلَيْكُمْ جُنَاحٌ أَنْ تَبْتَغُوا فَضْلًا مِنْ رَبِّكُمْ حَجَّ
فَإِذَا أَفَضْتُمْ مِنْ عَرَفَاتٍ فَادْكُرُوا اللَّهَ عِنْدَ الْمَشْعَرِ الْحَرَامِ طَوْلًا وَادْكُرُوهُ كَمَا هَدَأْكُمْ
وَإِنْ كُنْتُمْ مِنْ قَبْلِهِ لَمَنِ الضَّالِّينَ (١٩٨) ثُمَّ أَفْيَضُوا مِنْ حَيْثُ أَفَاضَ النَّاسُ
وَاسْتَغْفِرُوا اللَّهَ إِنَّ اللَّهَ غَفُورٌ رَحِيمٌ (١٩٩) فَإِذَا قَضَيْتُمْ مَنَاسِكُكُمْ فَادْكُرُوا اللَّهَ
كَذِكْرُكُمْ آبَاءَكُمْ أَوْ أَشَدَّ ذِكْرًا قَوْمًا مِنَ النَّاسِ مَنْ يَقُولُ رَبَّنَا آتَنَا فِي الدُّنْيَا وَمَا لَهُ
فِي الْآخِرَةِ مِنْ خَلَاقٍ (٢٠٠) وَمِنْهُمْ مَنْ يَقُولُ رَبَّنَا آتَنَا فِي الدُّنْيَا حَسَنَةً وَفِي
الْآخِرَةِ حَسَنَةً وَقِنَا عَذَابَ النَّارِ (٢٠١) أَوْلَئِكَ لَهُمْ نَصِيبٌ مِمَّا كَسَبُوا حَجَّ وَاللَّهُ
سَرِيعُ الْحِسَابِ (٢٠٢)

صدق الله العظيم

سورة: البقرة

\$http

ال Angular \$http هي Service موجودة بال Core الخاص بال جلب البيانات من خلال السيرفر.

والآن كيف يتم جلب البيانات؟
طبعا هناك أكثر من نوع لل Methods التي يمكننا استخدامها، ولكن
الطريقة التقليدية والسهلة هي باستخدام ال GET (طبعا لن أشرح ال GET أو ال POST وغيرها من ال Methods فيفترض باك معرفتها ..)

والآن سأقوم بشرح الطريقة الرئيسية وطريقة إعداد ال \$http بإذن الله تعالى *

\$http

أول معلومة أريده أن تعرفها هي أن ال \$http يمكنها أن تعمل الإتصال بين المتصفح والسيرفر من خلال وسعتين الأولى من خلال ال ..JOSNP والثانية من خلال XMLHttpRequest Object

ثاني معلومة وهي مهمة (ال Response Status)، فإذا كنت لا تعرفها أو تعرف أرقامها يمكنك الإطلاع عليها [من هنا](#) ، وتكون أهمية هذه النقطة بمعرفة النتيجة من الطلب الذي قمت بإرساله من المتصفح إلى السيرفر، وما يهمني هنا معرفة أن الرقم ٢٠٠ إلى ٢٩٩ تدل على حالات نجاح مختلفة .. ^ _ ..

\$http

ثالث معلومة وهي ال method الى يمكن استخدامها هنا وهي:

- \$http.get
- \$http.head
- \$http.post
- \$http.put
- \$http.delete
- \$http.jsonp
- \$http.patch

رابع معلومة، إن ال Http headers يتم انشائها بشكل تلقائي من قبل ال Angular، ويمكنك التعديل على ال header من خلال \$httpProvider.defaults.headers.*

\$http

المعلومة الخامسة: P

الصيغة العامة لل Request Method التالية

*(url, [config]); هي get, delete, head , josnp
أما ال * (url, data, [config]); فهي post, patch, put
شرح ما تم ذكره:

*: ترمز لأي method مثل (.. الخ
ال url: هو خيار الزامي ويشير الى رابط الصفحة المطلوبة على
السيرفر.

ال data: وهي تخص ال post, patch, put فتعني البيانات التي
سيتم ارسالها ..

ال config: خيار اختياري يشير الى أي إعداد آخر نحب إضافته..

\$http

والآن ..كيف يمكننا إستخدام ال \$http ؟ وكيف يمكنني تحويل ما تم ذكره في الشرائح القليلة الماضية الى شيفرة برمجية ^_^. لشاهد معا

```
// Simple GET request example :  
$http.get('/someUrl').  
success(function(data, status, headers, config) {  
  // this callback will be called asynchronously  
  // when the response is available  
}).  
error(function(data, status, headers, config) {  
  // called asynchronously if an error occurs  
  // or server returns response with an error status.  
});
```

مثال على

GET

(طبعاً هذه الصورة
القياسية للإستخدام،
والآن، لنتابع بإذن
الله تعالى ونشاهد
أمثلة على هذا

```
// Simple POST request example (passing data) :  
$http.post('/someUrl', {msg:'hello word!'}).  
success(function(data, status, headers, config) {  
  // this callback will be called asynchronously  
  // when the response is available  
}).  
error(function(data, status, headers, config) {  
  // called asynchronously if an error occurs  
  // or server returns response with an error status.  
});
```

مثال على

POST

(^_^ الموضع

\$http

(انتبه: تنفيذ الأمثلة التي سيتم ذكرها يجب أن يكون على السيرفر، كما أن الشيفرة البرمجية التي سيتم وضعها ستستخدم في جميع الأمثلة الأخرى، والخاصة بهذا الموضوع.)

```
{  
    "records": [  
        {  
            "Name" : "2nees.com",  
            "Price" : "Free",  
            "id" : "1"  
        },  
        {  
            "Name" : "Angular",  
            "Price" : "10$",  
            "id" : "2"  
        },  
        {  
            "Name" : "CSS",  
            "Price" : "20$",  
            "id" : "3"  
        },  
        {  
            "Name" : "HTML5",  
            "Price" : "30$",  
            "id" : "4"  
        }  
    ]
```

أولاً قم ببناء صفحة php وضعها على السيرفر، وهنا سأسميها .. 15.Php

\$http

ثانياً: قم بوضع هذه الشيفرة البرمجية داخل صفحة html على السيرفر
أيضا ..(طبعا هذه الشيفرة ستكون داخل ال
angular template كما في جميع الأمثلة التي تم ذكرها سابقا) ..

```
<div ng-controller="httpController">
    <li ng-repeat="x in 2neesCourses">
        {{ x.id + ' - ' + x.Name + ' * ' + x.Price}}
    </li>
</div>
```

\$http

ثالثاً: نقوم بإضافة الـ Controller الآن ^_^

```
app.controller('httpController', function($scope, $http){  
    $http.get("http://192.168.0.147/angularCourse/15.php").success(  
        function(response) {  
            $scope.aneesCourses = response.records;  
        }  
    );  
});
```

صيغة بسيطة لتوسيع الفكرة ..

طبعاً هذه الصيغة بأبسط صورها وذلك لتوضيح الفكرة، وهذا هو المثال الأول ..

طبعاً الـ 192.168.0.147 قم باستبدالها برقم الـ ip الخاص بك أو حسب إعدادات السيرفر الخاص بك .. localhost

\$http

مثال ١ بشكله الكامل:

```
<div ng-controller="httpController">
  <li ng-repeat="x in 2neesCourses">
    {{ x.id + ' ^_^ ' + x.Name + ' *_* ' + x.Price}}
  </li>
</div>
```

15.html

```
"records": [
  {
    "Name" : "2nees.com",
    "Price" : "Free",
    "id" : "1"
  },
  {
    "Name" : "Angular",
    "Price" : "10$",
    "id" : "2"
  },
  {
    "Name" : "CSS",
    "Price" : "20$",
    "id" : "3"
  },
  {
    "Name" : "HTML5",
    "Price" : "30$",
    "id" : "4"
  }
]
```

```
app.controller('httpController', function($scope, $http) {
  $http.get("http://192.168.0.147/angularCourse/15.php").success(
    function(response) {
      $scope.2neesCourses = response.records;
    }
));
```

صيغة بسيطة لتوضيح الفكرة..

App.js

1 ^_^ 2nees.com *_* Free

2 ^_^ Angular *_* 10\$

3 ^_^ CSS *_* 20\$

4 ^_^ HTML5 *_* 30\$

النتائج

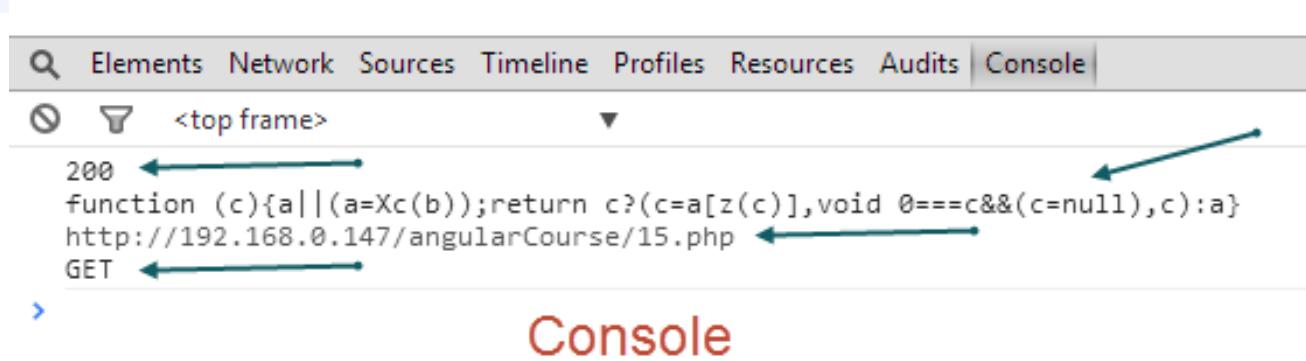
15.php (in my course)

أنيس حكمت أبو حميد (دورة مجانية) 2nees.com

\$http

مثال ٢:

```
// Example code in controller
app.controller('httpController', function($scope, $http) {
  var url = "http://192.168.0.147/angularCourse/15.php";
  $http.get(url).
    success(function(data, status, headers, config) {
      console.log(status + "\n" + headers + "\n" + config.url + "\n" + config.method );
    }).
    error(function(data, status, headers, config) {
      console.log("There is error, error number is: " + status);
    });
});
```



Console

قديماً:

كنت أقول أن الإسلام دين عمل وعبادة، وكان فهمي هذا يندرج من كون العبادة لا تتحقق إلا بالعمل الذي يؤدي لها، أي أن العبادة أصلاً هي عمل.

وأجد الدافع هنا: أن العمل يؤدي إلى تحقيق العبادة (في حال وافق العمل شروط العمل الصالح).
حديثاً (ما أتبناه الآن):

أصبحت أقول أن الإسلام دين عبادة وعمل؟!
وستقول مباشرة ما الفرق؟!

سأقول لك أن فهمي الآن أستخلصه من سورة الفاتحة "إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ" (٥) "ففي تفسير القرطبي قيل " و (نعبد) معناه نطيع ؛ والعبادة الطاعة والتذلل " و " وإياك نستعين أي نطلب العون والتأييد والتوفيق " وفي تفسير الطبرى " إياك نستعين على طاعتك وعلى أمورنا كلها " .. وفي تفسير البغوي " الاستعانة نوع تعبد فكانه ذكر جملة العبادة أولاً ثم ذكر ما هو من تفاصيلها" وهذا أي أنني أقر بربوبية الله تعالى، وحق عبادته، وأطلب منه العون والتأييد والتوفيق لتحقيق تلك العبودية ..

وبهذا يكون فهمي أن الإسلام دين عبادة وعمل أي أنني أطيع الله كما أمرني، وأطلب التوفيق منه، والآن أعمل بما آمنت به وأطعنته لتحقيق الهدف الذي خلقت من أجله ..

وأجد الدافع هنا: أن العبادة بمفهومها الشامل " وليست العبادة بمفهومها الشعائري فقط " هي التي ستقودني للعمل الذي سيكون بدوره عبادة .. والتي ستقودني بالمحصلة لتحقيق مفهوم " عمارة الأرض " ضمن الضوابط الشرعية.

ملاحظة: لا أقول أن المعنى الأول خطأ، ولا أقول أن المعنى الثاني هو الأصح، لكن هذا ما رأيته الآن وأحببت مشاركتكم به، وفقنا الله وإياكم لما يحب ويرضى وغفر لنا ولكم وجميع المسلمين.

مثال ٣: هذا المثال مهم، انتبه جيدا ^_^ (مثال على POST) أو لا سأقوم بعمل تعديل بسيط على 15.php لتصبح بالشكل التالي:

```
<?php
$var = "None";
if(isset($_POST['msg1']) && isset($_POST['msg2'])) {
    $var = $_POST['msg1'] . " " . $_POST['msg2'] . " in 2nees.com ^_^ " . " (POST METHOD Request) ";
}
?>

{
    "records": [
        {
            "Name" : "2nees.com",
            "Price" : "Free",
            "id" : "1"
        },
        ...
        ...
        {
            "Name" : "HTML5",
            "Price" : "30$",
            "id" : "4"
        }
    ],
    "postRes": [
        {
            "aneesWelcome": "<?= $var ?>"
        }
    ]
}
```

كل ما هو مظلل باللون الأصفر هو جديد يجب عليك إضافته للمثال

هذه حافظ على بياناتها .. لا يوجد عليها مشكلة

قمت بوضع النقاط فقط لتقليل حجم الصورة الخاصة بالمثال :P

\$http

الآن قم بإضافة هذه التعديلات على ال JS وال HTML

```
// Post Example
var simpleData = "msg1=Hello&msg2=Man";
$http.post(url, simpleData).
success(function(data, status, headers, config) {
    $scope.postMsg = data.postRes[0].aneesWelcome;
}).
error(function(data, status, headers, config) {
    console.log("There is error, error number is: " + status);
});
```

قم بإضافة هذه الشيفرة البرمجية أسفل المثل 2 (داخل ال CTRL المسمى ..(httpController

```
<div ng-controller="httpController">
<div ng-repeat="x in aneesCourses" class="col-lg-12">
    {{ x.id + ' ^_^ ' + x.Name + ' *_* ' + x.Price}}
</div>
```

15.html

\$http

والآن قم بتطبيق المثال، ولنشاهد النتائج معا ^_..
النتيجة:

1 ^_ 2nees.com *_ Free

2 ^_ Angular *_ 10\$

3 ^_ CSS *_ 20\$

4 ^_ HTML5 *_ 30\$

None

*، ما هذه النتيجة الغير متوقعة !! لماذا لم تظهر الرسالة !

لنقوم باستخدام ال POST يجب علينا القيام بتعديل بسيط على ال
HTTP الخاص بال Header !!؟
كيف يمكنني القيام بذلك ؟

\$http

الطريقة الأولى: هذه الطريقة سيتم تنفيذها على جميع الـ requests التي سيتم طلبها على شكل POST من خلال الـ module المستخدم حالياً...

شاهد الطريقة:

```
// Set Content-Type as a Defaults for all post request
app.config(function($httpProvider) {
    $httpProvider.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded; charset=UTF-8';
});
```

والآن لشاهد النتيجة:

1 ^_^ 2nees.com *_* Free

2 ^_^ Angular *_* 10\$

3 ^_^ CSS *_* 20\$

4 ^_^ HTML5 *_* 30\$

Hello Man in 2nees.com ^_^ (POST METHOD Request)

\$http

الطريقة الثانية: وتستخدم هذه الطريقة لتنفيذ ال header فقط على ال Request ... الحالي

شاهد المثال:

لاحظ أيضاً

هذه الطريقة

التي يمكنها

استخدامها

ويمكنك

تطبيق المحتوى

بالطريقة السابقة

```
// Post Example 2
var simpleData2 = "msg1=ANEES ^__* &msg2=POST 2";
var reqPost = {
  method: 'POST',
  url: url,
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'
  },
  data: simpleData2,
}
$http(reqPost).
success(function(data, status, headers, config) {
  $scope.postMsg2 = data.postRes[0].aneesWelcome;
}).
error(function(data, status, headers, config) {
  console.log("There is error, error number is: " + status);
});
```

\$http

نتيجة تطبيق المثال:

```
ANEES ^ __ * POST 2 in 2nees.com ^_^ (POST METHOD Request)
```

والآن قبل الإنتهاء من الحديث عن هذا الموضوع، رغب بالتنويه لنقطة مهمة وهي أن الصيغة التي تم استخدامها في المثال السابق هي على الشكل التالي:

```
$http(config);
```

شاهد الشريحة التالية والتي تحوي على المعلومات الى ذكرتها وكيف يمكنك أن تقرأ المزيد وتطور من مهاراتك حسب احتياجاتك في

العمل ^ ^

\$http

صورة من موقع [https://docs.angularjs.org/api/ng/service/\\$http](https://docs.angularjs.org/api/ng/service/$http)

Param	Type	Details
config	object	<p>Object describing the request to be made and how it should be processed. The object has following properties:</p> <ul style="list-style-type: none">• method – {string} – HTTP method (e.g. 'GET', 'POST', etc)• url – {string} – Absolute or relative URL of the resource that is being requested.• params – {Object.<string Object>} – Map of strings or objects which will be serialized with the <code>paramSerializer</code> and appended as GET parameters.• data – {string Object} – Data to be sent as the request message data.• headers – {Object} – Map of strings or functions which return strings representing HTTP headers to send to the server. If the return value of a function is null, the header will not be sent. Functions accept a config object as an argument.• transformRequest – {function} – Name of HTTP header to populate with the value taken

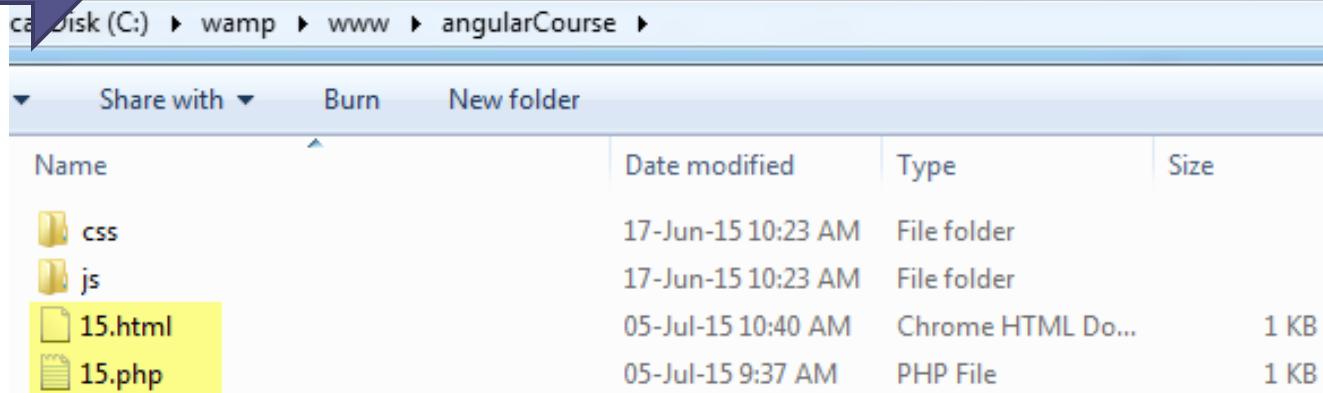
لاحظ أن النوع هنا هو
أوبجك، ولاحظ وجود
عناصر متنوعة، وكيف
يمكننا استخدامها كما
ذكرنا، وهناك المزيد

\$http

والآن، بإذن الله تعالى، نكون قد تحدثنا عن أهم النقاط الخاصة بهذه الموضوع، سائلين الله عز وجل أن تكون قد وفقنا في هذا الطرح.

والآن لنشاهد جميع الأمثلة التي ذكرها في الأمثلة السابقة: (لا تنسى تنفيذ هذا المثال على السيرفر)

شاهد المثال لك HTML



Name	Date modified	Type	Size
css	17-Jun-15 10:23 AM	File folder	
js	17-Jun-15 10:23 AM	File folder	
15.html	05-Jul-15 10:40 AM	Chrome HTML Do...	1 KB
15.php	05-Jul-15 9:37 AM	PHP File	1 KB

مسار المثال

192.168.0.147/angularCourse/15.html

طبعاً لن يعمل المثال لك HTML، وستجد مجموعة من الأخطاء في console، لكن لطلع على الشيفرة البرمجية.
*) مهم أن تقوم بأخذ هذا الملف إلى السيرفر وتطبيقه كما في الصورة.

الإلحاد هو **الجهل** بوجود الله، لا العلم بـعدم وجوده.

مشاركة من تغريدة أرجع لأصلها

التعامل مع الجداول في AngularJS

والآن، وبعد الحديث الشيق الذي سبق هذا الموضوع، نأتي الى موجو
الجدوال والتعامل معها في .. ^ AngularJS ..

طبعا هذا الموضوع سهل جدا .. ^ وهو موضوع للتسليه قبل
الموضوع القادم: P .. فهل أنت مستعد؟ ^

والآن ng-repeat، هل تذكر هذا ال Directive ؟
هذا ال Directive يمثل أفضل طريقة لبناء جدول أو تمثيل الجدول
على صفحة الويب .. كيف هذا ؟!
لنشاهد المثال في الشريحة التالية

التعامل مع الجداول في AngularJS

```

<div ng-controller="tableController as tblCtrl">
  <table class="table">
    <tr>
      <th>Course Name</th>
      <th>Course Price</th>
    </tr>
    <tr ng-repeat="x in tblCtrl.ctableItems">
      <td>{{x.name}}</td>
      <td>{{x.price}}</td>
    </tr>
  </table>

```

```

app.controller('tableController', function() {
  this.ctableItems = coursesTableItem;
});

```

```

var coursesTableItem = [
  {id:154, name: 'AngularJS', price: 'Free'},
  {id:223, name: 'jQuery', price: 'Free'},
  {id:345, name: 'PHP Advance', price: 'Free'},
  {id:454, name: 'CSS3', price: 'Free'},
];

```

مثال ١: لاحظ الـ Ng-repeat وكيف أنها استخدمت مع ..tr

Course Name	Course Price
AngularJS	Free
jQuery	Free
PHP Advance	Free
CSS3	Free

التعامل مع الجداول في AngularJS

مثال ٢: لقد تعلمنا سابقاً كيف يمكننا استخدام الفلتر، ولكن هل يمكننا تطبيق ذلك هنا أيضاً؟

* الجواب بكل تأكيد نعم، شاهد المثال ^

```

<table class="table">
  <tr>
    <th>Course ID</th>
    <th>Course Name</th>
    <th>Course Price</th>
  </tr>
  <tr ng-repeat="x in tblCtrl.ctableItems | orderBy: 'name' ">
    <td>{{x.id}}</td>
    <td>{{x.name | uppercase}}</td>
    <td>{{x.price}}</td>
  </tr>
</table>

```

ترتيب الجدول إعتماداً على اسم المادة، وجعل أحرف المادة uppercase *

Course ID	Course Name	سبب الترتيب	Course Price
154	ANGULARJS	١	Free
454	CSS3	٤	Free
223	JQUERY	٢	Free
345	PHP ADVANCE	٣	Free

التعامل مع الجداول في AngularJS

مثال ٣: الآن، ماذا لو أردنا ترقيم الصفوف داخل الجدول

.. هل سنقوم بإنشاء هذه الأرقام يدوياً؟ (indexing)

لا، لن ينفع هذا.. هل هذا عن طريق ال js أو ال jQuery !؟ أيضاً الجواب لا !! إذن كيف ذلك؟!

الجواب بكل سهولة وجمال من خلال ال **\$index** ... شاهد المثال

```
<table class="table">
  <tr>
    <th>Index (Row Number)</th>
    <th>Course ID</th>
    <th>Course Name</th>
    <th>Course Price</th>
  </tr>
  <tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' ">
    <td>{{$index + 1}}</td>
    <td>{{x.id}}</td>
    <td>{{x.name}}</td>
    <td>{{x.price}}</td>
  </tr>
</table>
```

القيمة الإفتراضية لل
\$index
هي ٠..لذلك نضيف لها ١
حتى يبدأ العد من ١

Index (Row Number)	Course ID	Course Name	Course Price
1	345	PHP Advance	Free
2	223	jQuery	Free
3	454	CSS3	Free
4	154	AngularJS	Free

التعامل مع الجداول في AngularJS

مثال ٤: الآن ما رأيك أن نعطي حركة تنسيق جميلة ، لنقوم بـ **تغيير لون الخلفية للصفوف الزوجية والفردية** ..
كيف هذا؟

بكل بساطة يا صديقي العزيز، من خلال ال \$even وال \$odd
لنشاهد المثال أ:

```
<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' ">
  <td ng-if="$odd" style="background:#777">{{ $index + 1 }}</td>
  <td ng-if="$odd" style="background:#777">{{ x.id }}</td>
  <td ng-if="$odd" style="background:#777">{{ x.name }}</td>
  <td ng-if="$odd" style="background:#777">{{ x.price }}</td>

  <td ng-if="$even" style="background:#c3c3c3">{{ $index + 1 }}</td>
  <td ng-if="$even" style="background:#c3c3c3">{{ x.id }}</td>
  <td ng-if="$even" style="background:#c3c3c3">{{ x.name }}</td>
  <td ng-if="$even" style="background:#c3c3c3">{{ x.price }}</td>
</tr>
```

Index (Row Number)	Course ID	Course Name	Course Price
1	345	PHP Advance	Free
2	223	jQuery	Free
3	454	CSS3	Free
4	154	AngularJS	Free

التعامل مع الجداول في AngularJS

أنيس حكمت أبو حميد (دورة مجانية) 2nees.com

شرح المثال ٤ - أ:

ال ng-if directive هو مثل جملة شرط، فإذا تحقق الشرط سيقوم بتنفيذ العنصر وما فيه، وإلا فإنه لن ينفذ ذلك، ولهذا فإنني قمت بتكرار الأعمدة كما ترى مرة بشكل فردي، ومرة بشكل زوجي، حتى يتم طباعة التنسيق بشكل مختلف حسب ال row (زوجي أم فردي ..). هل هناك طرق أخرى..**لشاهد معا المثال ؟- ب:** (ركز في المثال وتتابع للشريحة التالية لشرحه ..)

```
<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' ">
  <td class="{{ $even ? 'tbl-odd' : 'tbl-even' }}>{{ $index + 1 }}</td>
  <td class="{{ $even ? 'tbl-odd' : 'tbl-even' }}>{{ x.id }}</td>
  <td class="{{ $even ? 'tbl-odd' : 'tbl-even' }}>{{ x.name }}</td>
  <td class="{{ $even ? 'tbl-odd' : 'tbl-even' }}>{{ x.price }}</td>
</tr>
```

التعامل مع الجداول في AngularJS

```
.tbl-odd {  
background: #0f0f0f;  
color: #fff;  
}  
-
```

style.css

```
.tbl-even {  
background: #f0f0f0;  
}  
-
```

Index (Row Number)	Course ID	Course Name	Course Price
1	345	PHP Advance	Free
2	223	jQuery	Free
3	454	CSS3	Free
4	154	AngularJS	Free

شرح المثال ٤-ب: اذا لاحظت في المثال السابق، فإنني قمت باستخدام ال inline Condition داخل الكلاس، بحيث أطبع الكلاس الذي يحدد هل العنصر odd أو even من خلال الشرط ..

التعامل مع الجداول في AngularJS

المثال ٤-ج:

```
<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' ">
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'">{{ $index + 1 }}</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'">{{ x.id }}</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'">{{ x.name }}</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'">{{ x.price }}</td>
</tr>
```

Index (Row Number)	Course ID	Course Name	Course Price
1	345	PHP Advance	Free
2	223	jQuery	Free
3	454	CSS3	Free
4	154	AngularJS	Free

لاحظ هنا أن النتيجة هي ذاتها
لكن هنا قمت باستخدام ال
Ng-class وبهذا لم أضطر
لإستخدام ال {{}} لتنفيذ الشرط
لاحظ كيف يمكن استخدام ال
شكل مترابط ٨٨ AngularJS

التعامل مع الجداول في AngularJS

والآن السؤال الذي يطرح نفسه.. هل تعرف كيف يكون شكل الـ HTML الناتج(كشيفرة برمجية تقرأ من المتصفح) لكل من المثال ٤-ب والمثال ٤-ج؟!

شاهد الفرق: (يمكنك مشاهدته من خلال النقر على F12)

```
<!-- end ngRepeat: x in tblCtrl.ctableItems | orderBy: '-name' -->
▼<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' " class="ng-scope">
  <td class="tbl-odd">3</td>
  <td class="tbl-odd">454</td>
  <td class="tbl-odd">CSS3</td>
  <td class="tbl-odd">Free</td>
</tr>

<!-- end ngRepeat: x in tblCtrl.ctableItems | orderBy: '-name' -->
▼<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' " class="ng-scope">
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'" class="ng-binding tbl-cdd">3</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'" class="ng-binding tbl-cdd">454</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'" class="ng-binding tbl-cdd">CSS3</td>
  <td ng-class="$even ? 'tbl-odd' : 'tbl-even'" class="ng-binding tbl-cdd">Free</td>
</tr>
```

٤-ب

٤-ج

التعامل مع الجداول في AngularJS

المثال ٤-٤: ويمكننا أيضا القيام بنفس العملية دون القيام بوضع أي شرط ..كيف ذلك ؟ ..بكل بساطة من خلال ال Directive التالية:

..كيف يمكننا استخدام (ng-class-even) (ng-class-odd)

هذه ال Directive الجميلة والسهلة ..شاهد المثال

```
<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' ">
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'">{{ $index + 1 }}</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'">{{ x.id }}</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'">{{ x.name }}</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'">{{ x.price }}</td>
</tr>
```

شكل ال الناتج ..

```
▼<tr ng-repeat="x in tblCtrl.ctableItems | orderBy: '-name' " class="ng-scope">
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'" class="ng-binding tbl-odd">3</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'" class="ng-binding tbl-odd">454</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'" class="ng-binding tbl-odd">CSS3</td>
  <td ng-class-odd="'tbl-odd'" ng-class-even="'tbl-even'" class="ng-binding tbl-odd">Free</td>
</tr>
```

التعامل مع الجداول في AngularJS

وبهذا ^٨ .. نكون بإذن الله تعالى .. إنتهينا من هذا الموضوع الجميع والذى تطرقنا به الى حركات جميلة وأفكار متنوعة لكنها سهلة .. وبكل تأكيد، كثير مما تم ذكره يمكنك استخدامه على عناصر أخرى ^{٨٨}

والآن .. لنشاهد معا جميع الأمثلة السابقة ^٨



حَدَّثَنَا عَلِيُّ بْنُ مُحَمَّدٍ قَالَ: حَدَّثَنَا وَكِيعٌ، عَنْ كَهْمَسِ بْنِ الْحَسَنِ، عَنْ عَبْدِ اللَّهِ
بْنِ بُرِيدَةَ، عَنْ عَائِشَةَ، أَنَّهَا قَالَتْ: يَا رَسُولَ اللَّهِ ، أَرَأَيْتَ إِنْ وَافَقْتُ لَيْلَةَ الْقَدْرِ مَا أَذْعُو؟
قَالَ: " تَقُولِينَ: اللَّهُمَّ إِنَّكَ عَفُوٌ تُحِبُّ الْعَفْوَ فَاغْفِفْ عَنِّي "

AngularJS Create New Module

والآن، ننتقل الى الموضوع الاخير بإذن الله تعالى، وأول سؤال يجول بخاطرنا الان، هل يمكننا إنشاء أكثر من module ؟ وهل يمكن استخدام هذه ال module مع بعضها البعض في نفس المشروع؟ وما هي الفائدة المرجوة من هذا الأمر ؟ ^٨ .. للاجابة عن هذه الأسئلة تابع معنا .. P:

أولاً: يمكننا إنشاء أكثر من module، كما يمكننا أيضا استخدامها في نفس المشروع ^{٨٨}

ثانياً: هناك العديد من المزايا لإنشاء أكثر من module وقد يكون من أهمها هو تقسيم العمل والوظائف على شكل ملفات يسهل تتبعها وتحديثها واستخدامها في مشاريع أخرى، وهذه مزية رائعة

AngularJS Create New Module

والآن..لشاهد معا خطوات إنشاء module جديد، وسنقوم باستخدامه في نفس المشروع ..

۱) قم بإنشاء ملف 2neesCourses.js

(٢) الآن سنقوم بوضع الشيفرة البرمجة كما تعلمنا عندما أنشأنا الـ

لأول مرة module ٨٨

في هذه الدورة ..

(لا شيء جديد إلى الآن)

```
action() {
    // Declare Angular Module
    var app = angular.module('2nees-courses', []);
    // Example Used in 17.html
    app.controller('tableController', function(){
        this.ctableItems = coursesTableItem;
    });

    var coursesTableItem = [
        {id:154, name: 'AngularJS', price: 'Free'},
        {id:223, name: 'jQuery', price: 'Free'},
        {id:345, name: 'PHP Advance', price: 'Free'},
        {id:454, name: 'CSS3', price: 'Free'}
    ];
};


```

AngularJS Create New Module

٣) الآن سنقوم بالعودة إلى ملف ال app.js الذي كنا نقوم باستخدامه وسنقوم بإضافة اسم ال module الجديد والذي كان في النقطة السابقة هو `2nees-courses` إلى `[]` الموجودة في تعريف ال `module` شاهد الصورة (ملاحظة لتبسيط الفكرة قمت

بأخذ نسخة مصغرة

من ملف جديد

ووضعها في ملف جديد

حتى يكون المثال

واضحاً...)

```
function() {
  // Declare Angular Module
  var app = angular.module('store', ['2nees-courses']);

  // Declare Controller
  app.controller('StoreController', function(){
    this.product = productItem;
  });

  var productItem = {
    'name': 'angularJs Course',
    'desc': '2nees.com - Free courses',
    'cost': '0$'
  };
}

// Main module
angular.module('store').controller('StoreController', function(){});
```

لنفرض أن هذا هو الملف الرئيسي للمشروع والذى استخدمناه في جميع الأمثلة السابقة (لكن قمت باختصاره في ملف جديد أسميه `app2.js`) وذلك حتى يتضح المثال بشكل أسهل ..

AngularJS Create New Module

٤) والآن، سنذهب لنرى صفحة ال HTML .. طبعا سنقوم باستخدام نفس ال Template الذي قمنا باعتماده لجميع الأمثلة، لكن سيحدث تعديل بسيط وهو إضافة ملف الجافا سكريبت الخاص بال module الجديد بعد ملف ال app.js الرئيسي ... شاهد الصورة:

```
<script type="text/javascript" src="js/angular.min.js"></script> 1  
<script type="text/javascript" src="js/app2.js"></script> 2  
<script type="text/javascript" src="js/2neesCourses.js"></script> 3  
body>  
>
```

انته للترتيب ... فهر مهم

17.html

وبهذا يصبح قالبنا الجديد جاهزا إن شاء الله .. ٨٨ (انتقل للشريحة التالية)

AngularJS Create New Module

والآن لنطبق القالب الجديد في مثال لنرى النتائج ^{٨٨}، طبعا المثال الذي سأضعه هو نفسه الخاص بالجدوال والذي استخدمناه في ال app.js والسبب في ذلك أني مثلا سأستخدم نفس هذه ال CTRL في أكثر من مشروع، أو أريد فصلها لأهميتها وخدماتها وتحديثها بشكل سهل ^{٨٨} لأي أمر تحب .. الآن شاهد الشيفرة الموجودة بصفحة ال HTML .. (في الشريحة التالية)

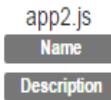
(لا تسنى تطبيق المثال أولا بأول... هذا الموضوع سهل جدا .. لكن يجب عليك تطبيق مثال وتتبعه، وال فكرة الخاصة به سهلة ...)

AngularJS Create New Module

```
app2.js
<div ng-controller="StoreController as store" ng-cloak>
  <div class="row">
    <label class="label label-default col-xs-1">Name</label>
    <span class="col-xs-offset-1">{{store.product.name}}</span>
  </div>
  <div class="row">
    <label class="label label-default col-xs-1">Description</label>
    <span class="col-xs-offset-1">{{store.product.desc}}</span>
  </div>
</div>
```

<hr />

```
2neesCourses.js
<div ng-controller="tableController as tblCtl">
  <table class="table">
    <tr>
      <th>Course ID</th>
      <th>Course Name</th>
      <th>Course Price</th>
    </tr>
    <tr ng-repeat="x in tblCtl.ctableI">
      <td>{{x.id}}</td>
      <td>{{x.name}}</td>
      <td>{{x.price}}</td>
    </tr>
  </table>
</div>
```



angularJs Course
2nees.com - Free courses

نتيجة تطبيق المثال على المتصفح ..
لاحظ أن النتائج ظهرت بشكل طبيعي، ولا يوجد أي مشكلة أو جيد سوى أنني قمت بتنظيم
الстиلاز البرمجية، أيضا يمكنني الآن استخدام الـ

Course ID	Course Name	Course Price
154	AngularJS	Free
223	jQuery	Free
345	PHP Advance	Free
454	CSS3	Free

2nees-courses module
في مشاريع أخرى بكل سهولة ^.^

AngularJS Create New Module

والآن لنشاهد تطبيق آخر مثال في هذه الدورة ٨٨، وأحبنا أن يكون آخر مثال من أسهل الأمثلة والمواضيع ٨٨ لذلك جعلنا هذا الموضوع هنا، ولن يكون نقطة إنطلاق لك للمتابعة والتعلم في أمور أخرى والتعرف على directive آخرى .. وأي من الأمور التي لم يتم التطرق لها، الآن أنت يمكنك فهم أو بناء أي مشروع Angular بسيط بإذن الله تعالى ... الموضوع سهل ورائع وعملي .. ويبقى الدور عليك الآن للتتابع ٨٨



أعوذ بالله من الشيطان الرجيم

وَقُلِ اعْمَلُوا فَسَيَرَى اللَّهُ عَمَلَكُمْ وَرَسُولُهُ وَالْمُؤْمِنُونَ ۖ وَسَتُرَدُّونَ إِلَىٰ عَالِمِ
الْغَيْبِ وَالشَّهَادَةِ فَيُبَيِّنُكُمْ بِمَا كُنْتُمْ تَعْمَلُونَ (١٠٥)

صدق الله العظيم

الخاتمة

وهكذا لكل بداية نهاية ، وخير العمل ما حسن آخره وخير الكلام ما قل ودل وبعد هذا الجهد المتواضع أتمنى أن أكون موفقا في سردي لهذه الدورة المتواضعة، سردا لا ملل فيه ولا تقصير موضحا الخصائص المهمة للغة الـ Angular، وفقني الله وإياكم لما فيه الخير وكل الخير.

وأذكركم إخواني أن هذا العمل عمل بشري، فإن أصبنا فبفضل توفيق من الله تعالى، وإن أخطأنا فمن أنفسنا، نسأل الله العفو والمغفرة لنا ولكم.

وأرجوا من حضرتكم الدعاء لي ولوالدي وللمسلمين، غفر الله لنا ولكم، ورزقنا الله الرزق الحلال الطيب لنا ولكم، وغفر اللهم لوالدينا كما ربونا صغرا ... اللهم آمين.

الخاتمة

ملاحظة:

أذكر أنني أشهد الله تعالى أن هذه الدورة لكل من لا يستخدمها في حرام، وإنما هي دورة لينتفع بها الشخص والمجتمع والأمة الإسلامية ككل، دون إخلال بمبادئنا وقواعدنا .. اللهم نسألك الرزق الحلال الطيب ... ونعود بك من الحرام ما ظهر منه وما بطن.. اللهم آمين.

أحكام: أنيس حكمت أبو حميد.

الموقع الإلكتروني: 2nees.com

الحمد لله صاحب المن والفضل .. الحمد لله رب العالمين.

وآخر دعوانا أن الحمد لله رب العالمين

الْهَمْبِلْ