

JavaScript + Angular

Part 5

\$scope, filters

Author: Andrey Kucherenko

filter

```
{{ filter_expression | filter : expression : comparator : anyPropertyKey}}  
$filter('filter')(array, expression, comparator, anyPropertyKey)  
  
<tr ng-repeat="friend in friends | filter:searchText">  
  <td>{{friend.phone}}</td>  
</tr>
```

```
{{ currency_expression | currency : symbol : fractionSize}}

$filter('currency')(amount, symbol, fractionSize)

<script>
  angular.module('currencyExample', [])
    .controller('ExampleController', ['$scope', function($scope) {
      $scope.amount = 1234.56;
    }]);
</script>

<div ng-controller="ExampleController">
  <input type="number" ng-model="amount" aria-label="amount">
  <br>
  default currency symbol ($):
    <span id="currency-default">{{amount | currency}}</span><br>
  custom currency identifier (USD$):
    <span id="currency-custom">{{amount | currency:"USD$"}}</span>
  no fractions (0):
    <span id="currency-no-fractions">{{amount | currency:"USD$":0}}</span>
</div>
```



```
{{ number_expression | number : fractionSize}}

$filter('number')(number, fractionSize)

angular.module('numberFilterExample', [])
  .controller('ExampleController', ['$scope', function($scope) {
    $scope.val = 1234.56789;
  }]);

<div ng-controller="ExampleController">
  <label>Enter number:
    <input ng-model='val'></label><br>
  Default formatting:
    <span id='number-default'>{{val | number}}</span><br>
  No fractions: <span>{{val | number:0}}</span><br>
  Negative number: <span>{{-val | number:4}}</span>
</div>
```

Tasks

- Create array of products objects
- Each object should have price, name and count
- Show list of products
- Add filter for name
- Output price with currency
- Output formatted count
- Apply filter in JS and in HTML

```
{{ date_expression | date : format : timezone}}
```

```
$filter('date')(date, format, timezone)
```

```
<span ng-non-bindable>{{1288323623006 | date:'medium'}}</span>:
```

```
<span>{{1288323623006 | date:'medium'}}</span><br>
```

```
<span ng-non-bindable>{{1288323623006 | date:'yyyy-MM-dd HH:mm:ss Z'}}</span>:
```

```
<span>{{1288323623006 | date:'yyyy-MM-dd HH:mm:ss Z'}}</span><br>
```

```
<span ng-non-bindable>{{1288323623006 | date:'MM/dd/yyyy @ h:mma'}}</span>:
```

```
<span>{{'1288323623006' | date:'MM/dd/yyyy @ h:mma'}}</span><br>
```

```
<span ng-non-bindable>{{1288323623006 | date:"MM/dd/yyyy 'at' h:mma"}}</span>:
```

```
<span>{{'1288323623006' | date:"MM/dd/yyyy 'at' h:mma"}}</span><br>
```

```
{{ lowercase_expression | lowercase }}
```

```
$filter('lowercase')()
```

```
{{ uppercase_expression | uppercase }}
```

```
$filter('uppercase')()
```



```
{{ limitTo_expression | limitTo : limit : begin}}
```

```
$filter('limitTo')(input, limit, begin)
```

```
<script>
```

```
  angular.module('limitToExample', [])
```

```
    .controller('ExampleController', ['$scope', function($scope) {
```

```
      $scope.numbers = [1,2,3,4,5,6,7,8,9];
```

```
      $scope.letters = "abcdefghi";
```

```
      $scope.longNumber = 2345432342;
```

```
      $scope.numLimit = 3;
```

```
      $scope.letterLimit = 3;
```

```
      $scope.longNumberLimit = 3;
```

```
    }]);
```

```
</script>
```

```
<div ng-controller="ExampleController">
```

```
  <label>
```

```
    Limit {{numbers}} to:
```

```
    <input type="number" step="1" ng-model="numLimit">
```

```
</label>
```

```
<p>Output numbers: {{ numbers | limitTo:numLimit }}</p>
```

```
<label>
```

```
    Limit {{letters}} to:
```

```
    <input type="number" step="1" ng-model="letterLimit">
```

```
</label>
```

```
<p>Output letters: {{ letters | limitTo:letterLimit }}</p>
```

```
<label>
```

```
    Limit {{longNumber}} to:
```

```
    <input type="number" step="1" ng-model="longNumberLimit">
```

```
</label>
```

```
<p>Output long number: {{ longNumber | limitTo:longNumberLimit }}</p>
```

```
</div>
```


Tasks

- Create array of products objects
- Each object should have price, name, release date and quantity
- Show list of products
- Order list by price
- Show list of top3 products with high price
- Show release date
- Add possibility to order list by price, release date, quantity and name

```
angular.module('app', [])  
.filter('mfilter', function() {  
  return function(input, param1, param2) {  
    //... implementation of filter  
    return input + param1 + param2;  
  };  
})
```

```
{{hello|mfilter:'hello':'javascript'}}
```

```
$filter('mfilter')(input, param1, param2)
```

Tasks

- Create your own filter for convert string to smiles
- Create filter for show number as size in Kb, Mb and Gb
- Filter for convert camelCase to string separated by space

\$scope

Scopes

```
< Scope (1)
  < Scope (2)
    < Scope (3)
```

Models for (3)

```
{
  ctrl: { }
}
```

Enable Inspector

Watch Tree

```
Scope (1) | toggle
  Scope (2) | toggle
    Scope (3) | toggle
      • {{'Hola ' + framework + 'JS!' +
        ctrl.fName}}!
```

Watch Expressions

```
{{'Hola ' + framework + 'JS!' +
ctrl.fName}}! | 100% | 0.5250ms
```



```
<!DOCTYPE html>
<html ng-app="Lesson5App">
  <head>
    <meta charset="utf-8">
    <title>JS+Angular Classes: Lesson 5</title>
    <script src="lib/bundle.js" charset="utf-8"></script>
  </head>
  <body ng-controller="IndexController">
    <h1>JS+Angular Classes: Lesson 5</h1>
    <div class="HolaApp">
      <div ng-controller="MyController as ctrl">
        <h2>Hola App</h2>
        <p>{{ 'Hola ' + framework + 'JS!' + ctrl.fName }}!</p>
      </div>
    </div>
  </body>
</html>
```

\$rootScope

```
let $scope = $rootScope.$new();  
  
$rootScope.$on(name, listener);  
  
$rootScope.$emit(name, args);  
  
$rootScope.$broadcast(name, args);  
  
$scope.$on(name, listener);  
  
$scope.$emit(name, args);  
  
$scope.$broadcast(name, args);
```

```

<div ng-controller="EventController">
  Root scope <tt>MyEvent</tt> count: {{count}}
  <ul>
    <li ng-repeat="i in [1]" ng-controller="EventController">
      <button ng-click="$emit('MyEvent')">$emit('MyEvent')</button>
      <button ng-click="$broadcast('MyEvent')">$broadcast('MyEvent')</button>
      <br>
      Middle scope <tt>MyEvent</tt> count: {{count}}
      <ul>
        <li ng-repeat="item in [1, 2]" ng-controller="EventController">
          Leaf scope <tt>MyEvent</tt> count: {{count}}
        </li>
      </ul>
    </li>
  </ul>
</div>

```

```

angular.module('eventExample', [])
.controller('EventController', ['$scope', function($scope) {
  $scope.count = 0;
  $scope.$on('MyEvent', function() {
    $scope.count++;
  });
}]);

```

\$watch

```
$scope.$watch('name', function(newValue, oldValue) {  
    $scope.counter = scope.counter + 1;  
});  
  
$scope.names = ['igor', 'matias', 'misko', 'james'];  
$scope.dataCount = 4;  
  
$scope.$watchCollection('names', function(newNames, oldNames) {  
    $scope.dataCount = newNames.length;  
});
```

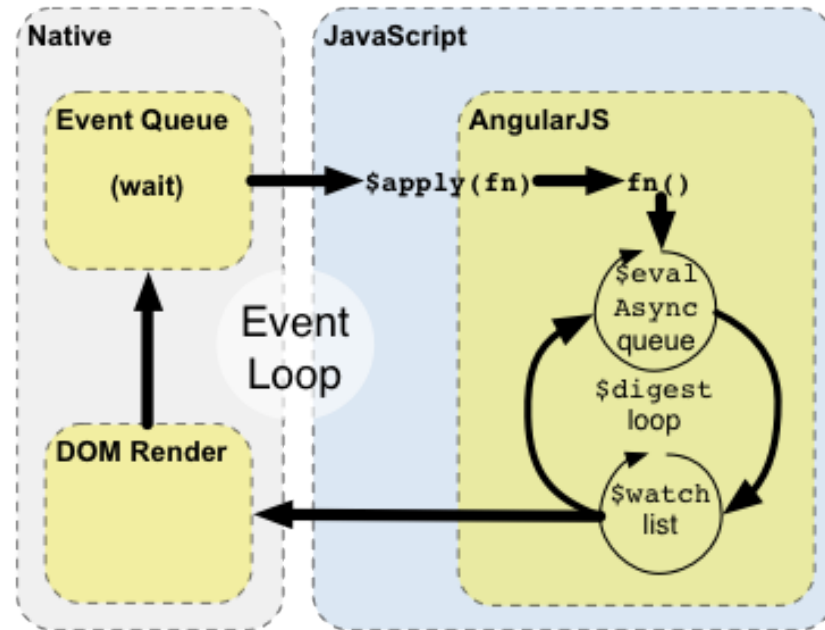
\$scope

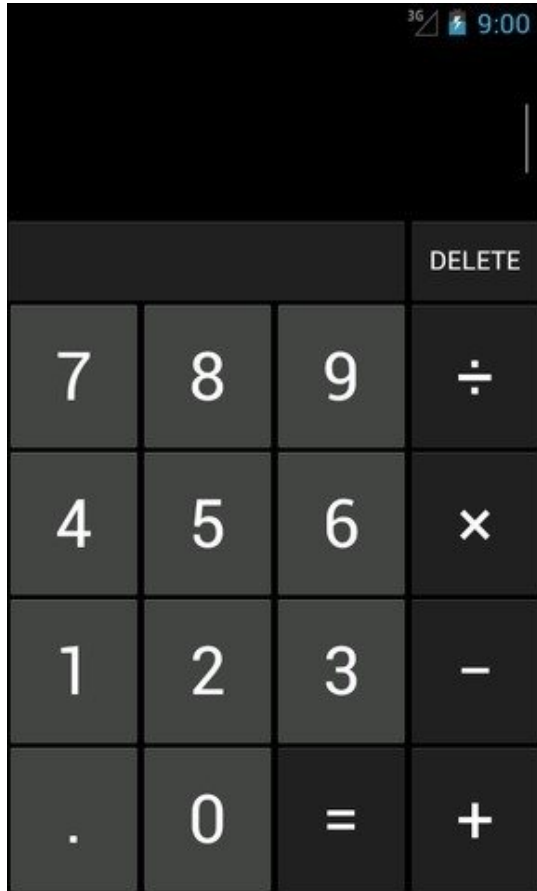
```
$scope.$digest();
```

```
$scope.$destroy();
```

```
$scope.$new(isolate, parent);
```

```
$scope.$apply(function() {})
```





Tasks

- Calculator

services

\$document

```
angular.module('documentExample', [])  
.controller('ExampleController', ['$scope', '$document', function($scope, $document) {  
  $scope.title = $document[0].title;  
  $scope.windowTitle = angular.element(window.document)[0].title;  
}]);
```

\$cacheFactory

```
let cache = $cacheFactory( 'cacheId' );  
$cacheFactory.get( 'cacheId' );  
$cacheFactory.get( 'noSuchCacheId' );  
  
cache.put( "key", "value" );  
cache.put( "another key", "another value" );
```

\$controller

```
$controller(constructor, locals);
```

```
let MyController = $controller('MyController', {  
    $http,  
    MyService  
});
```

\$interpolate

```
let $interpolate = ...; // injected
let exp = $interpolate('Hello {{name | uppercase}}!');
expect(exp({name: 'Angular'})).toEqual('Hello ANGULAR!');
```

\$interval & \$timeout

```
$interval(fn, delay, [count], [invokeApply], [Pass]);
```

```
$timeout([fn], [delay], [invokeApply], [Pass]);
```

```
let stopTime = $interval(updateTime, 1000);
```

```
$interval.cancel(stopTime);
```

\$templateCache

```
<script type="text/ng-template" id="templateId.html">  
  <p>This is the content of the template</p>  
</script>
```

```
$templateCache.put('templateId.html', 'This is the content of the template');
```


\$q

```
let promise = $q(function(resolve, reject) {
  setTimeout(function() {
    if (okToGreet(name)) {
      resolve('Hello, ' + name + '!');
    } else {
      reject('Greeting ' + name + ' is not allowed.');
```



```
    }
  }, 1000);
});

promise.then(function(greeting) {
  alert('Success: ' + greeting);
}, function(reason) {
  alert('Failed: ' + reason);
});
```

```
angular.service('myService', myServiceFunction);  
angular.factory('myFactory', myFactoryFunction);
```

```
myInjectedService // new myServiceFunction();  
myInjectedFactory // myFactoryFunction()
```



Pet Project