Agenda: Filters

- Purpose of Filters
- Syntax
- Built-In Filters
- Uppercase and Lowercase Filters
- Currency and Number Formatting Filters
- OrderBy Filter
- Filter Filter
- Creating Custom Filter

Purpose of Filters

A filter formats the value of an expression for display to the user.

Filters can be added to expressions and directives using a pipe character.

AngularJS filters can be used to transform data:

Filter	Description
uppercase	Returns the upper case of the string.
lowercase	Returns the lower case of the string
orderBy: 'Field'	Orders an array by an expression. Should be used with ng-repeat only
currency	Converts a number to a currency format.
number : digitsAfterDecimal	Number rounded to decimalPlaces and places a “,” after each third digit
filter : model	Select a subset of items from an array as matted in the model.
date : format	Returns a date as per specified format
json : spaces	Allows you to convert a JavaScript object into JSON string. Default spacing is 2 spaces

Syntax

Filters can be applied to expressions in view templates using the following syntax:

{{ expression | filter }}

Chaining of Filtes

{{ expression | filter1 | filter2 | filter3 ……}}

If Filters containing any arguments

{{ expression | filter :arg1:arg2:arg3:…..}}

Example to demonstrate all filters

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title></title>
   <script src="angular.js"></script>
</head>
<body>
     <div ng-app="myApp" ng-controller="studentController">
     <input type="text" name="txtName"  ng-model="nameFilter" />
      <table>
         <tr>
            <th>Id</th>
            <th>Name</th>
            <th>Marks</th>
            <th>Fees Paid</th>
         </tr>
         <tr ng-repeat="stud in students |  filter:nameFilter  | orderBy : 'Marks'">
            <td>{{stud.Id}}</td>
            <td>{{stud.Name | uppercase}}</td>
            <td>{{stud.Marks}}</td>
            <td>{{stud.FeesPaid | currency}}</td>
         </tr>
      </table>
   </div>
   <script>
      var myApp = angular.module('myApp', []);
      myApp.controller('studentController', ['$scope', function ($scope) {
         $scope.students = [{ Name: "Ravi", Id: 1, Marks: 90, FeesPaid:1000 },
                            { Name: "Sam", Id: 2, Marks: 50, FeesPaid:2000 },
                            { Name: "Abdul", Id: 3, Marks: 80, FeesPaid:500 },
                            { Name: "Rajesh", Id: 4, Marks: 60, FeesPaid:900 }];
      }]);
   </script>
</body>
</html>
```

Date Examples:

In controller:

```
$scope.GetDate = function(str) {
   return new Date(str)
}
```

In Html Template:

```
Enter Date: <input type="text" ng-model="SomeDate" ng-init="SomeDate='12/15/2015 10:15:30 PM'" /> <br />
Medium Format: {{GetDate(SomeDate) | date: 'medium'}}  <br />
yyyy-MM-dd HH:mm:ss Z Format: {{GetDate(SomeDate) | date:'yyyy-MM-dd HH:mm:ss Z'}} <br />
dd/MMM/yyyy @ h:mma Format: {{GetDate(SomeDate) | date:'dd/MMM/yyyy @ h:mma'}}
```

Output:

Enter Date: 12/15/2015 10:15:30 PM
Medium Format: Dec 15, 2015 10:15:30 PM
yyyy-MM-dd HH:mm:ss Z Format: 2015-12-15 22:15:30 +0530
dd/MMM/yyyy @ h:mma Format: 15/Dec/2015 @ 10:15PM

Number formatting:

```
<input type="text" ng-model="SomeNumber" /> <br />
Default formatting: {{SomeNumber | number}}  <br />
No fractions: {{SomeNumber | number: 0}} <br />
Negative numbers: {{-SomeNumber | number: 4}} <br />
```

Output:

Enter number: 1234.56789
Default formatting: 1,234.568
No fractions: 1,235
Negative number: -1,234.5679

JSON Filter: Allows you to convert a JavaScript object into JSON string

Its mostly used for debugging:

```
<pre>{{ {'name':'value'} | json }}</pre>
<pre> {{ {'name':'value'} | json:4 }}</pre>
```

Custom Filters

To write a custom filter we need to register a **new filter factory function** with the module.

This factory function should return a **new filter function** which takes the input value as the first argument. Any filter arguments are passed in as additional arguments to the filter function.

Note: filter names must be valid angular expression identifiers, such as uppercase or orderBy. Names with special characters, such as hyphens and dots, are not allowed.

Static Custom Filter:

The following sample filter reverses a text string. In addition, it conditionally makes the text upper-case

```
myApp.filter('reverse', function () {
   return function (input, uppercase) {
      input = input || '';
      var out = "";
      for (var i = 0; i < input.length; i++) {
         out = input.charAt(i) + out;
      }
      // conditional based on optional argument
      if (uppercase) {
         out = out.toUpperCase();
      }
      return out;
   };
})
<input type="text" ng-model="SomeText" /> <br />
Actual Value: {{SomeText}}  <br />
Reversed Value: {{SomeText | reverse}} <br />
Reversed Uppercase Value: {{SomeText | reverse: true}} <br />
```

Custom Filter for repeats:

```
myApp.filter('Passed', function () {
   return function (studs, passingMarks) {
      if (typeof passingMarks == 'undefined')
         passingMarks = 35;
      var filteredStuds = [];
      for (var i = 0; i < studs.length; i++) {
         if (studs[i].Marks > passingMarks)
            filteredStuds.push(studs[i]);
      }
      return filteredStuds;
   }
})
<tr ng-repeat="stud in students |  Passed">
or
<tr ng-repeat="stud in students |  Passed: 60 ">
```

Custom Filter for Displaying text in CamelCase Format

```
var app = angular.module('myApp', []);
app.filter('myFilter', function () {
        return function (x) {
            var i,txt = "";
            for (i = 0; i < x.length; i++) {
                if (x[i] == " ") {
                    i++;
                    txt += " "+x[i].toUpperCase();
                }
                else {
                    txt += x[i];
                }
            }
            txt = txt.substring(0, 1).toUpperCase() + txt.substring(1);
            return txt;
        };
    });
    app.controller('namesCtrl', function ($scope) {
        $scope.names = [
            'Jani Wool',
            'carl',
            'hello margareth',
            'Hege rooman rellex',
            'Joe',
        ];
    });
<ul ng-app="myApp" ng-controller="namesCtrl">
    <li ng-repeat="x in names">
      {{x | myFilter}}
    </li>
  </ul>
```