



SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approved by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai



Department of MCA

AngularJS Filters



Course: 19CAT901 – Web Programming Essentials

Unit IV : Advanced Java Script

I Semester / I MCA



AngularJS Filters



Filters can be added in AngularJS to format data.

AngularJS Filters

AngularJS provides filters to transform data:

currency Format a number to a currency format.

date Format a date to a specified format.

filter Select a subset of items from an array.

json Format an object to a JSON string.

limitTo Limits an array/string, into a specified number of elements/characters.

lowercase Format a string to lower case.

number Format a number to a string.

orderBy Orders an array by an expression.

uppercase Format a string to upper case.



Adding Filters to Expressions



Filters can be added to expressions by using the pipe character |, followed by a filter.

The uppercase filter format strings to upper case:

Example

```
<div ng-app="myApp" ng-controller="personCtrl">
```

```
<p>The name is {{ lastName | uppercase }}</p>
```

```
</div>
```



Adding Filters to Expressions



Filters are added to directives, like ng-repeat, by using the pipe character |, followed by a filter:

Example

The orderBy filter sorts an array:

```
<div ng-app="myApp" ng-controller="namesCtrl">
```

```
<ul>
```

```
  <li ng-repeat="x in names | orderBy:'country'">
    {{ x.name + ', ' + x.country }}
  </li>
```

```
</ul>
```

```
</div>
```

```
</div>
```



Adding Filters to Expressions



```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="myApp" ng-controller="namesCtrl">
<p>Looping with objects:</p>
<ul>
<li ng-repeat="x in names | orderBy:'country'">
  {{ x.name + ', ' + x.country }}
</li>
</ul>
</div>
<script>
angular.module('myApp', []).controller('namesCtrl', function($scope) {
  $scope.names = [
    {name:'Jani',country:'Norway'},
    {name:'Carl',country:'Sweden'},
    {name:'Margareth',country:'England'},
    {name:'Hege',country:'Norway'},
    {name:'Joe',country:'Denmark'},
    {name:'Gustav',country:'Sweden'},
    {name:'Birgit',country:'Denmark'},
    {name:'Mary',country:'England'},
    {name:'Kai',country:'Norway'}
  ];
});
</script></body></html>
```

Looping with objects:

Joe, Denmark

Birgit, Denmark

Margareth, England

Mary, England

Jani, Norway

Hege, Norway

Kai, Norway

Carl, Sweden

Gustav, Sweden



Adding Filters to Expressions



The currency Filter

The currency filter formats a number as currency:

Example

```
<div ng-app="myApp" ng-controller="costCtrl">
```

```
<h1>Price: {{ price | currency }}</h1>
```

```
</div>
```

Price: \$58.00

The currency filter formats a number to a currency format.



Adding Filters to Expressions



The lowercase filter format strings to lower case:

Example

```
<div ng-app="myApp" ng-controller="personCtrl">
```

```
<p>The name is {{ lastName | lowercase }}</p>
```

```
</div>
```

The name is DOE



AngularJS Filter



The filter Filter

The filter filter selects a subset of an array.

The filter filter can only be used on arrays, and it returns an array containing only the matching items.

Example

Return the names that contains the letter "i":

```
<div ng-app="myApp" ng-controller="namesCtrl">
<ul>
  <li ng-repeat="x in names | filter : 'i'">
    {{ x }}
  </li>
</ul>
</div>
```

Jani
Birgit
Kai

This example displays only the names containing the letter "i".



AngularJS Filter an Array Based on User Input



By setting the ng-model directive on an input field, we can use the value of the input field as an expression in a filter.

Type a letter in the input field, and the list will shrink/grow depending on the match:

Jani

Carl

Margareth

Hege

Joe

Gustav

Birgit

Mary

Kai



AngularJS Filter an Array Based on User Input



```
<div ng-app="myApp" ng-controller="namesCtrl">  
  
<p><input type="text" ng-model="test"></p>  
  
<ul>  
  <li ng-repeat="x in names | filter : test">  
    {{ x }}  
  </li>  
</ul>  
  
</div>
```

Type a letter in the input field:

- Jani
- Carl
- Margareth
- Hege
- Joe
- Gustav
- Birgit
- Mary
- Kai

The list will only consists of names matching the filter.

Type a letter in the input field:

- Kai

The list will only consists of names matching the filter.



Sort an Array Based on User Input



Click the table headers to change the sort order::

Name	Country
Jani	Norway
Carl	Sweden
Margareth	England
Hege	Norway
Joe	Denmark
Gustav	Sweden
Birgit	Denmark
Mary	England
Kai	Norway

By adding the `ng-click` directive on the table headers, we can run a function that changes the sorting order of the array:



AngularJS Directives



```
<script>
angular.module('myApp',
[ ]).controller('namesCtrl', function($scope) {
  $scope.names = [
    {name: 'Jani', country: 'Norway'},
    {name: 'Carl', country: 'Sweden'},
    {name: 'Margareth', country: 'England'},
    {name: 'Hege', country: 'Norway'},
    {name: 'Joe', country: 'Denmark'},
    {name: 'Gustav', country: 'Sweden'},
    {name: 'Birgit', country: 'Denmark'},
    {name: 'Mary', country: 'England'},
    {name: 'Kai', country: 'Norway'}
  ];
  $scope.orderByMe = function(x) {
    $scope.myOrderBy = x;
  };
});
</script>
```



AngularJS Filters



Output

Click the table headers to change the sorting order:

Name	Country
Birgit	Denmark
Carl	Sweden
Gustav	Sweden
Hege	Norway
Jani	Norway
Joe	Denmark
Kai	Norway
Margareth	England
Mary	England



Custom Filters



You can make your own filters by registering a new filter factory function with your module:

Example

Make a custom filter called "myFormat":



Custom Filters



```
<ul ng-app="myApp" ng-controller="namesCtrl">
  <li ng-repeat="x in names">
    {{x | myFormat}}
  </li>
</ul>
```

```
<script>
```

```
var app = angular.module('myApp', []);
app.filter('myFormat', function() {
```

```
  return function(x) {
    var i, c, txt = "";
    for (i = 0; i < x.length; i++) {
      c = x[i];
      if (i % 2 == 0) {
        c = c.toUpperCase();
      }
      txt += c;
    }
    return txt;
  };
});
```

```
app.controller('namesCtrl', function($scope) {
  $scope.names = ['Jani', 'Carl', 'Margareth', 'Hege', 'Joe', 'Gustav', 'Birgit', 'Mary', 'Kai'];
});
</script>
```

- JaNi
- CaRl
- MaRgArEtH
- HeGe
- JoE
- GuStAv
- BiRgIt
- MaRy
- Kai

Make your own filters.

This filter, called "myFormat", will uppercase every other character.



References



https://www.w3schools.com/angular/angular_filters.asp

<https://www.javatpoint.com/angularjs-filters>