



# Learn Angular

BY MAKING A QUIZ APP  
FROM SCRATCH

[HUNGRYTURTLECODE.COM](http://HUNGRYTURTLECODE.COM)

## Part 6



# Check Out The Whole Course Index

1. [Getting Started](#)
2. [Ng-controller directive and the \(mis\)use of \\$scope](#)
3. [Looping around with the ng-repeat directive](#)
4. [Markup for the Bootstrap Modal](#)
5. [Using Angular Filters to create real time search](#)
6. You are here
7. [Services in Angular Make everything easier](#)
8. [What is this infamous dependency injection in Angular?](#)
9. [Let's Build A Factory](#)
10. [The ng-class directive](#)
11. [More Bootstrap Markup - The Well](#)
12. [Adding some logic to the controller](#)
13. [Making things disappear with ng-if](#)
14. [The \\$index property for ng-repeat](#)
15. [Reusing code is always a good idea](#)
16. [Using Bootstrap to help with styling error messages](#)
17. [The final prompt after the quiz](#)
18. [Marking the quiz](#)
19. [More dependency injection](#)
20. [Reusing and slightly modifying some previous Bootstrap](#)
21. [More than one way to use ng-class](#)
22. [Another Angular Filter](#)
23. [More usage of Ng-if](#)
24. [Finishing The App](#)

## Ng-Click Directive And Ng-Show / Ng-Hide

We're nearly in a position to move away from our list controller and start building out the quiz controller. But before we can do that we need to create the "start quiz" button, which is what we will tackle first. We will then use the ng-click directive to control functionality when the button is clicked.

If you want to see the app for yourself, [check it out here](#).

The git repo [can be found here](#).

### Video Or Text Tutorial?

Take your pick. You can watch the video below or you can read through this article to get the same information.



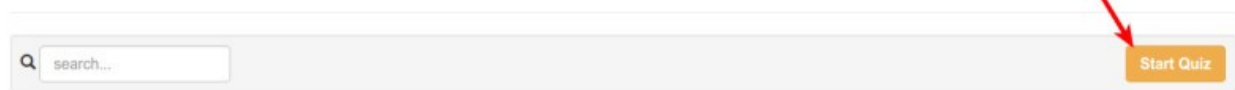
Click the image to go to the video on youtube or go to [https://www.youtube.com/watch?v=ZKbPtYYbSOg&index=6&list=PLqr0oBkln1FBmOjK24\\_B4y\\_VAA8736wPq](https://www.youtube.com/watch?v=ZKbPtYYbSOg&index=6&list=PLqr0oBkln1FBmOjK24_B4y_VAA8736wPq)

[The next part can be found here](#)

As you can see from the [example application](#) in the video the button is on the right hand side of the grey area where the search bar is. This is part of the same form in the HTML. So we will start by creating a button with some bootstrap classes to pull it to the right.

## Turtle Facts Quiz

Learn about all the turtles below before you decide to take on the **TURTLE QUIZ**



```
<form class="form-inline well well-sm clearfix">
  <span class="glyphicon glyphicon-search"></span>
  <input
    type="text"
    placeholder="Search..."
    class="form-control"
    ng-model="list.search">

  <button class="btn btn-warning pull-right">
    <strong>Start Quiz</strong>
  </button>
</form>
```

## Showing And Hiding HTML elements with Angular

Now that we actually have the button we needed to do something. When we click the button we want our turtle list markup to disappear and be replaced with the quiz. To do this we will utilise a partnership between the ng-click directive and [ng-hide](#).

Both of these are new directives but they're also pretty self explanatory. We will first focus on the ng-show and ng-hide directives.

Ng-hide will accept a [boolean](#) or an expression that evaluates to

a boolean. If this boolean is true then that element will hide and if it's false the element will show. [Ng-show](#) is another similar directive that is the exact opposite of ng-hide. Either can be used but they tend to both be useful to keep things more semantically explicit.

The boolean that we will use to hide the list controller markup, as well as showing the quiz controller markup (using ng-hide and ng-show respectively) will be a property on our controllers which we will call quizActive.

We will add the following code to our list controller.

```
function ListController(){
  var vm = this;

  vm.data = turtlesData;
  vm.activeTurtle = {};
  vm.search = "";

  //Set to false initially so the list shows when the app loads
  vm.quizActive = false;

  vm.changeActiveTurtle = changeActiveTurtle;

  function changeActiveTurtle(index){
    vm.activeTurtle = index;
  }
}
```

Property is initially set to false as when the application loads the quiz is not active and we want the list controller to display. For this reason we can use ng-hide on our list controller div and it will display when the application first loads just like we want (because quizActive is false, so it does not hide the div).

We can add the ng-hide directive to our list controller markup like so.

```
<div ng-controller="listCtrl as list" ng-hide="list.quizActive">
```



## But Nothing Happens When I Click The Button!

We can now show and hide the list controller markup (or any html elements for that matter) using Angular. But now we want to programmatically hide the list controller when we click the start quiz button.

This is achieved using the ng-click directive to trigger a function on our controller that will change the quizActive property. Inside the quotes of the ng-click directive we will add the name of a function or literal expression (in our case a function name) that will run when that HTML element is clicked.

```
<button class="btn btn-warning pull-right"
  ng-click="list.activateQuiz()">
  <strong>Start Quiz</strong>
</button>
```

Now we need to create the activateQuiz function. We will do this in the same way that we have created a function in the past – by initialising and named function inside or controller.

The activateQuiz function will be extremely simple as all we need to do is set the quizActive property to true.

```
vm.activateQuiz = activateQuiz;
function activateQuiz(){
  vm.quizActive = true;
}
```

## Moving On To Part 7

In [part 7](#) of this series we will start diving deeper into Angular. We will be thinking about the problems that arise when we are trying to share information across controllers (the quizActive state between the list controller and the quiz controller).

Of course, we will solve these problems and introduce some new Angular features in the process – Services. We will also be kicking things off with our quiz controller. So lot's to look forward to.

See you over there in [part 7](#).

Adrian