AngularJS Controllers

[« Previous](http://www.w3schools.com/angular/angular_model.asp)

[Next Chapter »](http://www.w3schools.com/angular/angular_scopes.asp)

AngularJS controllers **control the data** of AngularJS applications.

AngularJS controllers are regular **JavaScript Objects**.

AngularJS Controllers

AngularJS applications are controlled by controllers.

The **ng-controller** directive defines the application controller.

A controller is a **JavaScript Object**, created by a standard JavaScript **object constructor**.

AngularJS 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>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_controller)

Application explained:

The AngularJS application is defined by  **ng-app="myApp"**. The application runs inside the <div>.

The **ng-controller="myCtrl"** attribute is an AngularJS directive. It defines a controller.

The **myCtrl** function is a JavaScript function.

AngularJS will invoke the controller with a **$scope** object.

In AngularJS, $scope is the application object (the owner of application variables and functions).

The controller creates two properties (variables) in the scope (**firstName** and **lastName**).

The **ng-model** directives bind the input fields to the controller properties (firstName and lastName).

Controller Methods

The example above demonstrated a controller object with two properties: lastName and firstName.

A controller can also have methods (variables as functions):

AngularJS Example

<div ng-app="myApp" ng-controller="personCtrl">  
  
First Name: <input type="text" ng-model="firstName"><br>  
Last Name: <input type="text" ng-model="lastName"><br>  
<br>  
Full Name: {{fullName()}}  
  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('personCtrl', function($scope) {  
    $scope.firstName = "John";  
    $scope.lastName = "Doe";  
    $scope.fullName = function() {  
        return $scope.firstName + " " + $scope.lastName;  
    };  
});  
</script>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_controller_property)

Controllers In External Files

In larger applications, it is common to store controllers in external files.

Just copy the code between the <script> tags into an external file named [personController.js](http://www.w3schools.com/angular/personController.js):

AngularJS Example

<div ng-app="myApp" ng-controller="personCtrl">  
  
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 src="personController.js"></script>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_controller_js)

Another Example

For the next example we will create a new controller file:

angular.module('myApp', []).controller('namesCtrl', function($scope) {  
    $scope.names = [  
        {name:'Jani',country:'Norway'},  
        {name:'Hege',country:'Sweden'},  
        {name:'Kai',country:'Denmark'}  
    ];  
});

Save the file as  [namesController.js](http://www.w3schools.com/angular/namesController.js):

And then use the controller file in an application:

AngularJS Example

<div ng-app="myApp" ng-controller="namesCtrl">  
  
<ul>  
  <li ng-repeat="x in names">  
    {{ x.name + ', ' + x.country }}  
  </li>  
</ul>  
  
</div>  
  
<script src="namesController.js"></script>

[Try it Yourself »](http://www.w3schools.com/angular/tryit.asp?filename=try_ng_controller_names)