

Web-App con



Relatore: Leonardo Buscemi



Presentazioni

- Lavoro sulla prototipazione di apparecchiature biomedicali
- Collaboro con diversi web developers professionisti
- Sviluppo algoritmi e software con diversi linguaggi



Indice degli argomenti

- Perché scegliere AngularJS
- Cos'è / Cosa non è
- AngularJS come linguaggio di programmazione
 - Varie piccole applicazioni
- Altre peculiarità esclusive di AngularJS
- Applicazione finale

Perché scegliere AngularJS

*I can trace Angular's beginnings to 2009, on a project called Google Feedback. We'd gone through months of frustration with our development speed and ability to write testable code. At around the six month mark, we had around 17,000 lines of front-end code. At that point, one of the team members, **Misko Hevery**, made a bold statement that **he'd be able to rewrite the whole thing in two weeks** using an open source library that he'd created as a hobby.*

I figured that a two week delay couldn't hurt us that much and we'd at least be entertained by Misko scrambling to build something.

Misko missed his time estimate.

*It took **three weeks**.*

*We were all astounded, but even more astounding was that the line count for this new app had dropped from **17,000** to a mere **1,500**.*

It seemed that Misko was onto something worth pursuing.

"AngularJS" di Brad Green, Shyam Seshadri, O'REILLY, 2013, pg VII

Perché scegliere AngularJS

- L'HTML è nato per creare pagine statiche, AngularJS permette di gestire le viste dinamiche
- Permette di estendere il vocabolario dei tag HTML
- È facile da imparare
- Permette di scrivere meno codice ed in modo più veloce
- Il codice risultante è molto espressivo e leggibile
- Il codice risultante è solido e facilmente testabile
- Fornisce una metodologia di lavoro per app scalabili

Cos'è AngularJS

- ✓ Un framework JS per la scrittura di Web-Apps e SPA
- ✓ Design pattern MVW o MV* → Approfondimento [qui](#)
- ✓ Si utilizza per la scrittura della parte Front-End
- ✓ E' molto utilizzato per applicazioni CRUD e fast prototyping

Cosa NON è AngularJS

- ✗ Per sviluppare la parte Back-End di un sito o applicazione
- ✗ La soluzione a tutti i vostri problemi di programmazione

AngularJS come linguaggio di programmazione

AngularJS estende il codice HTML trasformandolo in un linguaggio di programmazione, in quanto aggiunge:

- Espressioni – Variabili
- Gestione scope – Funzioni
- Istruzioni condizionali
- Cicli

Utilizzo AngularJS

Per utilizzare AngularJS sul nostro sito o web-app è sufficiente:

- Includere la libreria di AngularJS
 - Scaricandone una copia dal [sito](#)
 - Includendo una versione del CDN
- Inserire il tag `ng-app=""`

Template di AngularJS

```
<!DOCTYPE html>
<html ng-app="">
<head lang="en">
  <meta charset="utf-8">

  <title>Titolo della pagina</title>

  <!-- Librerie CSS -->
  <link rel="stylesheet" href="../vendor/foundation/css/foundation.min.css">

  <!-- Libreria di AngularJS -->
  <script src="../vendor/angular.js" type="text/javascript"></script>
  <!-- <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.2/angular.min.js">
  </script> -->

  <!-- Eventuali file JS -->
  <script src="main.js" type="text/javascript"></script>

</head>

<body>

  <div>
  </div>

</body>
</html>
```

Template di AngularJS

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="utf-8">

  <title>Titolo della pagina</title>

  <!-- Librerie CSS -->
  <link rel="stylesheet" href="../vendor/foundation/css/foundation.min.css">

  <!-- Libreria di AngularJS -->
  <script src="../vendor/angular.js" type="text/javascript"></script>
  <!-- <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.2/angular.min.js">
  </script> -->

  <!-- Eventuali file JS -->
  <script src="main.js" type="text/javascript"></script>

</head>

<body>

  <div ng-app="">
  </div>

</body>
</html>
```

Le Espressioni

E' possibile scrivere delle espressioni (aritmetiche o testuali) che AngularJS calcola e visualizza

```
<div ng-app="">
  <h2>Vediamo alcuni esempi di espressioni </h2>
  <h3> {{2+2}} </h3>
  <h3> {{5*5}} </h3>
  <h3> _{{"Hello"+" "+"world"}}_ </h3>
</div>
```

Vediamo alcuni esempi di espressioni
4
25
Hello world

Le Variabili

Possono essere dichiarate direttamente su HTML e sono subito disponibili ad essere utilizzate

```
<div ng-app=""  
  ng-init="testo1='Ciao'; testo2='Mondo';  
          a = 5;  
          b = 10" >  
  
  <h2>Variabili </h2>  
  
  <h3> {{2+2}} </h3>  
  
  <h3> {{5*5}} </h3>  
  
  <h3> {"Hello"+" "+"world"} </h3>  
  
  <h3> {{testo1 + " " + testo2}} </h3>  
  
  <h3> {{ a * b}} </h3>  
  
</div>
```

Variabili

4

25

Hello world

Ciao Mondo

50

Il Binding

Il binding permette di ottenere una sincronizzazione immediata tra la modifica di una variabile e la sua rappresentazione

```
<div ng-app="">
  <h1> Dimostrazione del Binding </h1>

  <label>Name:</label>
  <input type="text" ng-model="yourName" placeholder="Enter a name here">
  <hr>
  <h1>Hello {{yourName}}!</h1>
</div>
```

[Link demo](#)

Il Binding (2 way)

Il binding, inoltre è "2-way", ovvero l'abbinamento tra le variabili è bidirezionale

```
<div ng-app="">
  <h1> Dimostrazione del Binding (2 way)</h1>

  <label>Prima casella di input</label>
  <input type="text" ng-model="testo" placeholder="Scrivere qualcosa qui">

  <label>Seconda casella di input</label>
  <input type="text" ng-model="testo" placeholder="Oppure qui">

  <label>Terza casella di input</label>
  <input type="text" ng-model="testo" placeholder="O ancora qui">

  <h2>Il testo sinora digitato é: {{testo}}</h2>
</div>
```

[Link demo](#)

Il Controller

Problemi di "scope" ? In Javascript è frequente!

Con il Controller è possibile ottenere uno scope differente per una determinata porzione di pagina o applicazione

```
<div ng-app="">
  <h2> Il Controller: esempio 2 </h2>
  <div ng-controller="PrimoController">
    <label>Primo Controller</label>
    <input type="text" ng-model="testo">
  </div>
  <div ng-controller="SecondoController">
    <label>Secondo Controller</label>
    <input type="text" ng-model="testo">
  </div>
</div>
```

Il Controller: esempio 1

Benvenuto AngularJS

Ciao mondo

[Link demo](#)

Le Funzioni

All'interno di un Controller possiamo definire una funzione

```
function PrimoController($scope) {  
  
    $scope.testo = "Benvenuto AngularJS";  
  
    $scope.pulisciTesto = function () {  
        $scope.testo = '';  
    }  
  
    $scope.scriviNuovoTesto = function () {  
        $scope.testo = 'Leonardd';  
    }  
  
    $scope.aggiungiPunto = function () {  
        $scope.testo += '.';  
    }  
  
}
```

```
function SecondoController($scope) {  
  
    $scope.testo = "Ciao mondo";  
  
    $scope.pulisciTesto = function () {  
        $scope.testo = '';  
    }  
  
    $scope.scriviNuovoTesto = function () {  
        $scope.testo = 'Buscemi';  
    }  
  
    $scope.aggiungiPunto = function () {  
        $scope.testo += '!';  
    }  
  
}
```


Le Funzioni

All'interno di un Controller possiamo definire una funzione

```
<div ng-app="">

  <h1> Controllers con Funzioni </h1>

  <div ng-controller="PrimoController">
    <label>Primo Controller</label>
    <input type="text" ng-model="testo">

    <button ng-click="pulisciTesto()"> Cancella il testo </button>
    <button ng-click="scriviNuovoTesto()"> Scrivi nuovo testo </button>
    <button ng-click="aggiungiPunto()"> Aggiungi punto </button>
  </div>

  <div ng-controller="SecondoController">
    <label>Secondo Controller</label>
    <input type="text" ng-model="testo">

    <button ng-click="pulisciTesto()"> Cancella il testo </button>
    <button ng-click="scriviNuovoTesto()"> Scrivi nuovo testo </button>
    <button ng-click="aggiungiPunto()"> Aggiungi punto </button>
  </div>

</div>
```

[Link demo](#)

Istruzioni Condizionali

Esistono alcuni tag di AngularJS che permettono di visualizzare o meno gli elementi a cui sono assegnati

```
<body align="center" ng-app="">
<input id="checkbox" ng-model="checked" type="checkbox">
  <label class="small radius button" for="checkbox">Cliccami</label>

  <div>
    <h2>ng-show</h2>
    <div ng-show="checked">
      <a href="#" class="medium success button">Io vengo mostrato quando la Check-Box viene spuntata</a>
    </div>
  </div>
  <div>
    <h2>ng-hide</h2>
    <div ng-hide="checked">
      <a href="#" class="medium alert button">Io vengo mostrato quando la Check-Box NON viene spuntata</a>
    </div>
  </div>
</body>
</html>
```

[Link demo](#)

Direttive

Sono la peculiarità principale di AngularJS in quanto estendono il vocabolario dei tag HTML, aggiungendo nuove funzionalità.

AngularJS ne fornisce moltissime !

a	ngBindHtml	ngCopy	ngInit	ngMouseup	ngStyle
form	ngBindTemplate	ngCsp	ngKeydown	ngNonBindable	ngSubmit
input	ngBlur	ngCut	ngKeyPress	ngOpen	ngSwitch
input.checkbox	ngChange	ngDbclick	ngKeyUp	ngPaste	ngTransclude
input.email	ngChecked	ngDisabled	ngList	ngPluralize	ngValue
input.number	ngClass	ngFocus	ngModel	ngReadOnly	script
input.radio	ngClassEven	ngForm	ngMouseDown	ngRepeat	select
input.text	ngClassOdd	ngHide	ngMouseenter	ngSelected	textarea
input.url	ngClick	ngHref	ngMouseleave	ngShow	
ngApp	ngCloak	ngIf	ngMouseMove	ngSrc	
ngBind	ngController	ngInclude	ngMouseover	ngSrcset	

Direttive personalizzate

Oltre a tutte le direttive fornite da AngularJS, è possibile anche scriverne di proprie.

Demo non ancora disponibile

```
<body ng-app="DemoApp">
  <div ng-controller="MainCtrl">
    
    <div class="thumbs">
      <icon-list-mouse-movement
        data="images"
        thumb-width="125"
        selected-index="selectedIndex"
        click="iconClick(selectedIndex)">
      </icon-list-mouse-movement>
    </div>
  </div>
</body>
</html>
```

```
var app = angular.module('DemoApp', ['FBAngular']);
function MainCtrl($scope, $http) {
  $scope.selectedIndex = 10;
  $http.get('data/slides.json')
    .success(function(data) {
      $scope.images = data.items;
      $scope.selectedItem =
        $scope.images[$scope.selectedIndex]
    });
  $scope.iconClick = function (index) {
    $scope.selectedItem = $scope.images[index]
  }
}
```

Il ciclo FOR

Un ciclo FOR applicato direttamente sul file HTML

```
<div align="center" ng-app=""
ng-init="letters=['A','B','C','D','E','F','G','H']">
  <h1> Esempio di ciclo </h1>
  <table>
    <tr>
      <th>Elemento</th>
      <th>$index</th>
      <th>$first</th>
      <th>$middle</th>
      <th>$last</th>
      <th>$even</th>
      <th>$odd</th>
    </tr>
    <tr ng-repeat="item in letters">
      <td>{{item}} </td>
      <td>{{ $index }}</td>
      <td>{{ $first }}</td>
      <td>{{ $middle }}</td>
      <td>{{ $last }}</td>
      <td>{{ $even ? '\u2713' : '\u2718' }}</td>
      <td><input type="checkbox" ng-checked="$odd" ></td>
    </tr>
  </table>
</div>
```

Esempio di ciclo

Elemento	\$index	\$first	\$middle	\$last	\$even	\$odd
A	0	true	false	false	✓	<input type="checkbox"/>
B	1	false	true	false	✗	<input checked="" type="checkbox"/>
C	2	false	true	false	✓	<input type="checkbox"/>
D	3	false	true	false	✗	<input checked="" type="checkbox"/>
E	4	false	true	false	✓	<input type="checkbox"/>
F	5	false	true	false	✗	<input checked="" type="checkbox"/>
G	6	false	true	false	✓	<input type="checkbox"/>
H	7	false	false	true	✗	<input checked="" type="checkbox"/>

Il ciclo FOR-EACH

Con la direttiva `ngRepeat` è possibile “ciclare” tutti gli elementi o gli oggetti di un array, emulando il comportamento di un ciclo `for-each`.

```
1 ▼ function AvengersCtrl($scope) {  
2  
3 ▶   $scope.avengers = [  
64   ];  
65  
66 }  
67
```

```
$scope.avengers = [  
  {  
    name: 'Robert Downey Jr.',  
    character: 'Tony Stark / Iron Man'  
  },  
  {  
    name: 'Chris Evans',  
    character: 'Steve Rogers / Captain America'  
  },  
  {  
    name: 'Mark Ruffalo',  
    character: 'Bruce Banner / The Hulk'  
  },  
  {  
    name: 'Chris Hemsworth',  
    character: 'Thor'  
  },  
  {  
    name: 'Scarlett Johansson',  
    character: 'Natasha Romanoff / Black Widow'  
  },  
  {  
    name: 'Jeremy Renner',  
    character: 'Clint Barton / Hawkeye'  
  },  
  {  
    name: 'Tom Hiddleston',  
    character: 'Loki'  
  },  
]
```

Il ciclo FOR-EACH

```
<div ng-app="">
  <div ng-controller="AvengersCtrl">
    <h1> Esempio di filtri </h1>
    <table>
      <tr>
        <th>Attore</th>
        <th>Ruolo</th>
      </tr>
      <tr ng-repeat="actor in avengers">
        <td>{{ actor.name }}</td>
        <td>{{ actor.character }}</td>
      </tr>
    </table>
  </div>
</div>
```

```
{
  name: 'Robert Downey Jr.',
  character: 'Tony Stark / Iron Man'
},
```

[Link demo](#)

I Filtri

Permettono di formattare un dato a seconda delle nostre esigenze.

Si applicano tramite il carattere pipe “|”

```
<div ng-app="">
  <div ng-controller="FiltriCtrl">
    <h1> Demo sui Filtri </h1>

    <h2> Stringhe </h2>
    <h3> Conversione maiuscole-minuscole </h3>
    <h4>{{ testo | lowercase}}</h4>
    <h4>{{ testo | uppercase }}</h4>

    <h3> Numero caratteri o elementi</h3>
    <h4>{{ testo | limitTo:1 }}</h4>
    <h4>{{ testo | limitTo:4 }}</h4>

    <hr><h2> Numeri </h2>
    <h4>{{ numero | number }}</h4>
    <h4>{{ numero | number:0 }}</h4>
    <h4>{{ numero | number:2 }}</h4>

    <hr><h2> Data </h2>
    <h4>{{data | date:'medium'}}</h4>
    <h4>{{data | date:'dd/MM/yyyy @ hh:mm:ss'}}</h4>
    <h4>{{data | date: "'Oggi è il ' dd 'di' MMM '"}}</h4>
  </div>
</div>
```

```
function FiltriCtrl($scope){
  $scope.numero = 1234.56789
  $scope.testo = "aBCdEfGHijkLMn"
  $scope.data = new Date() }
```

[Link demo](#)

I Filtri

Applichiamo dei filtri all'esempio precedente

```
<div ng-app="">
  <div align="center" ng-controller="AvengersCtrl">
    <h1> Esempio di filtri </h1>
    <label>Cerca tra gli Avengers</label>
    <input type="text" ng-model="search" />
    <table>
      <tr>
        <th>Attore</th>
        <th>Ruolo</th>
      </tr>
      <tr ng-repeat="actor in avengers | filter: search">
        <td>{{ actor.name | lowercase }}</td>
        <td>{{ actor.character | uppercase }}</td>
      </tr>
    </table>
  </div>
</div>
```

```
1 function AvengersCtrl($scope) {
2
3   $scope.avengers = [
64 ];
65
66 }
67
```

[Link demo 1](#)

I Filtri

Applichiamo dei filtri all'esempio precedente

```
<label>Cerca su Tutto</label>
<input type="text" ng-model="search.$" />
</div>
<div class="large-4 medium-4 columns">
  <label>Cerca solo Attori</label>
  <input type="text" ng-model="search.name" />
</div>
<div class="large-4 medium-4 columns">
  <label>Cerca solo Ruoli</label>
  <input type="text" ng-model="search.character" />
</div>
</div>
</div>
<table align="center">
  <tr>
    <th>Attore</th>
    <th>Ruolo</th>
  </tr>
  <tr ng-repeat="actor in avengers | filter: search">
    <td>{{ actor.name | lowercase }}</td>
    <td>{{ actor.character | uppercase }}</td>
  </tr>
</table>
```

```
1 function AvengersCtrl($scope) {
2
3   $scope.avengers = [
64 ];
65
66 }
67
```

[Link demo 2](#)

I Servizi di AngularJS

Un servizio è una sorta di funzione o oggetto che effettua uno specifico compito.

AngularJS fornisce una serie di servizi per le operazioni comuni alle web-apps, ma è possibile creare servizi personalizzati.

\$anchorScroll	\$http	\$q
\$animate	\$httpBackend	\$rootScope
\$cacheFactory	\$interpolate	\$scope
\$compile	\$interval	\$sceDelegate
\$controller	\$locale	\$templateCache
\$document	\$location	\$timeout
\$exceptionHandler	\$log	\$window
\$filter	\$parse	

Il servizio \$http

Per utilizzare un servizio di AngularJS è sufficiente "iniettarlo" all'interno di un controller, AngularJS si occuperà del resto

```
function ProvCtrl($scope, $http){  
  
    // Load a JSON  
    $http.get('tutte_le_province.json')  
        .success(function(data) {  
            $scope.province = data.province;  
  
            $scope.numProv = $scope.province.length;  
            $scope.quantità = $scope.numProv;  
        });  
  
};
```

Dependency injection è un design pattern della programmazione orientata agli oggetti il cui scopo è quello di **semplificare lo sviluppo** e **migliorare la testabilità** di software di grandi dimensioni.

http://it.wikipedia.org/wiki/Dependency_injection

Il servizio \$http

Per utilizzare un servizio di AngularJS è sufficiente "iniettarlo" all'interno di un controller, AngularJS si occuperà del resto

[Link demo](#)

```
{
  "ID": "25",
  "Cod_RPart": "2",
  "RPart_Geo": "Nord-est",
  "Cod_Reg": "05",
  "Nome_Reg": "Veneto",
  "Cod_Prov": "25",
  "Nome_Prov": "Belluno",
  "Targa": "BL"
},
```

```
<div ng-app="">
  <h1> Esempio di servizio $http </h1>
  <div ng-controller="ProvCtrl">
    <table>
      <tr>
        <th>Nome Provincia</th>
        <th>Sigla automobilistica</th>
        <th>Regione</th>
        <th>Zona Geografica</th>
      </tr>
      <tr ng-repeat="prov in province">
        <td>{{ prov.Nome_Prov }}</td>
        <td>{{ prov.Targa }}</td>
        <td>{{ prov.Nome_Reg }}</td>
        <td>{{ prov.RPart_Geo }}</td>
      </tr>
    </table>
  </div>
</div>
```

Il servizio personalizzato

Per creare un servizio personalizzato dobbiamo prima dichiarare la nostra "App" ed il relativo "Module"

```
main_factory.js x
1  var myApp = angular.module('NomeApp', []);
2
3  myApp.factory('Avengers', function () {
68 });
69
70 function AvengersCtrl($scope, Avengers) {
71     $scope.avengers = Avengers;
72 }
73
```

```
<div ng-app="NomeApp">
```

[Link demo](#)

Applicazione finale

Costruiamo un'applicazione che permette la ricerca delle province italiane con le relative sigle automobilistiche, le regione e le aree geografiche.

[Link demo](#)

Fonti

1. Community [AngularJS Italia](#) su Google+
2. Sito ufficiale di [AngularJS](#)
3. John Lindquist su [YouTube](#) oppure su [egghead.io](#)
4. Altre fonti interessanti [Thinkster.io](#)

Grazie per l'attenzione

Domande ?