



Learn Angular

BY MAKING A QUIZ APP
FROM SCRATCH

HUNGRYTURTLECODE.COM

Part 20



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. [The powerful ng-click directive](#)
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 styling an error message](#)
17. [The final prompt after the quiz](#)
18. [Marking the quiz](#)
19. [More dependency injection](#)
20. You are here
21. [More than one way to use ng-class](#)
22. [Another Angular Filter](#)
23. [More usage of Ng-if](#)
24. [Finishing The App](#)

Some Really Familiar Bootstrap Markup

This lesson isn't introducing anything new. Infact, you will have seen much of this code before. What we are tackling is the area at the top of the results controller that shows what questions the user got right and wrong along with a legend. This is very much like the progress area in the quiz controller. Let's take a look at some familiar bootstrap code.

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

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

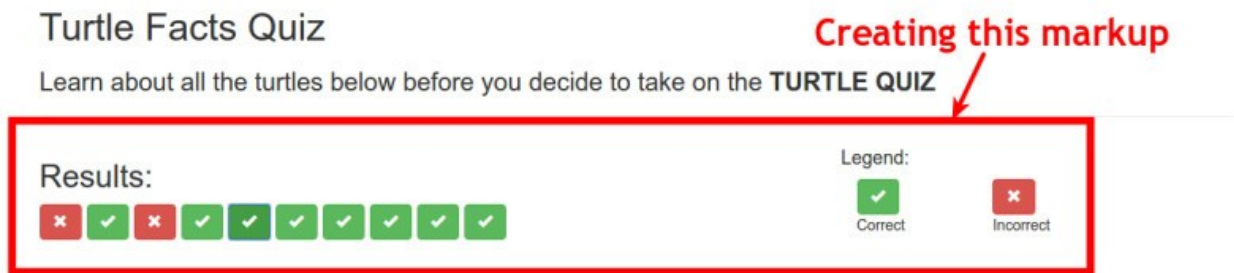


Click the image to go to the video on youtube or go here https://www.youtube.com/watch?v=AxQcLIROxYw&index=20&list=PLqr0oBkln1FBmOjK24_B4y_VAA8736wPq

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

I say that it is similar, the markup is almost identical with a few key differences. Mainly the colors used are no longer red and blue but are instead red and green to indicate correct or incorrect. The glyphicons used are also different - a tick and an

x this time.



Other than those differences the code is identical. So I will not bore you by explaining every line of the markup. Instead, I will just show you what we have.

```
<div class="row">
  <div class="col-xs-8">
    <h2>Results:</h2>
    <div class="btn-toolbar">
      <button class="btn"
        ng-repeat="question in results.dataService.quizQuestions"
        ng-class="{ 'btn-success': question.correct, 'btn-danger': !question.correct}"
        ng-click="results.setActiveQuestion($index)">
        <span class="glyphicon"
          ng-class="{ 'glyphicon-ok': question.correct, 'glyphicon-remove': !question.correct}"></span>
      </button>
    </div>
  </div>
  <div class="col-xs-4">
    <div class="row">
      <h4>Legend:</h4>
      <div class="col-sm-4">
        <button class="btn btn-success">
          <span class="glyphicon glyphicon-ok"></span>
        </button>
        <p>Correct</p>
      </div>
      <div class="col-sm-4">
        <button class="btn btn-danger">
          <span class="glyphicon glyphicon-remove"></span>
        </button>
        <p>Incorrect</p>
      </div>
    </div>
  </div>
</div>
```


Note the key differences. We are now using the correct property we set to true or false on every question when we marked the quiz to style the buttons accordingly using ng-class on the button and the glyphsicons.

We are referencing question.correct here has question is the alias for the ng-repeat. So each iteration through ng-repeat will be a different question in the quiz.

Also note that we do not have to say something like question.correct == true (although, this would also be valid and provide the output we need) because question.correct is already a boolean value so ng-class can evaluate it straight away. There is no need to add in extra computation when it is not needed.

The Definition Of Short And Sweet

This lesson really was short. I don't want to get into the [next part](#) in this article as it will then be way too long. So you will just have to click through to the [next part](#) to get all that wonderful goodness there.

See you in [part 21](#)

Adrian