

User Defined Filters

EX-1a (Simple filter exercise)

Given this controller:

```
app.controller('MyController', [function() {
  var self = this;
  self.persons = [
    {name: 'Hans', gender: 'male', age: 8}, {name: 'Grethe', gender: 'female', age: 7},
    {name: 'Frederik', gender: 'male', age: 61}, {name: 'Hassan', gender: 'male', age: 13},
    {name: 'Karen', gender: 'female', age: 31}, ]
  }]);
```

Create a custom filter to show only children of school age (6-15 years old)

Hint: look at this tutorial: <https://scotch.io/tutorials/building-custom-angularjs-filters#filters-that-actually-filter>

EX-1b (Optional exercise of more complex filter)

Given this style:

```
<style>
  table {border-collapse: collapse; }
  td { border: 1px solid black; height: 30px; width: 30px; font-size: 28px; }
  .dark { background: lightgray; }
</style>
```

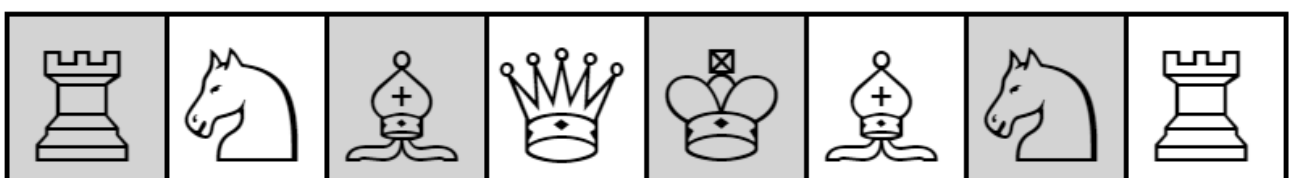
And this table:

```
<table ng-controller="ChessCtrl">
  <tr>
    <td ng-class="{ 'dark': $seven}" ng-repeat="sq in board track by $index">{{sq | Chess}}</td>
  </tr>
</table>
```

And this controller:

```
app.controller("ChessCtrl", function($scope) {
  $scope.board = ["wr", "wkn", "wb", "wq", "wk", "wb", "wkn", "wr"];
});
```

Implement a "Chess" filter that will render the table as sketched below:



Hints:

Custom Filters: https://docs.angularjs.org/tutorial/step_09

Unicode: <http://character-code.com/special-character-html-codes.php>

User Defined Directives

Ex-2a

Directives are one of the coolest things in AngularJS. Here we will only scratch the surface and write a simple Template-expanding Directive. Se <https://docs.angularjs.org/guide/directive> for all the details related to user defined directives.

Given this controller:

```
app.controller("DirectiveController", ['$scope', function($scope) {
    $scope.user = {
        companyName : "Coolest Company on Earth",
        companyUrl   : "http://www.cool.cooler.com",
        created      : new Date()
    }
}])
```

Write a user defined directive which when used like this from our html:

```
<format-company></format-company>
```

Or like this:

```
<div format-company></div>
```

Will produce the following output:

Company:	Coolest Company on Earth
URL:	Visit us
Created:	Nov 10, 2014

Use the first example in the link above, as a template for your solution.

Ex-2b

Create a directive: that can insert a gallery of pictures.

Put the pictures in a folder on the rootfolder: "web pages".

Create a directive that can add pictures from the folder based on comma seperated string like:

```
<div myDirective="img1, img2, img3, img4"></div>
```

Hint:

- 1) Use the link property of the directive to manipulate the DOM.
- 2) Use this example to get you started:
<https://github.com/Thomas-Hartmann/angulardemo/blob/master/web/directiveDemo.html>

Services and factories

Ex-3

Write a service or factory with two methods:

- getRandomNumber(n) → Should return a random number between 1 and n, both inclusive
- getRandomString(n) → Should return a random string of length n.

Implement a Controller that should use (inject) the service and expose the following properties:

- \$scope.randomNumber
- \$scope.randomString

Implement an html file that should present the two values.

Hint: The following code will produce a random string of length N¹

```
new Array(N+1).join((Math.random().toString(36)+'0000000000000000').slice(2, 18)).slice(0, N)
```

Ex-4

a)

Write a factory which should fetch country data from restcountries.eu using the url's below:

- Get all countries: <https://restcountries.eu/rest/v1/all>
- Get all countries from region: <https://restcountries.eu/rest/v1/region/africa>
- Get all country from code: <https://restcountries.eu/rest/v1/alpha?codes=gb>

b)

Implement a controller with methods that use the factory created above and sets relevant "binding-properties" holding the results.

c)

Implement one or more html pages that use the controller implemented above.

Hint: Use this article as a reference for how to implement the factory for both this, and the next exercise.

<http://weblogs.asp.net/dwahlin/using-an-angularjs-factory-to-interact-with-a-restful-service>

Ex-5

Change the exercises from the exercise set *ExercisesAngularWithBackend.pdf* so that all REST traffic is encapsulated in a factory or service.

Hint: Use this article as a reference:

<http://weblogs.asp.net/dwahlin/using-an-angularjs-factory-to-interact-with-a-restful-service>

¹ Ref: <http://stackoverflow.com/questions/1349404/generate-a-string-of-5-random-characters-in-javascript>

Ex-6

exam-preparation_Angular2_FiltersDirectivesFactories.pdf

