How does it work?

1. In **app.js**, we created a new *module* named myApp. A *module* contains the different components of an AngularJS app.
2. Then, in **index.html** we added <body ng-app="myApp">. The ng-app is called a *directive*. It tells AngularJS that the myApp module will live within the <body> element, termed the application's *scope*. In other words, we used the ng-app directive to define the application scope.
3. In **MainController.js** we created a new *controller* named MainController. A *controller* manages the app's data. Here we use the property title to store a string, and attach it to $scope.
4. Then, in **index.html**, we added <div class="main" ng-controller="MainController">. Like ng-app, ng-controller is a *directive* that defines the controller scope. This means that properties attached to $scope in MainController become available to use within <div class="main">.
5. Inside <div class="main"> we accessed $scope.titleusing {{ title }}. This is called an *expression*. Expressions are used to display values on the page.
6. The value of title showed up when we viewed the app in the browser.

Let's do a quick review:

* A **module** contains the different components of an AngularJS app
* A **controller** manages the app's data
* An **expression** displays values on the page
* A **filter** formats the value of an expression

We've used a few directives so far - ng-app, ng-controller, ng-repeat, and ng-src. What can we generalize about directives?

Directives bind behavior to HTML elements. When the app runs, AngularJS walks through each HTML element looking for directives. When it finds one, AngularJS triggers that behavior (like attaching a scope or looping through an array).