**Filters**

Filters are used to change modify the data and can be clubbed in expression or directives using pipe character. Following is the list of commonly used filters.

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Name** | **Description** |
| 1 | uppercase | converts a text to upper case text. |
| 2 | lowercase | converts a text to lower case text. |
| 3 | currency | formats text in a currency format. |
| 4 | filter | filter the array to a subset of it based on provided criteria. |
| 5 | orderby | orders the array based on provided criteria. |

## uppercase filter

Add uppercase filter to an expression using pipe character. Here we've added uppercase filter to print student name in all capital letters.

Enter first name:<input type = "text" ng-model = "student.firstName">

Enter last name: <input type = "text" ng-model = "student.lastName">

Name in Upper Case: {{student.fullName() | uppercase}}

## lowercase filter

Add lowercase filter to an expression using pipe character. Here we've added lowercase filter to print student name in all lowercase letters.

Enter first name:<input type = "text" ng-model = "student.firstName">

Enter last name: <input type = "text" ng-model = "student.lastName">

Name in Lower Case: {{student.fullName() | lowercase}}

## currency filter

Add currency filter to an expression returning number using pipe character. Here we've added currency filter to print fees using currency format.

Enter fees: <input type = "text" ng-model = "student.fees">

fees: {{student.fees | currency}}

## filter filter

To display only required subjects, we've used subjectName as filter.

Enter subject: <input type = "text" ng-model = "subjectName">

Subject:

<ul>

<li ng-repeat = "subject in student.subjects | filter: subjectName">

{{ subject.name + ', marks:' + subject.marks }}

</li>

</ul>

## orderby filter

To order subjects by marks, we've used orderBy marks.

Subject:

<ul>

<li ng-repeat = "subject in student.subjects | orderBy:'marks'">

{{ subject.name + ', marks:' + subject.marks }}

</li>

</ul>

## Example

Following example will showcase all the above mentioned filters.

*testAngularJS.htm*

<html>

<head>

<title>Angular JS Filters</title>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "mainApp" ng-controller = "studentController">

<table border = "0">

<tr>

<td>Enter first name:</td>

<td><input type = "text" ng-model = "student.firstName"></td>

</tr>

<tr>

<td>Enter last name: </td>

<td><input type = "text" ng-model = "student.lastName"></td>

</tr>

<tr>

<td>Enter fees: </td>

<td><input type = "text" ng-model = "student.fees"></td>

</tr>

<tr>

<td>Enter subject: </td>

<td><input type = "text" ng-model = "subjectName"></td>

</tr>

</table>

<br/>

<table border = "0">

<tr>

<td>Name in Upper Case: </td><td>{{student.fullName() | uppercase}}</td>

</tr>

<tr>

<td>Name in Lower Case: </td><td>{{student.fullName() | lowercase}}</td>

</tr>

<tr>

<td>fees: </td><td>{{student.fees | currency}}

</td>

</tr>

<tr>

<td>Subject:</td>

<td>

<ul>

<li ng-repeat = "subject in student.subjects | filter: subjectName |orderBy:'marks'">

{{ subject.name + ', marks:' + subject.marks }}

</li>

</ul>

</td>

</tr>

</table>

</div>

<script>

var mainApp = angular.module("mainApp", []);

mainApp.controller('studentController', function($scope) {

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fees:500,

subjects:[

{name:'Physics',marks:70},

{name:'Chemistry',marks:80},

{name:'Math',marks:65}

],

fullName: function() {

var studentObject;

studentObject = $scope.student;

return studentObject.firstName + " " + studentObject.lastName;

}

};

});

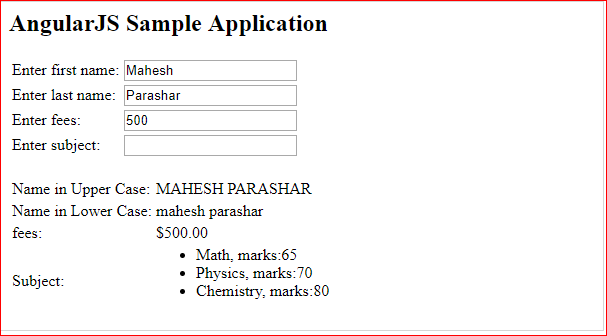
</script>

</body>

</html>

## Output

Open textAngularJS.htm in a web browser. See the result.



**Table**

Table data is normally repeatable by nature. ng-repeat directive can be used to draw table easily. Following example states the use of ng-repeat directive to draw a table.

<table>

<tr>

<th>Name</th>

<th>Marks</th>

</tr>

<tr ng-repeat = "subject in student.subjects">

<td>{{ subject.name }}</td>

<td>{{ subject.marks }}</td>

</tr>

</table>

Table can be styled using CSS Styling.

<style>

table, th , td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f2f2f2;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

</style>

## Example

Following example will showcase all the above mentioned directive.

*testAngularJS.htm*

<html>

<head>

<title>Angular JS Table</title>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

<style>

table, th , td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f2f2f2;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

</style>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "mainApp" ng-controller = "studentController">

<table border = "0">

<tr>

<td>Enter first name:</td>

<td><input type = "text" ng-model = "student.firstName"></td>

</tr>

<tr>

<td>Enter last name: </td>

<td>

<input type = "text" ng-model = "student.lastName">

</td>

</tr>

<tr>

<td>Name: </td>

<td>{{student.fullName()}}</td>

</tr>

<tr>

<td>Subject:</td>

<td>

<table>

<tr>

<th>Name</th>.

<th>Marks</th>

</tr>

<tr ng-repeat = "subject in student.subjects">

<td>{{ subject.name }}</td>

<td>{{ subject.marks }}</td>

</tr>

</table>

</td>

</tr>

</table>

</div>

<script>

var mainApp = angular.module("mainApp", []);

mainApp.controller('studentController', function($scope) {

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fees:500,

subjects:[

{name:'Physics',marks:70},

{name:'Chemistry',marks:80},

{name:'Math',marks:65},

{name:'English',marks:75},

{name:'Hindi',marks:67}

],

fullName: function() {

var studentObject;

studentObject = $scope.student;

return studentObject.firstName + " " + studentObject.lastName;

}

};

});

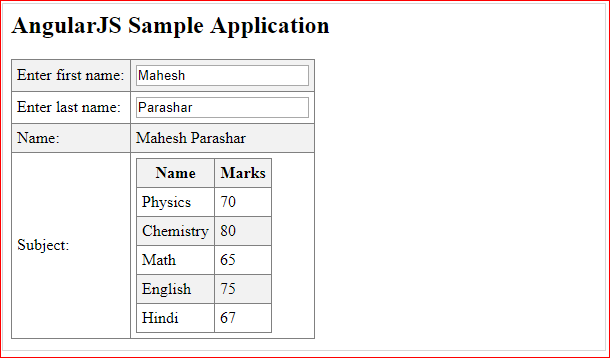
</script>

</body>

</html>

## Output

Open textAngularJS.htm in a web browser. See the result.



**HTML DOM**

Following directives can be used to bind application data to attributes of HTML DOM Elements.

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Name** | **Description** |
| 1 | ng-disabled | disables a given control. |
| 2 | ng-show | shows a given control. |
| 3 | ng-hide | hides a given control. |
| 4 | ng-click | represents a AngularJS click event. |

## ng-disabled directive

Add ng-disabled attribute to a HTML button and pass it a model. Bind the model to an checkbox and see the variation.

<input type = "checkbox" ng-model = "enableDisableButton">Disable Button

<button ng-disabled = "enableDisableButton">Click Me!</button>

## ng-show directive

Add ng-show attribute to a HTML button and pass it a model. Bind the model to an checkbox and see the variation.

<input type = "checkbox" ng-model = "showHide1">Show Button

<button ng-show = "showHide1">Click Me!</button>

## ng-hide directive

Add ng-hide attribute to a HTML button and pass it a model. Bind the model to an checkbox and see the variation.

<input type = "checkbox" ng-model = "showHide2">Hide Button

<button ng-hide = "showHide2">Click Me!</button>

## ng-click directive

Add ng-click attribute to a HTML button and update a model. Bind the model to html and see the variation.

<p>Total click: {{ clickCounter }}</p>

<button ng-click = "clickCounter = clickCounter + 1">Click Me!</button>

## Example

Following example will showcase all the above mentioned directives.

*testAngularJS.htm*

<html>

<head>

<title>AngularJS HTML DOM</title>

</head>

<body>

<h2>AngularJS Sample Application</h2>

<div ng-app = "">

<table border = "0">

<tr>

<td><input type = "checkbox" ng-model = "enableDisableButton">Disable Button</td>

<td><button ng-disabled = "enableDisableButton">Click Me!</button></td>

</tr>

<tr>

<td><input type = "checkbox" ng-model = "showHide1">Show Button</td>

<td><button ng-show = "showHide1">Click Me!</button></td>

</tr>

<tr>

<td><input type = "checkbox" ng-model = "showHide2">Hide Button</td>

<td><button ng-hide = "showHide2">Click Me!</button></td>

</tr>

<tr>

<td><p>Total click: {{ clickCounter }}</p></td>

<td><button ng-click = "clickCounter = clickCounter + 1">Click Me!</button></td>

</tr>

</table>

</div>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</body>

</html>

## Output

Open textAngularJS.htm in a web browser. See the result.

