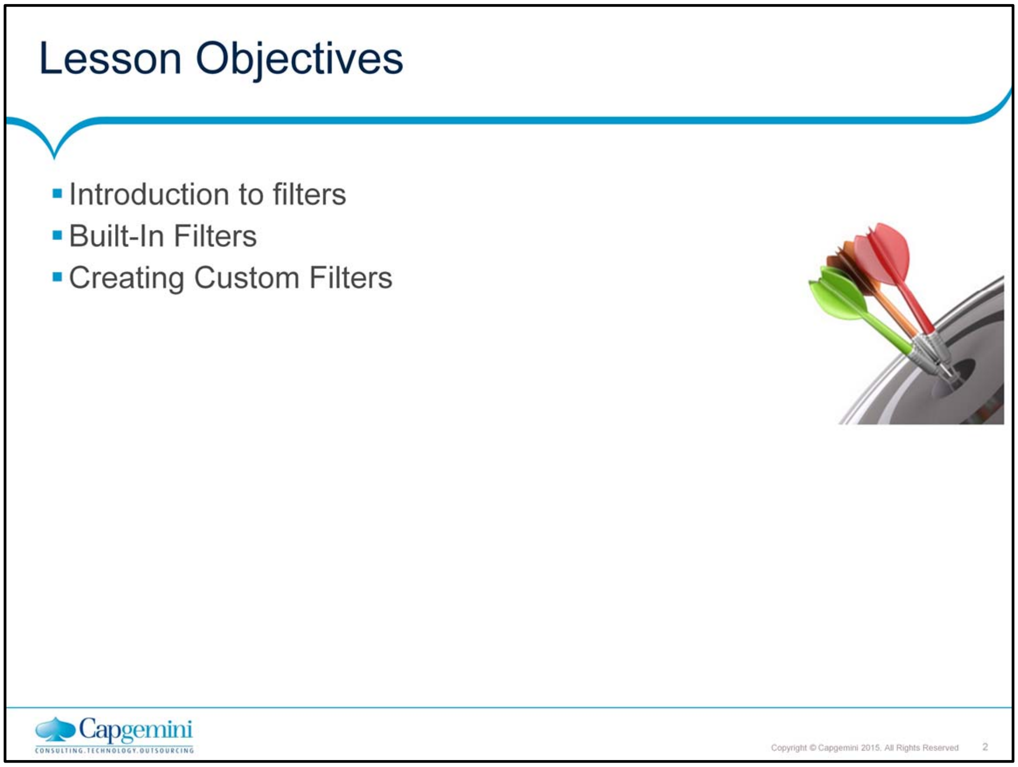
AngularJS

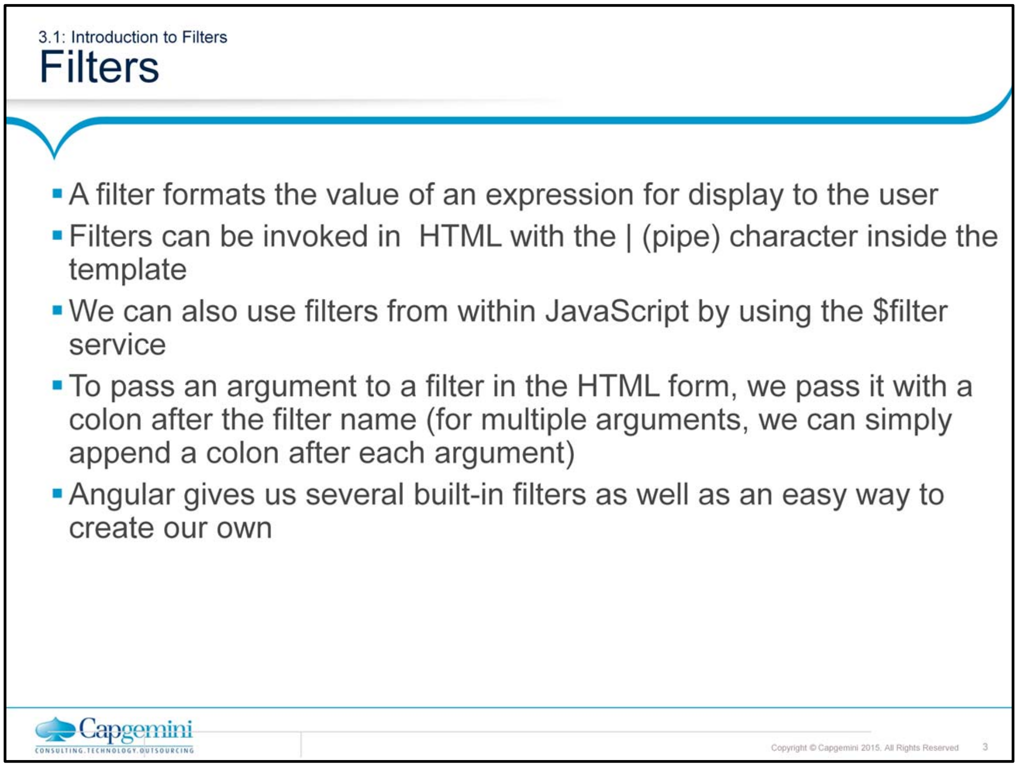
AngularJS Filters



Lesson Objectives

- Introduction to filters
- Built-In Filters
- Creating Custom Filters

2



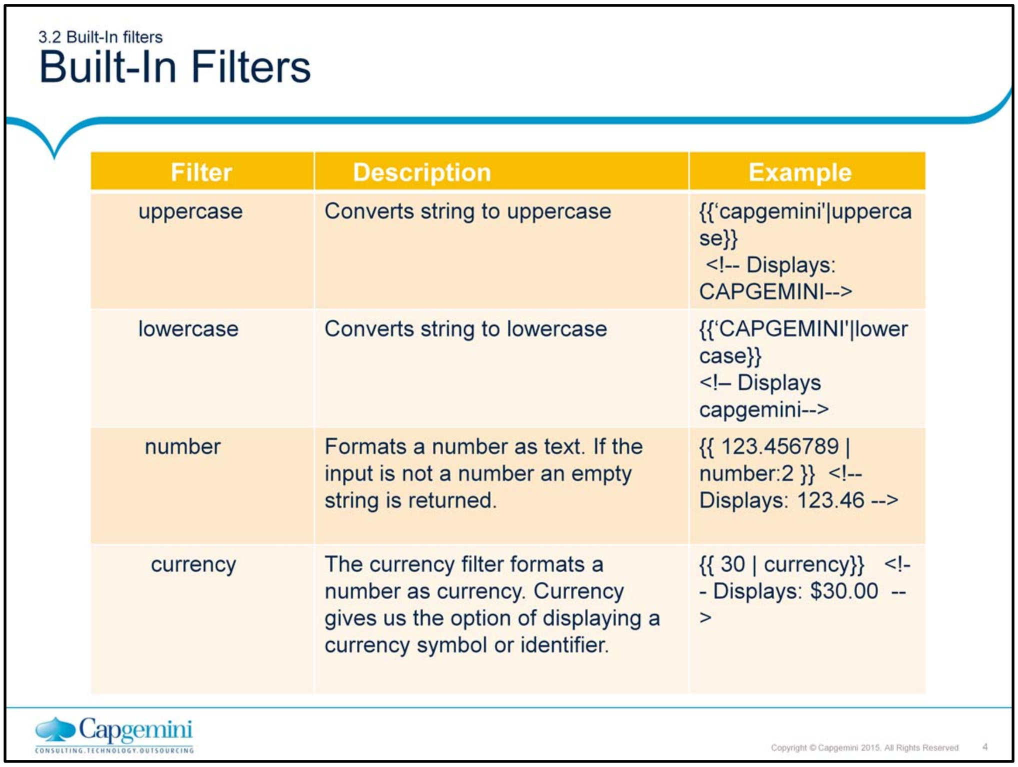
3.1: Introduction to Filters

Filters

- A filter formats the value of an expression for display to the user
- Filters can be invoked in HTML with the | (pipe) character inside the template
- We can also use filters from within JavaScript by using the $filter service
- To pass an argument to a filter in the HTML form, we pass it with a colon after the filter name (for multiple arguments, we can simply append a colon after each argument)
- Angular gives us several built-in filters as well as an easy way to create our own

A filter formats the value of an expression for display to the user. They can be used in view templates, controllers or services and it is easy to define your own filter.
Filters can be applied to expressions in view templates using the following syntax:

{{ expression | filter }}
{{ expression | filter1 | filter2 | ... }}
{{ expression | filter:argument1:argument2:... }}



3.2 Built-In filters

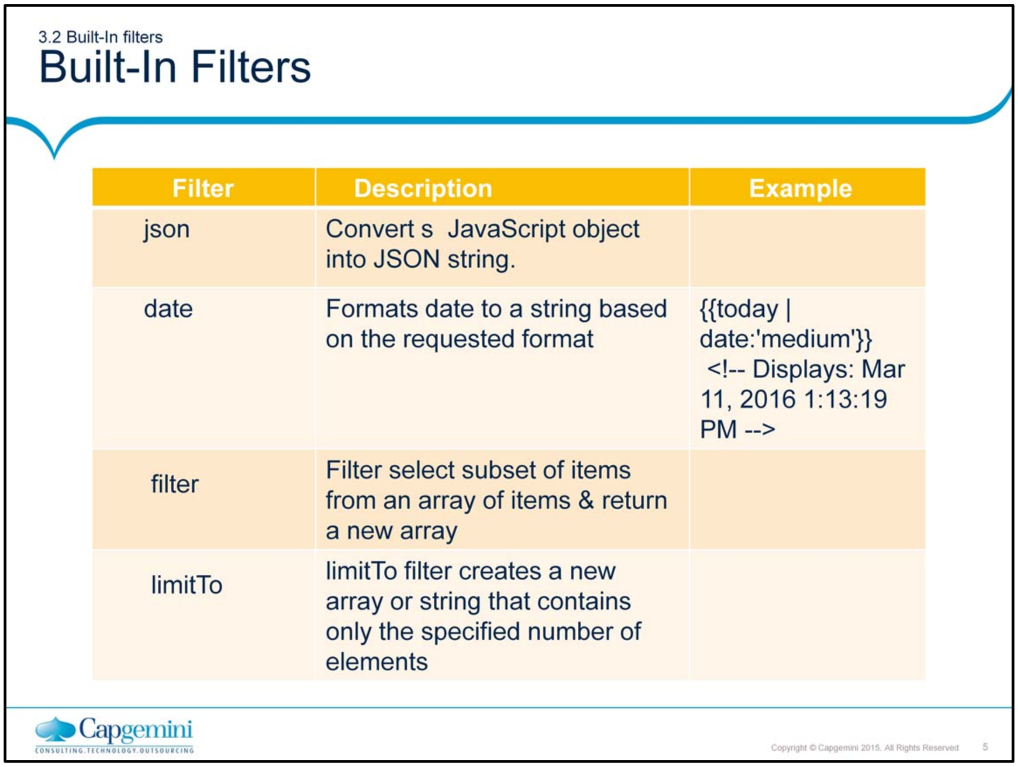
Built-In Filters

Filter	Description	Example
uppercase	Converts string to uppercase	{{'capgemini'|uppercase}} <!-- Displays: CAPGEMINI-->
lowercase	Converts string to lowercase	{{'CAPGEMINI'|lowercase}} <!– Displays capgemini-->
number	Formats a number as text. If the input is not a number an empty string is returned.	{{ 123.456789 | number:2 }} <!-- Displays: 123.46 -->
currency	The currency filter formats a number as currency. Currency gives us the option of displaying a currency symbol or identifier.	{{ 30 | currency}} <!-- Displays: $30.00 -->

Currency :The default currency option is that of the current locale we can also pass in a currency to display.

{{ 30 | currency}} <!-- Displays: $30.00 -->

{{ 30 | currency : "Rs."}} <!-- Displays: Rs.30.00 -->



3.2 Built-In filters

Built-In Filters

Filter	Description	Example
json	Convert s JavaScript object into JSON string.	
date	Formats date to a string based on the requested format	{{today | date:'medium'}} <!-- Displays: Mar 11, 2016 1:13:19 PM -->
filter	Filter select subset of items from an array of items & return a new array	
limitTo	limitTo filter creates a new array or string that contains only the specified number of elements	

Copyright © Capgemini 2015. All Rights Reserved 5

Json Filter This filter is mostly useful for debugging

{{{ 'Id':714709,'Name':‘ABCD'} | json }}

<!-- Displays: { "Id": 714709, "Name": “ABCD"} -->

date Filter

Date to be formatted can be either a Date object or milliseconds (string or number)

The date formatter provides us several built-in options. If no date format is passed, then it set to the default mediumDate.

```
{{today | date:'medium'}} <!-- Displays: Jul 30, 2014 5:13:19 PM -->
{{today | date:'short'}} <!-- Displays: 7/30/14 5:13 PM -->
{{today | date:‘fullDate'}} <!-- Displays: Wednesday, July 30, 2014-->
{{today | date:‘longDate'}} <!-- Displays: July 30, 2014-->
{{today | date:'mediumDate'}} <!-- Displays: Jul 30, 2014-->
{{today | date:‘shortDate'}} <!-- Displays: 7/30/14 -->
{{today | date:‘mediumTime'}} <!-- Displays: 5:22:21 PM -->
{{today | date:‘mediumTime'}} <!-- Displays: 5:23 PM -->
{{today | date:'d-M-y'}}  <!-- Displays: 30-7-2014 -->
{{today | date:'d-M-yyyy'}} <!-- Displays: 30-7-2014 -->
{{today | date:'dd-MM-yy'}} <!-- Displays: 30-07-14 -->
{{today | date:'EEEE dd, MMMM yyyy'}}  <!-- Displays: Wednesday 30, July 2014 -->
{{today | date:'EEE dd MMM yyyy'}}  <!-- Displays: Wed 30 Jul 2014 -->
{{ today | date:'hh:mm:ss.sss a' }}  <!-- Displays: 05:35:31.951 PM  -->
{{ today | date:'hh:mm:ss a' }}   <!-- Displays: 05:35:31 PM  -->
{{ today | date:'H:m:s a' }} <!-- Displays: 17:35:31 PM  -->
{{ today | date:'Z' }}  <!-- Displays: +0530 -->
```

```
<div >
<span><b>Printing Data FOR FILTER: </b></span>
<span ng-repeat="product in products | filter:1000 ">{{product.name }}</span>
</div>
```

Example2:

filter based on a string :

```
{{ ['Anil', 'Latha', 'Mahima', 'Sachin', 'Veena'] | filter:'e' }}
<!-- returns array ["Veena"]        ‘!e’ returns other than ‘Veena’     -->
```

filter based on a function:

```
    angular.module('filterApp',[])
    .controller('FilterController', function($scope,$filter) {
                    var pattern = /^\d{6}$/;
                    $scope.getSixDigitsPattern = function(item) {
                                return pattern.test(item);
                    }
    });
{{ ['714709', '562A', '044-235', '801234','ABC'] | filter:getSixDigitsPattern }}
<!-- return array ["714709","801234"] -->
```

filter based on a object:

```
{{
     [
               {"Id":1,"Name":"Anil","Location":"Mumbai"},

               {"Id":2,"Name":"Latha","Location":"Bangalore"},

               {"Id":3,"Name":"Mahima","Location":"Pune"},

               {"Id":4,"Name":"Sachin","Location":"Mumbai"},

               {"Id":5,"Name":"Veena","Location":"Pune"}
        ]  | filter:{"Location":"Mumbai"}
}}
<!-- returns array
[
        {"Id":1,"Name":"Anil","Location":"Mumbai"},
        {"Id":4,"Name":"Sachin","Location":"Mumbai"}
]
-->
```

limitTo:

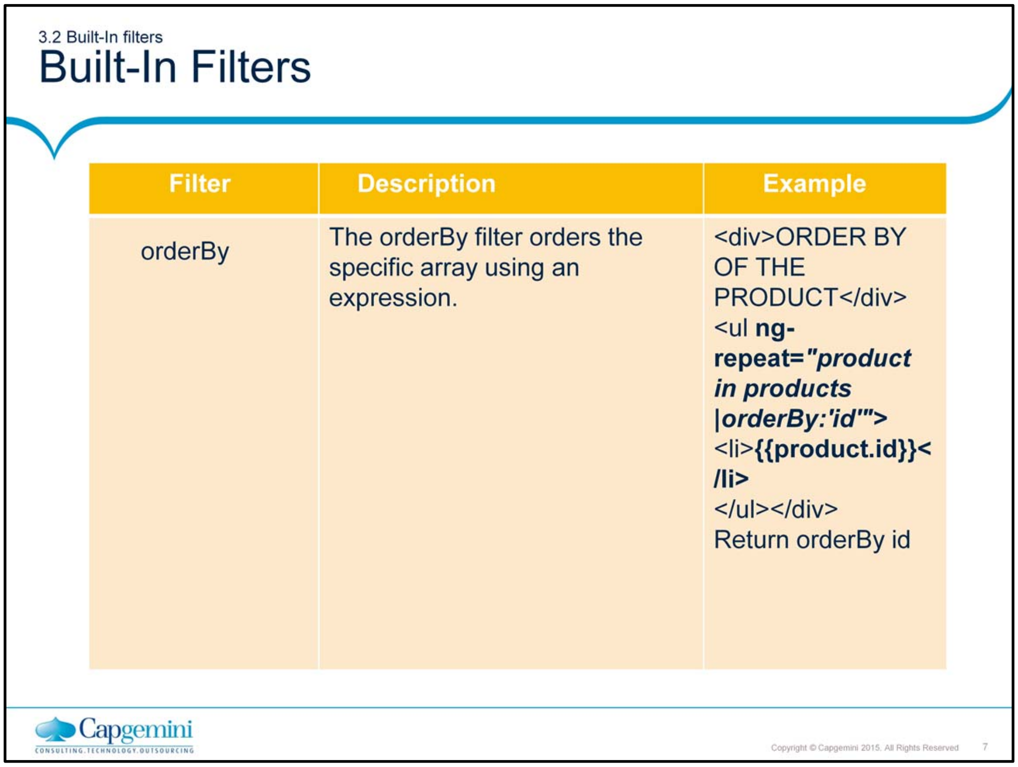
limitTo filter creates a new array or string that contains only the specified number of elements, either taken from the beginning or end, depending on whether the value is positive or negative.

```
{{ ‘CAPGEMINI' | limitTo:5 }}
<!-- returns  first 5 characters: CAPGE -->

{{ 'CAPGEMINI INDIA LEARNING' | limitTo:-8}}
<!-- returns  last 8 characters: LEARNING' -->

{{

['Bangalore','Chennai','Hyderabad','Gandhinagar','Mumabai','Noida','Pune']
        | limitTo:2
}}
<!--  returns  first 2 array elements :  ["Bangalore","Chennai"] -->
```



3.2 Built-In filters

Built-In Filters

Filter	Description	Example
orderBy	The orderBy filter orders the specific array using an expression.	<div>ORDER BY OF THE PRODUCT</div> <ul **ng-repeat=*"product in products |orderBy:'id'"*>** <li>**{{product.id}}</li>** </ul></div> Return orderBy id

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

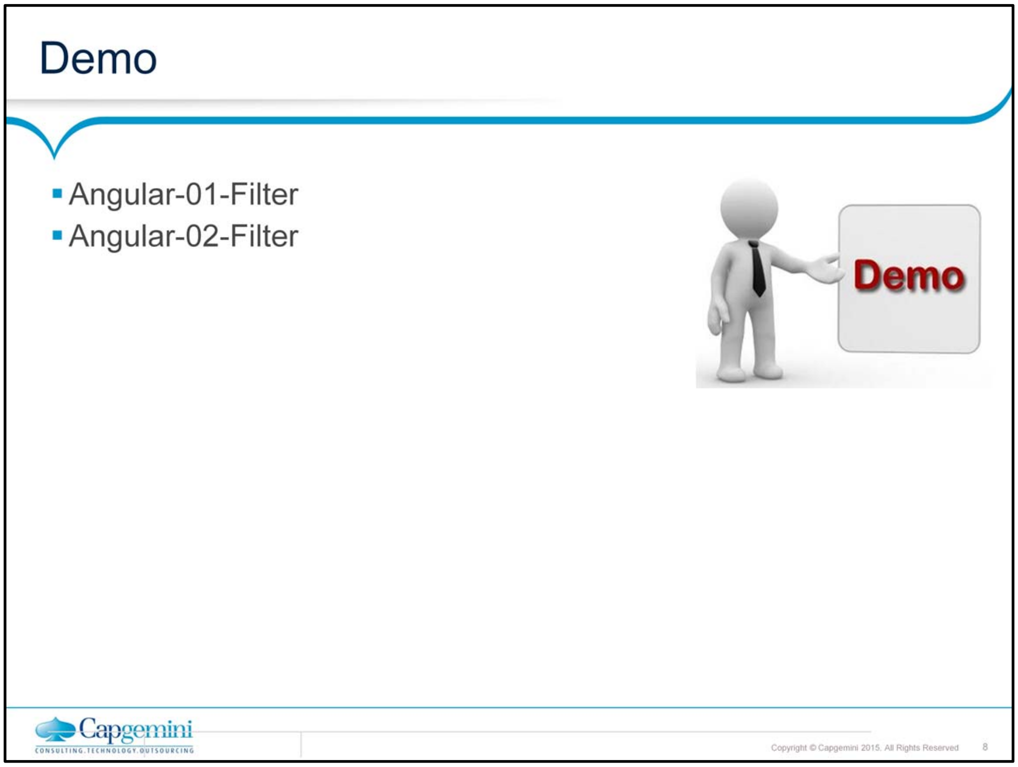
orderBy:

Orders alphabetically for strings and numerically for numbers.

It takes 2 parameters. First parameter is the predicate used to determine the order of the sorted array. Second parameter(optional) is a boolean value, if it is true it will sort the data in reverse order.

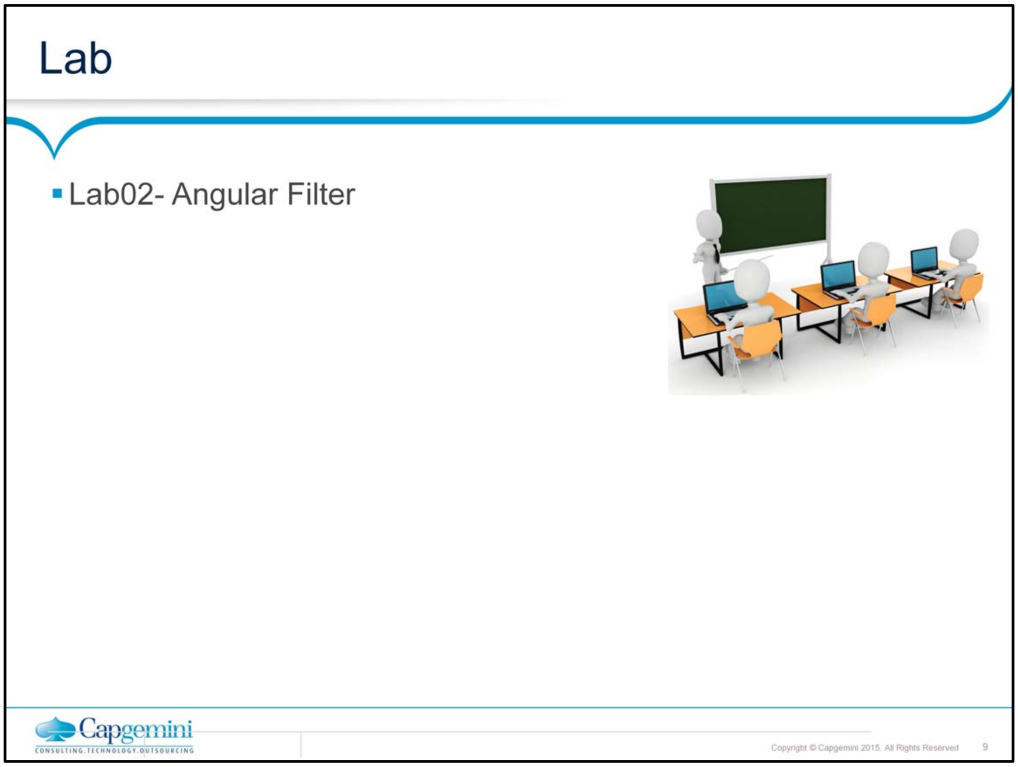
```
{{['Chennai','Bangalore','Pune','Mumbai'] | orderBy:'toString()'}}
<!-- returns ["Bangalore","Chennai","Mumbai","Pune"] -->


{{[{'Id':1,'Location':'Bangalore'},{'Id':2,'Location':'Chennai'}] |
orderBy:'Id':true}}
<!-- returns
[{"Id":2,"Location":"Chennai"},{"Id":1,"Location":"Bangalore"}] -->
```



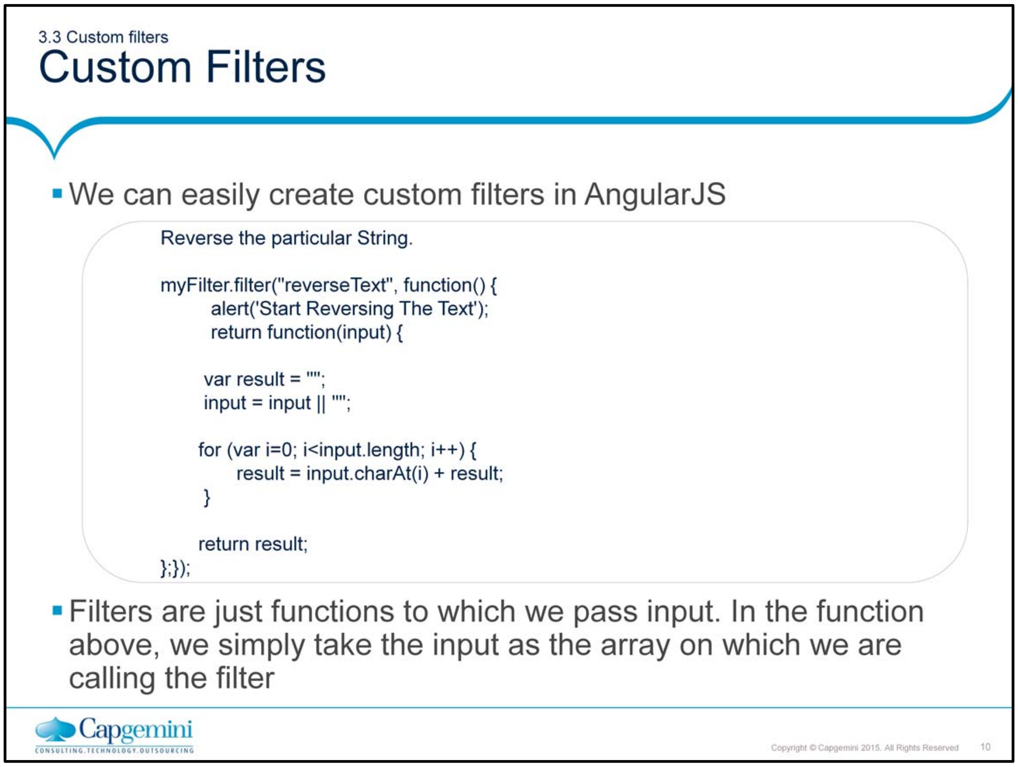
Demo

- Angular-01-Filter
- Angular-02-Filter



Lab

- Lab02- Angular Filter



3.3 Custom filters

Custom Filters

- We can easily create custom filters in AngularJS

```
Reverse the particular String.

myFilter.filter("reverseText", function() {
        alert('Start Reversing The Text');
        return function(input) {

       var result = "";
       input = input || "";

      for (var i=0; i<input.length; i++) {
            result = input.charAt(i) + result;
       }

      return result;
};});
```

- Filters are just functions to which we pass input. In the function above, we simply take the input as the array on which we are calling the filter

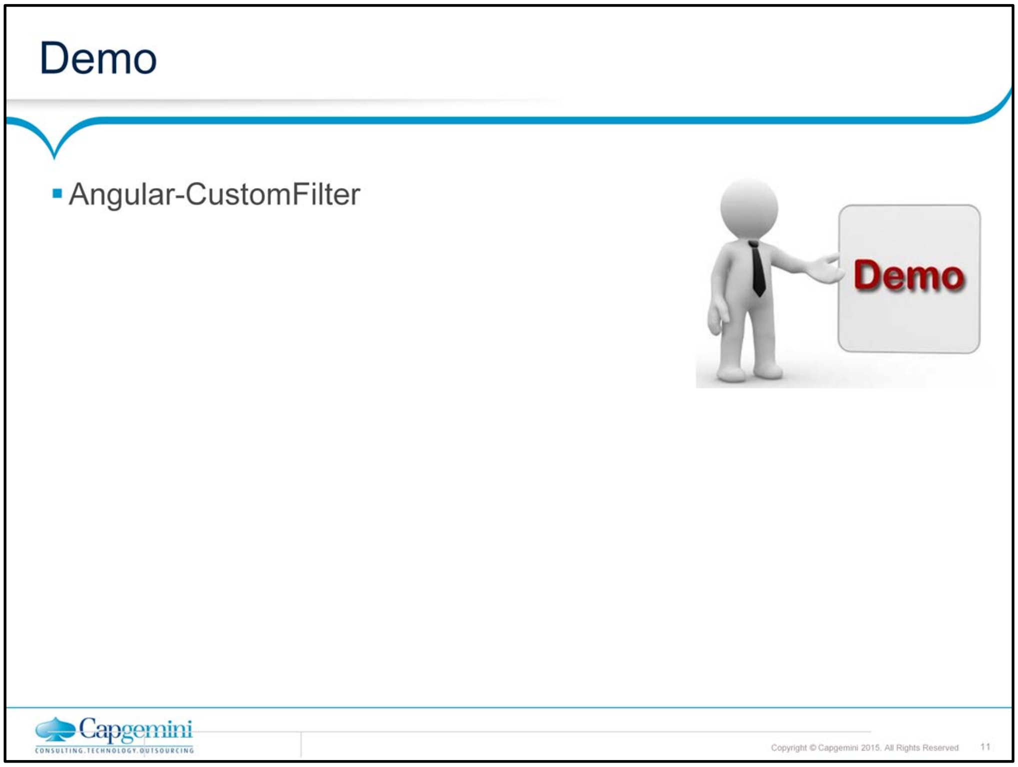
10

Just register a new filter factory function with your module. Internally, this uses the filterProvider 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.

```
<div ng-controller="MyController">
<input ng-model="greeting" type="text"><br>
No filter: {{greeting}}<br>
 Reverse: {{greeting|reverse}}<br>
 Reverse + uppercase: {{greeting|reverse:true}}<br>
Reverse, filtered in controller: {{filteredGreeting}}<br> </div>
```

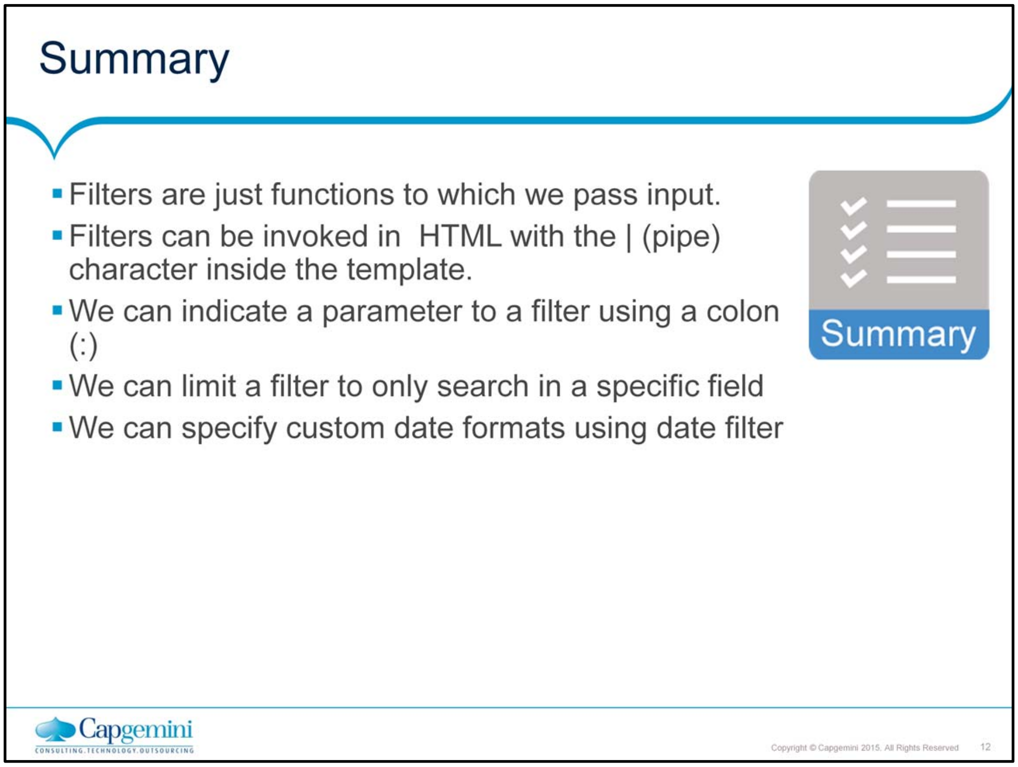
Script

```
angular.module('myReverseFilterApp', []) .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; }; })
.controller('MyController', ['$scope', 'reverseFilter', function($scope,
reverseFilter) { $scope.greeting = 'hello'; $scope.filteredGreeting =
reverseFilter($scope.greeting); }]);
```



Demo

- Angular-CustomFilter

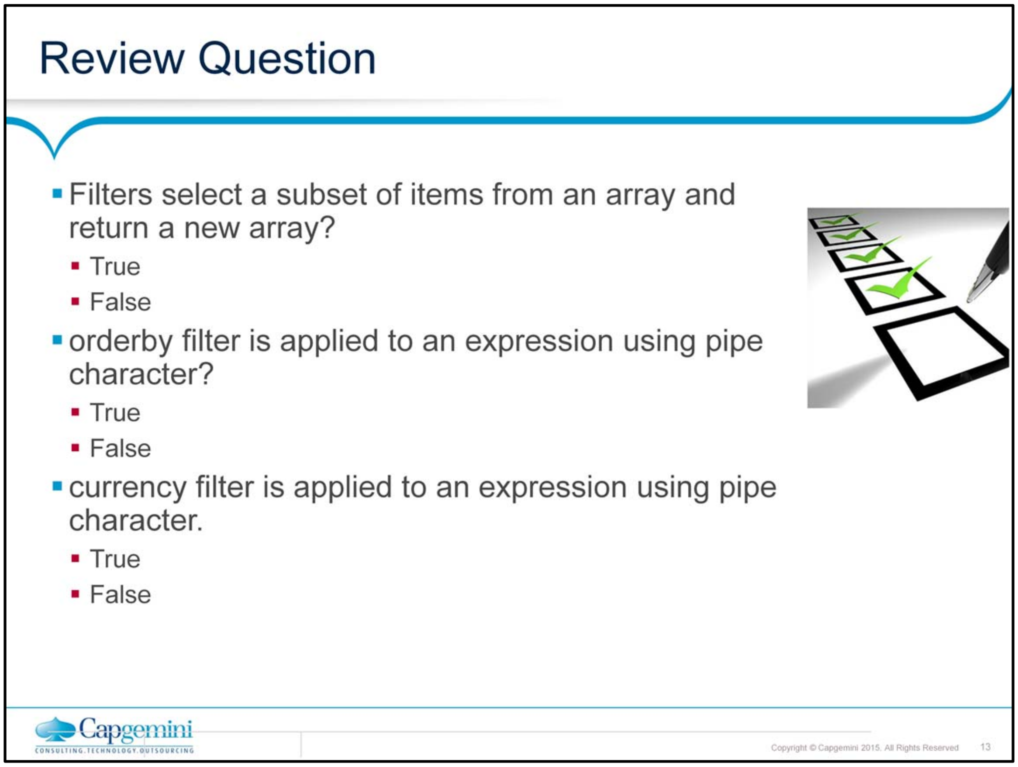


Summary

- Filters are just functions to which we pass input.
- Filters can be invoked in HTML with the | (pipe) character inside the template.
- We can indicate a parameter to a filter using a colon (:)
- We can limit a filter to only search in a specific field
- We can specify custom date formats using date filter

12

Add the notes here.



Review Question

- Filters select a subset of items from an array and return a new array?
  - True
  - False
- orderby filter is applied to an expression using pipe character?
  - True
  - False
- currency filter is applied to an expression using pipe character.
  - True
  - False