**Directives**

AngularJS directives are used to extend HTML. These are special attributes starting with ng- prefix. We're going to discuss following directives −

* **ng-app** − This directive starts an AngularJS Application.
* **ng-init** − This directive initializes application data.
* **ng-model** − This directive binds the values of AngularJS application data to HTML input controls.
* **ng-repeat** − This directive repeats html elements for each item in a collection.

ng-app directive

ng-app directive starts an AngularJS Application. It defines the root element. It automatically initializes or bootstraps the application when web page containing AngularJS Application is loaded. It is also used to load various AngularJS modules in AngularJS Application. In following example, we've defined a default AngularJS application using ng-app attribute of a div element.

<div ng-app = "">

...

</div>

ng-init directive

ng-init directive initializes an AngularJS Application data. It is used to put values to the variables to be used in the application. In following example, we'll initialize an array of countries. We're using JSON syntax to define array of countries.

<div ng-app = "" ng-init = "countries = [{locale:'en-US',name:'United States'}, {locale:'en-GB',name:'United Kingdom'}, {locale:'en-FR',name:'France'}]">

...

</div>

ng-model directive

This directive binds the values of AngularJS application data to HTML input controls. In following example, we've defined a model named "name".

<div ng-app = "">

...

<p>Enter your Name: <input type = "text" ng-model = "name"></p>

</div>

ng-repeat directive

ng-repeat directive repeats html elements for each item in a collection. In following example, we've iterated over array of countries.

<div ng-app = "">

...

<p>List of Countries with locale:</p>

<ol>

<li ng-repeat = "country in countries">

{{ 'Country: ' + country.name + ', Locale: ' + country.locale }}

</li>

</ol>

</div>

Example

Following example will showcase all the above mentioned directives.

*testAngularJS.htm*

<html>

<head>

<title>AngularJS Directives</title>

</head>

<body>

<h1>Sample Application</h1>

<div ng-app = "" ng-init = "countries = [{locale:'en-US',name:'United States'}, {locale:'en-GB',name:'United Kingdom'}, {locale:'en-FR',name:'France'}]">

<p>Enter your Name: <input type = "text" ng-model = "name"></p>

<p>Hello <span ng-bind = "name"></span>!</p>

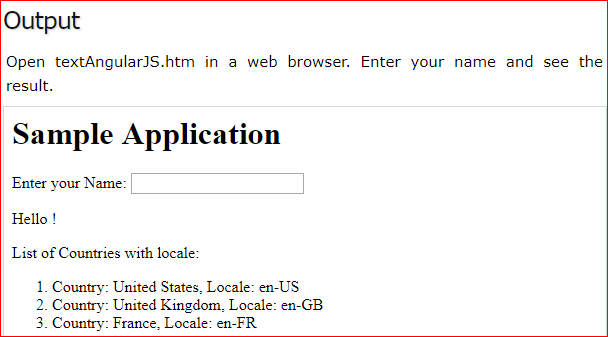
<p>List of Countries with locale:</p>

<ol>

<li ng-repeat = "country in countries">

{{ 'Country: ' + country.name + ', Locale: ' + country.locale }}

</li>



**Expression**

Expressions are used to bind application data to html. Expressions are written inside double braces like **{{expression}}**. Expressions behaves in same way as ng-bind directives. AngularJS application expressions are pure javascript expressions and outputs the data where they are used.

## Using numbers

<p>Expense on Books : {{cost \* quantity}} Rs</p>

## Using strings

<p>Hello {{student.firstname + " " + student.lastname}}!</p>

## Using object

<p>Roll No: {{student.rollno}}</p>

## Using array

<p>Marks(Math): {{marks[3]}}</p>

### **Example**

Following example will showcase all the above mentioned expressions.

*testAngularJS.htm*

<html>

<head>

<title>AngularJS Expressions</title>

</head>

<body>

<h1>Sample Application</h1>

<div ng-app = "" ng-init = "quantity = 1;cost = 30; student = {firstname:'Mahesh',lastname:'Parashar',rollno:101};marks = [80,90,75,73,60]">

<p>Hello {{student.firstname + " " + student.lastname}}!</p>

<p>Expense on Books : {{cost \* quantity}} Rs</p>

<p>Roll No: {{student.rollno}}</p>

<p>Marks(Math): {{marks[3]}}</p>

</div>

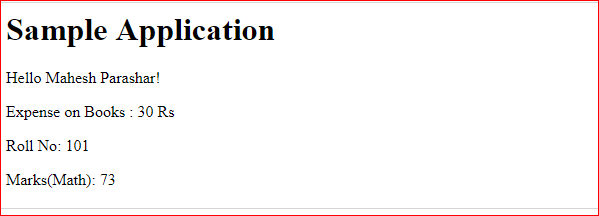
<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</body>

</html>

## Output

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



**Controller**

AngularJS application mainly relies on controllers to control the flow of data in the application. A controller is defined using **ng-controller** directive. A controller is a JavaScript object containing **attributes/properties** and **functions**. Each controller accepts **$scope** as a parameter which refers to the application/module that controller is to control.

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

...

</div>

Here we've declared a controller **studentController** using ng-controller directive. As a next step we'll define the studentController as follows −

<script>

function studentController($scope) {

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fullName: function() {

var studentObject;

studentObject = $scope.student;

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

}

};

}

</script>

* studentController defined as a JavaScript object with $scope as argument.
* $scope refers to application which is to use the studentController object.
* $scope.student is property of studentController object.
* firstName and lastName are two properties of $scope.student object. We've passed the default values to them.
* fullName is the function of $scope.student object whose task is to return the combined name.
* In fullName function we're getting the student object and then return the combined name.
* As a note, we can also define the controller object in separate JS file and refer that file in the html page.

Now we can use studentController's student property using ng-model or using expressions as follows.

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

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

<br>

You are entering: {{student.fullName()}}

* We've bounded student.firstName and student.lastname to two input boxes.
* We've bounded student.fullName() to HTML.
* Now whenever you type anything in first name and last name input boxes, you can see the full name getting updated automatically.

## Example

Following example will showcase use of controller.

*testAngularJS.htm*

<html>

<head>

<title>Angular JS Controller</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">

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

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

<br>

You are entering: {{student.fullName()}}

</div>

<script>

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

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

$scope.student = {

firstName: "Mahesh",

lastName: "Parashar",

fullName: function() {

var studentObject;

studentObject = $scope.student;

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

}

};

});

</script>

</body>

</html>

Another Example

<div ng-app="myApp" ng-controller="myCtrl">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{firstName + " " + lastName}}

</div>

<script>

var app = angular.module('myApp', []);

app.controller('myCtrl', function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

});

</script>

## Output

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



**Fname=”Pankaj”;//Object**

**Var a; //Variable**