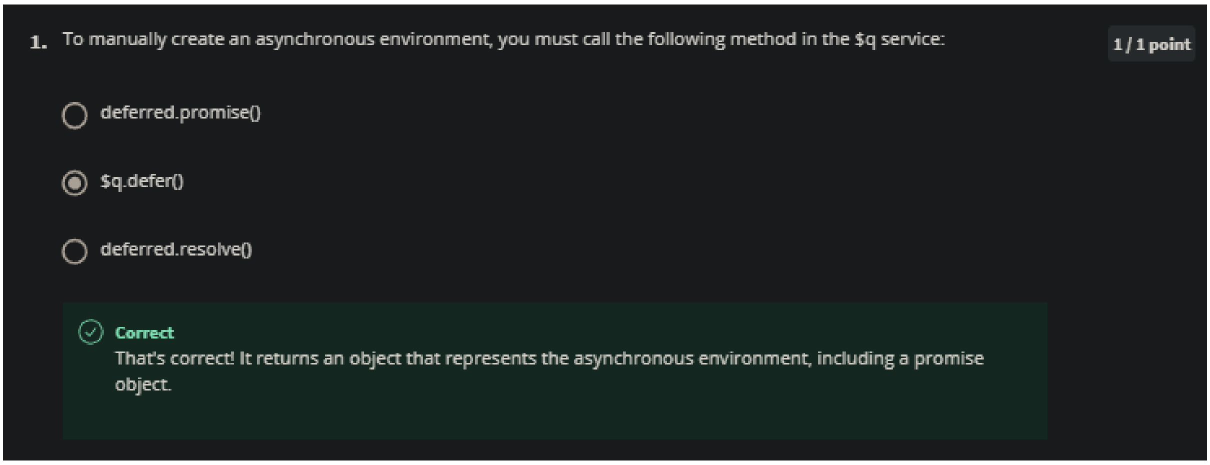
Week 3

Quiz 22



1. To manually create an asynchronous environment, you must call the following method in the $q service:

1 / 1 point

○ deferred.promise()

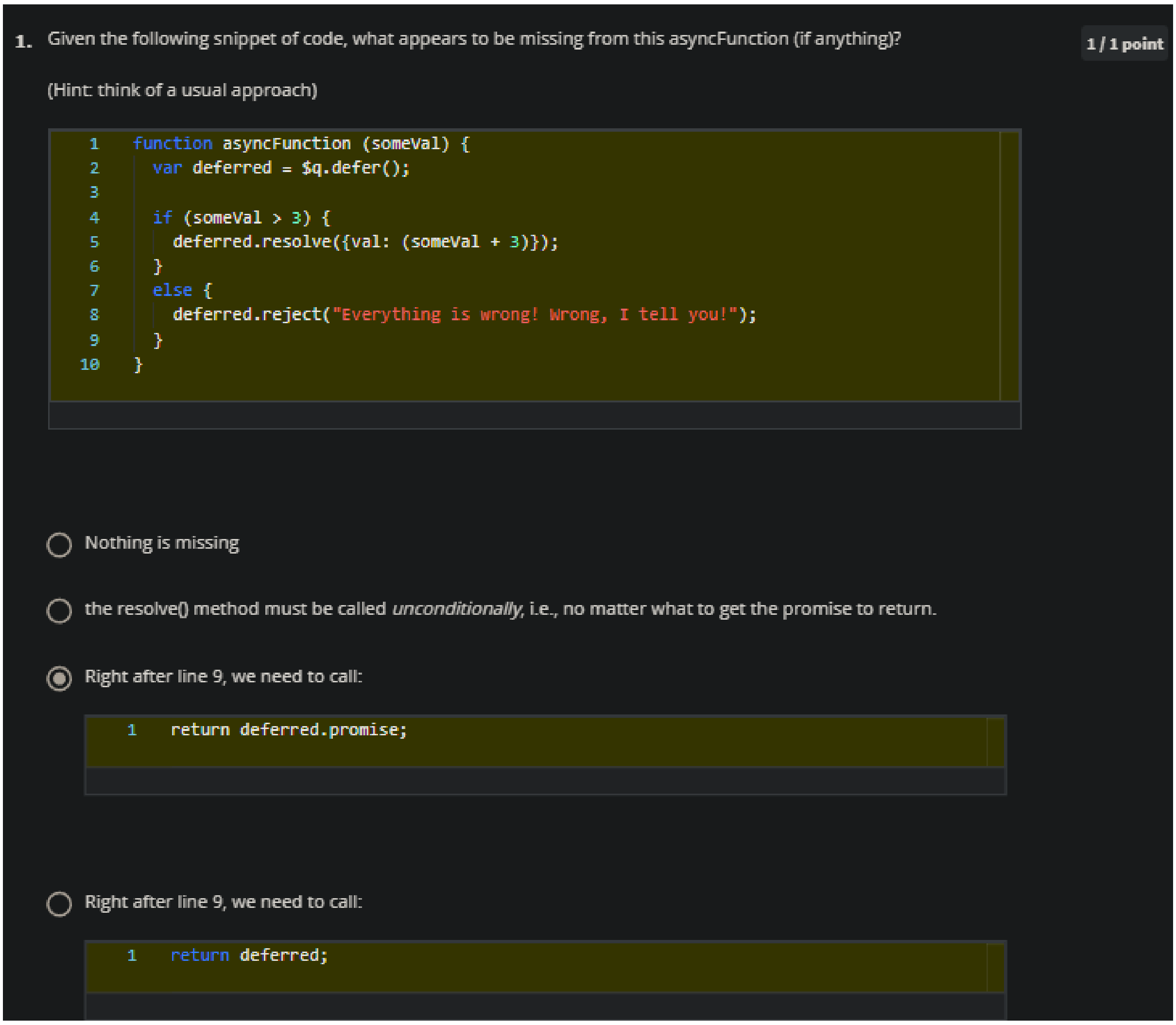
● $q.defer()

○ deferred.resolve()

✓ Correct

That's correct! It returns an object that represents the asynchronous environment, including a promise object.

Quiz 23



1. Given the following snippet of code, what appears to be missing from this asyncFunction (if anything)?

1 / 1 point

(Hint: think of a usual approach)

```
 1  function asyncFunction (someVal) {
 2    var deferred = $q.defer();
 3
 4    if (someVal > 3) {
 5      deferred.resolve({val: (someVal + 3)});
 6    }
 7    else {
 8      deferred.reject("Everything is wrong! Wrong, I tell you!");
 9    }
10  }
```

○ Nothing is missing

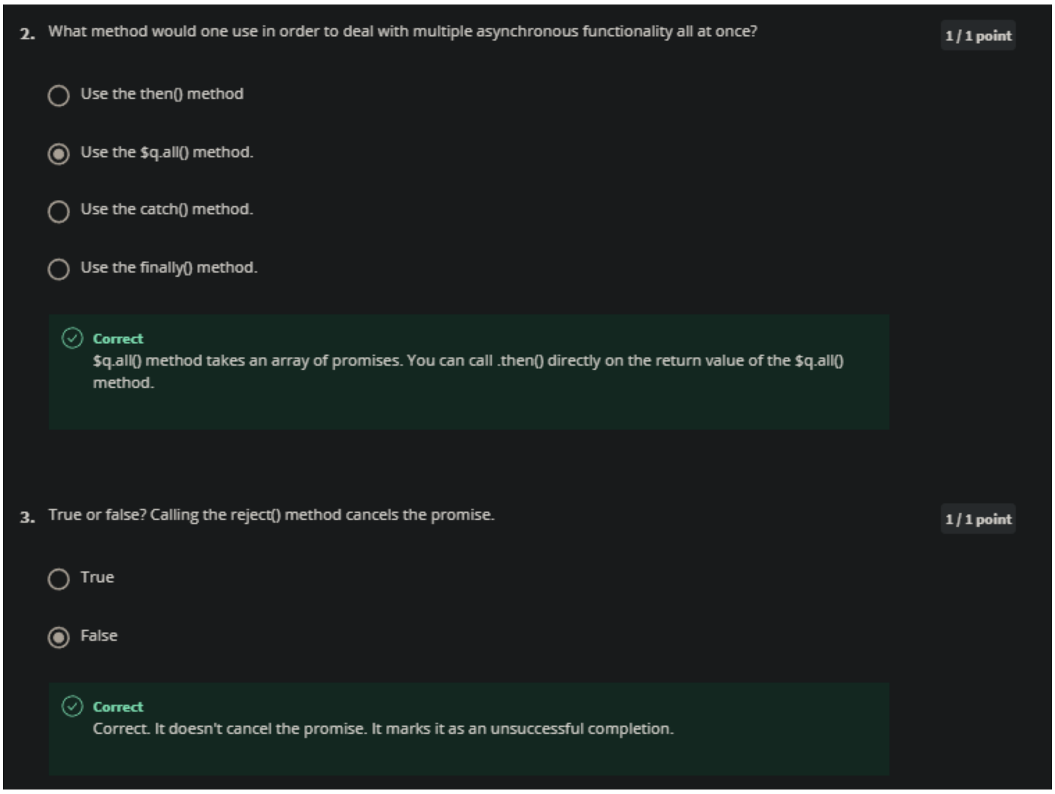
○ the resolve() method must be called *unconditionally*, i.e., no matter what to get the promise to return.

● Right after line 9, we need to call:

```
1  return deferred.promise;
```

○ Right after line 9, we need to call:

```
1  return deferred;
```



2. What method would one use in order to deal with multiple asynchronous functionality all at once?

1 / 1 point

○ Use the then() method

● Use the $q.all() method.

○ Use the catch() method.

○ Use the finally() method.

✓ Correct
$q.all() method takes an array of promises. You can call .then() directly on the return value of the $q.all() method.

3. True or false? Calling the reject() method cancels the promise.

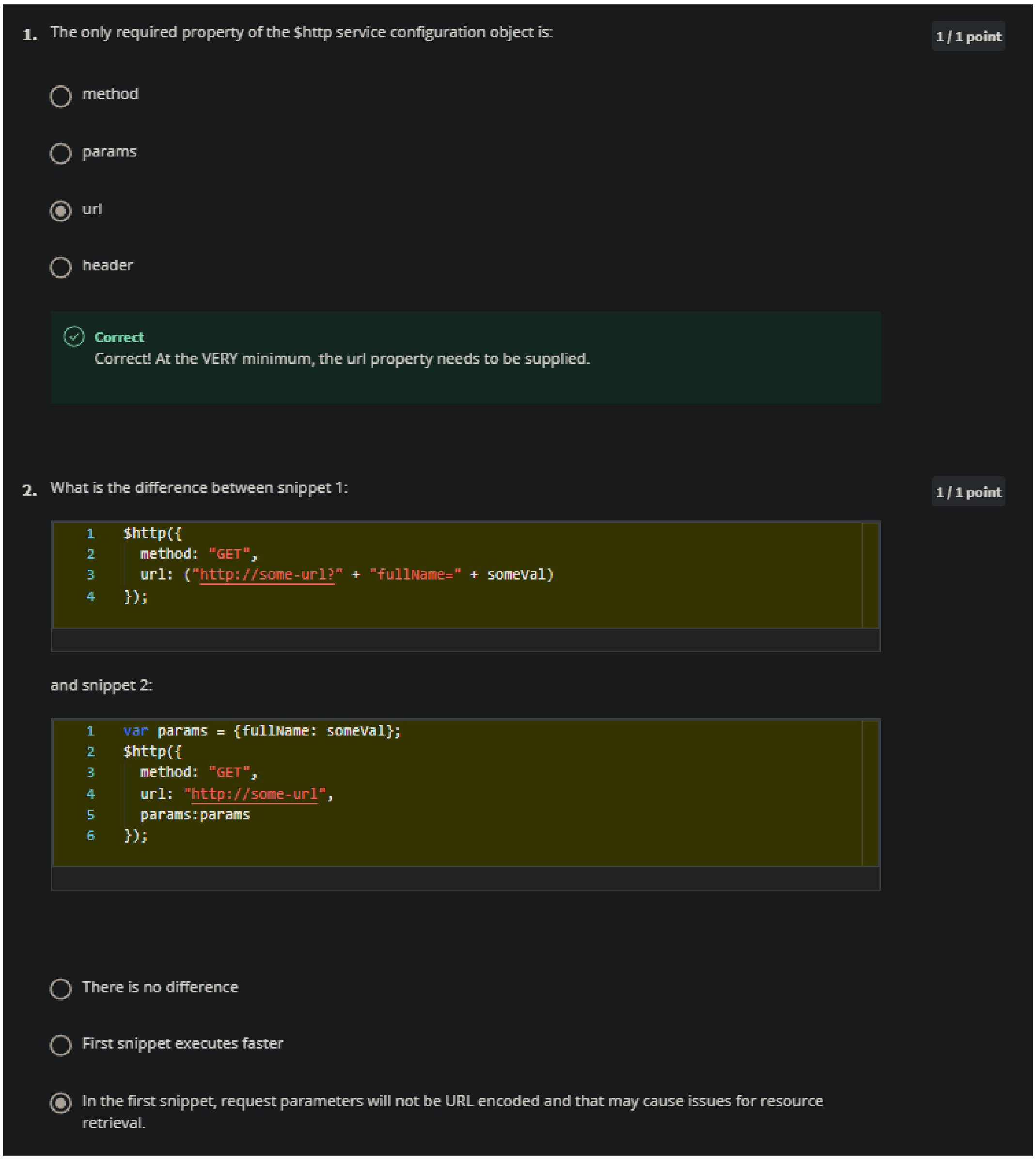
1 / 1 point

○ True

● False

✓ Correct
Correct. It doesn't cancel the promise. It marks it as an unsuccessful completion.

Quiz 24



1. The only required property of the $http service configuration object is:

1 / 1 point

○ method

○ params

● url

○ header

✓ Correct
Correct! At the VERY minimum, the url property needs to be supplied.

2. What is the difference between snippet 1:

1 / 1 point

```
$http({
  method: "GET",
  url: ("http://some-url?" + "fullName=" + someVal)
});
```

and snippet 2:

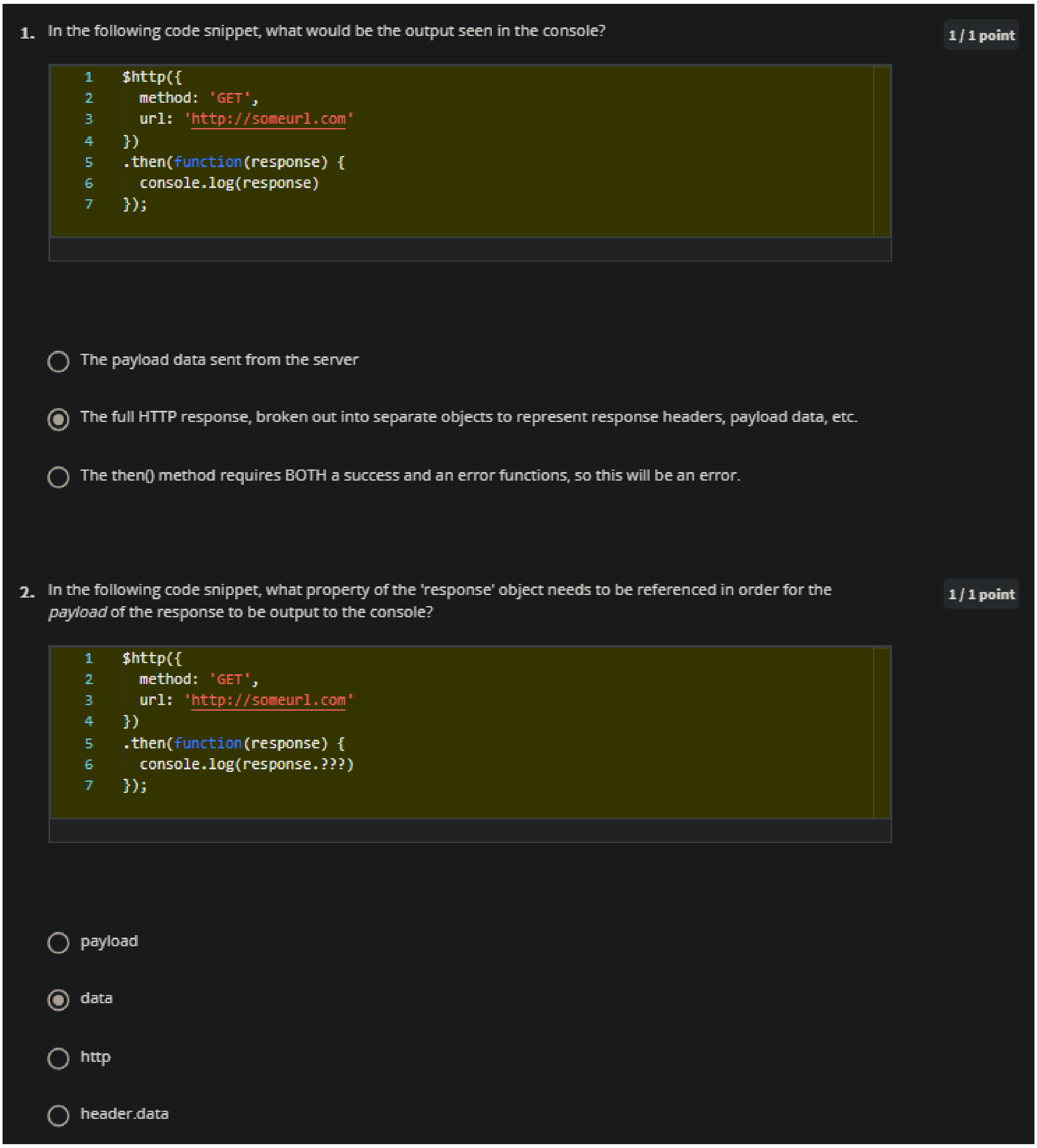
```
var params = {fullName: someVal};
$http({
  method: "GET",
  url: "http://some-url",
  params:params
});
```

○ There is no difference

○ First snippet executes faster

● In the first snippet, request parameters will not be URL encoded and that may cause issues for resource retrieval.

Quiz 25



1. In the following code snippet, what would be the output seen in the console?

1 / 1 point

```
$http({
  method: 'GET',
  url: 'http://someurl.com'
})
.then(function(response) {
  console.log(response)
});
```

○ The payload data sent from the server

● The full HTTP response, broken out into separate objects to represent response headers, payload data, etc.

○ The then() method requires BOTH a success and an error functions, so this will be an error.

2. In the following code snippet, what property of the 'response' object needs to be referenced in order for the *payload* of the response to be output to the console?

1 / 1 point

```
$http({
  method: 'GET',
  url: 'http://someurl.com'
})
.then(function(response) {
  console.log(response.???)
});
```

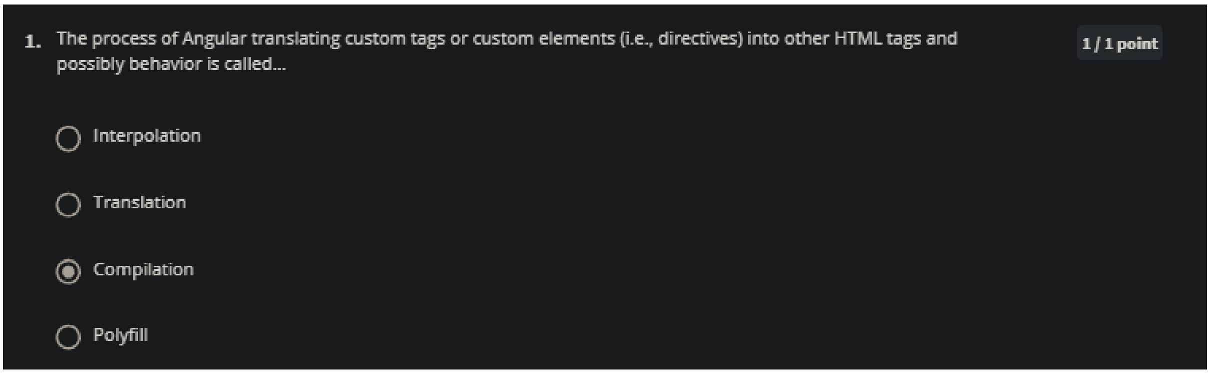
○ payload

● data

○ http

○ header.data

Quiz 26



1. The process of Angular translating custom tags or custom elements (i.e., directives) into other HTML tags and possibly behavior is called...

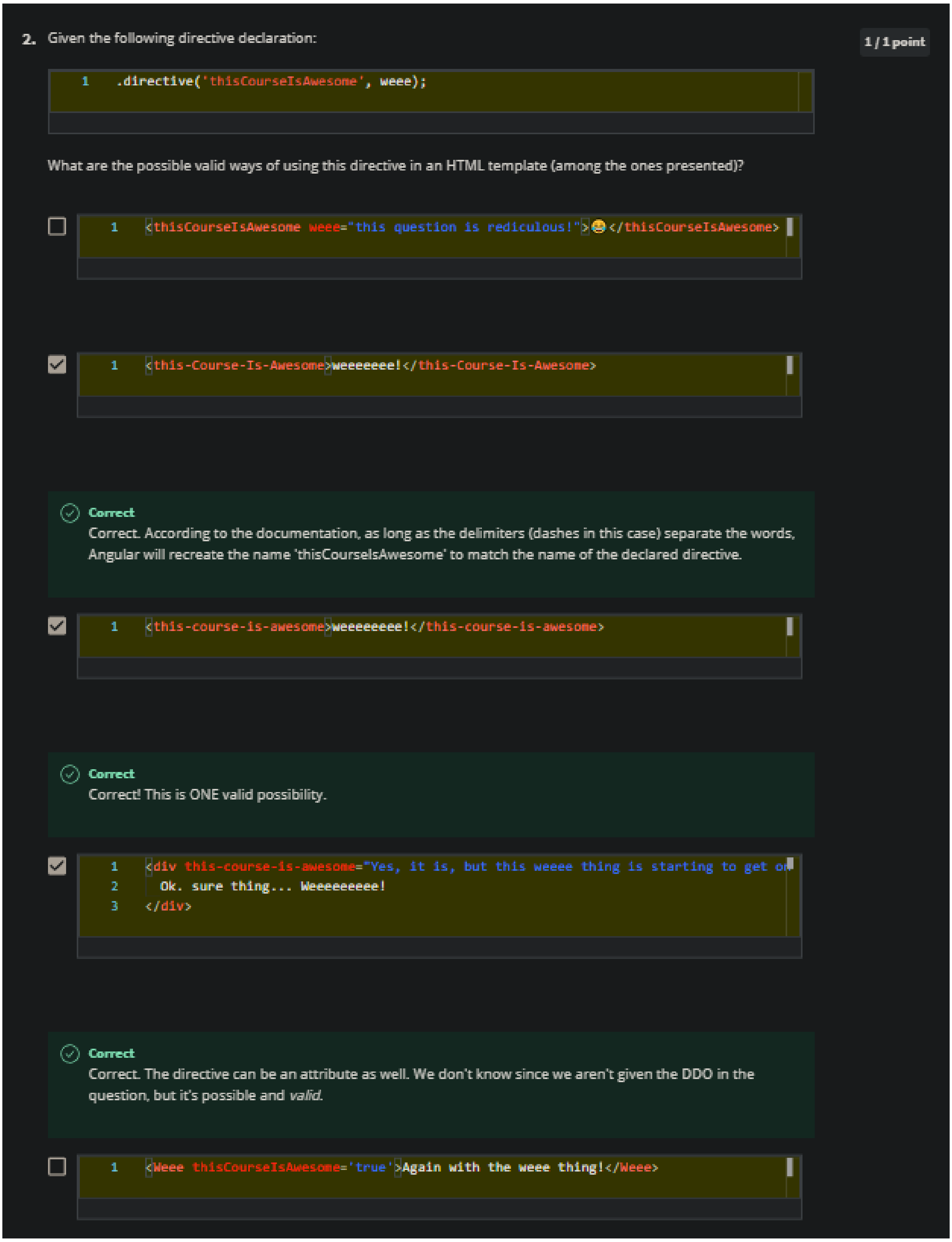
1 / 1 point

○ Interpolation

○ Translation

● Compilation

○ Polyfill



2. Given the following directive declaration:

1 / 1 point

```
.directive('thisCourseIsAwesome', weee);
```

What are the possible valid ways of using this directive in an HTML template (among the ones presented)?

☐
```
<thisCourseIsAwesome weee="this question is rediculous!">😂</thisCourseIsAwesome>
```

☑
```
<this-Course-Is-Awesome>weeeeeee!</this-Course-Is-Awesome>
```

✓ Correct
Correct. According to the documentation, as long as the delimiters (dashes in this case) separate the words, Angular will recreate the name 'thisCourseIsAwesome' to match the name of the declared directive.

☑
```
<this-course-is-awesome>weeeeeeee!</this-course-is-awesome>
```

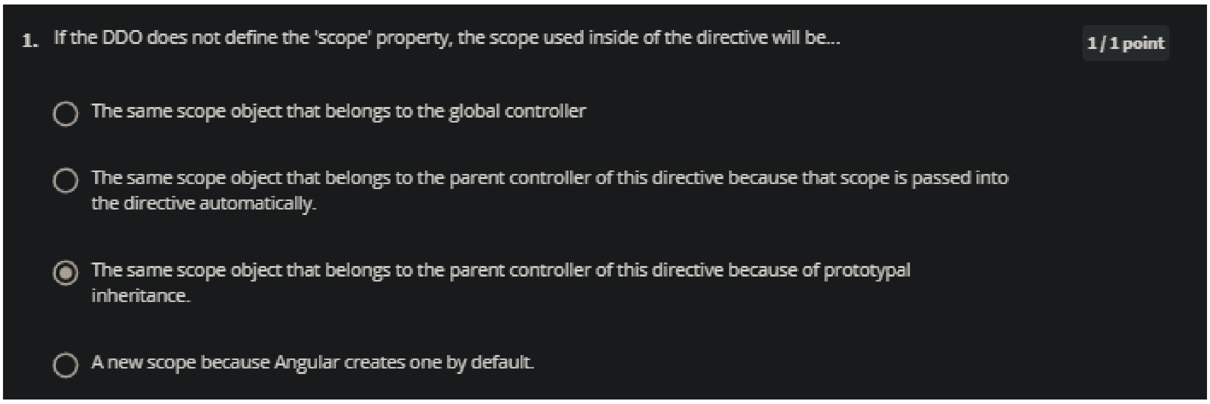
✓ Correct
Correct! This is ONE valid possibility.

☑
```
<div this-course-is-awesome="Yes, it is, but this weeee thing is starting to get on
  Ok. sure thing... Weeeeeeeee!
</div>
```

✓ Correct
Correct. The directive can be an attribute as well. We don't know since we aren't given the DDO in the question, but it's possible and *valid*.

☐
```
<Weee thisCourseIsAwesome='true'>Again with the weee thing!</Weee>
```

Quiz 27



1. If the DDO does not define the 'scope' property, the scope used inside of the directive will be...

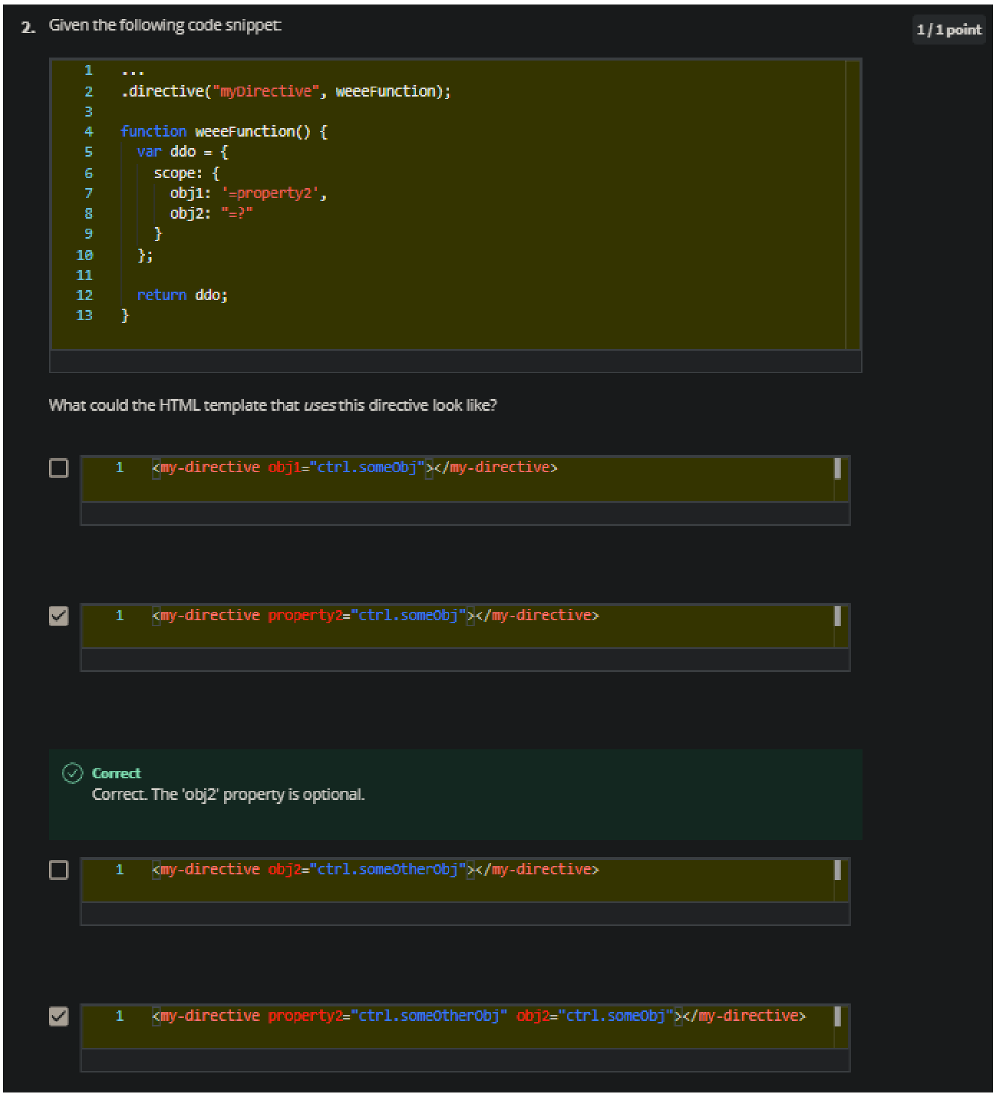
1 / 1 point

○ The same scope object that belongs to the global controller

○ The same scope object that belongs to the parent controller of this directive because that scope is passed into the directive automatically.

● The same scope object that belongs to the parent controller of this directive because of prototypal inheritance.

○ A new scope because Angular creates one by default.



2. Given the following code snippet:

1 / 1 point

```
...
.directive("myDirective", weeeFunction);

function weeeFunction() {
  var ddo = {
    scope: {
      obj1: '=property2',
      obj2: "=?"
    }
  };

  return ddo;
}
```

What could the HTML template that *uses* this directive look like?

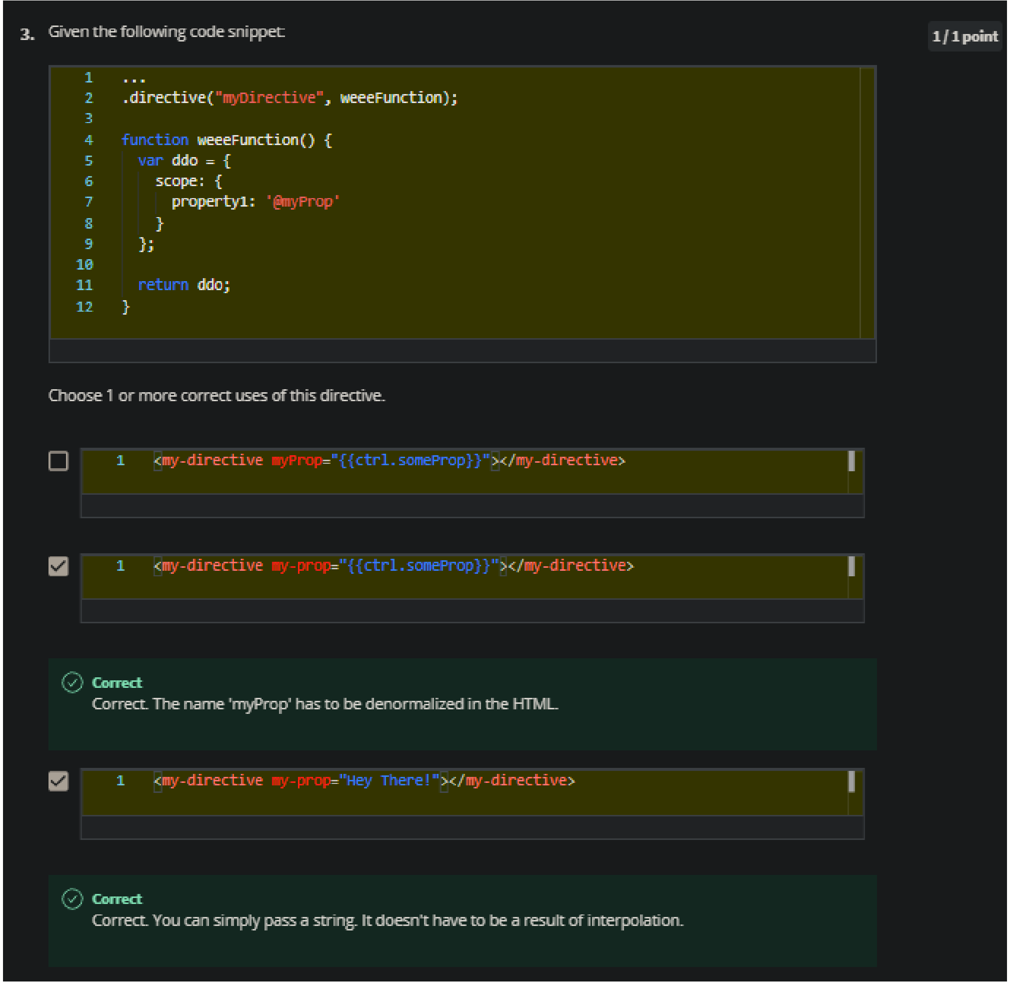
☐
```
<my-directive obj1="ctrl.someObj"></my-directive>
```

☑
```
<my-directive property2="ctrl.someObj"></my-directive>
```

✓ **Correct**
Correct. The 'obj2' property is optional.

☐
```
<my-directive obj2="ctrl.someOtherObj"></my-directive>
```

☑
```
<my-directive property2="ctrl.someOtherObj" obj2="ctrl.someObj"></my-directive>
```



3. Given the following code snippet:

1 / 1 point

```
...
.directive("myDirective", weeeFunction);

function weeeFunction() {
  var ddo = {
    scope: {
      property1: '@myProp'
    }
  };

  return ddo;
}
```

Choose 1 or more correct uses of this directive.

☐
```
<my-directive myProp="{{ctrl.someProp}}"></my-directive>
```

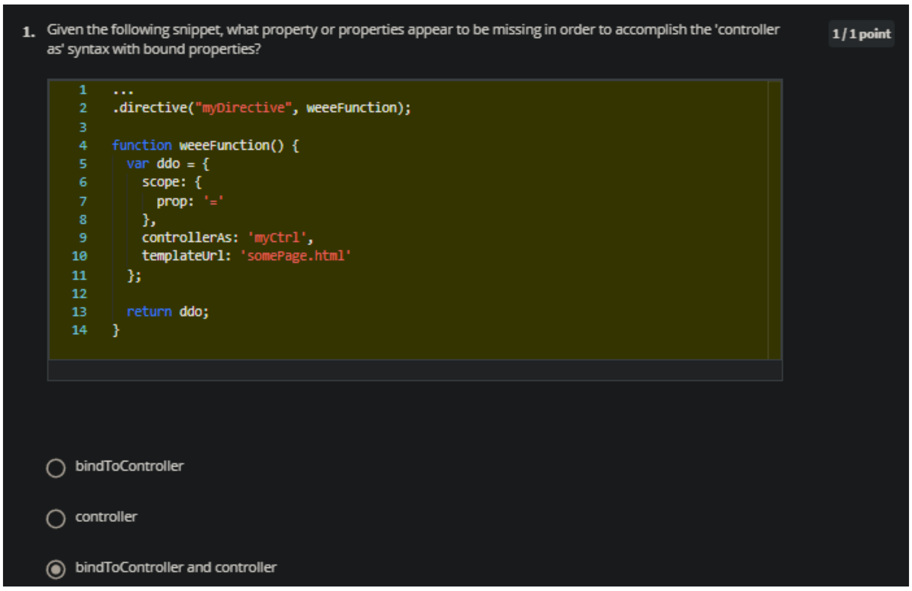
☑
```
<my-directive my-prop="{{ctrl.someProp}}"></my-directive>
```

✓ **Correct**
Correct. The name 'myProp' has to be denormalized in the HTML.

☑
```
<my-directive my-prop="Hey There!"></my-directive>
```

✓ **Correct**
Correct. You can simply pass a string. It doesn't have to be a result of interpolation.

Quiz 28



1. Given the following snippet, what property or properties appear to be missing in order to accomplish the 'controller as' syntax with bound properties?

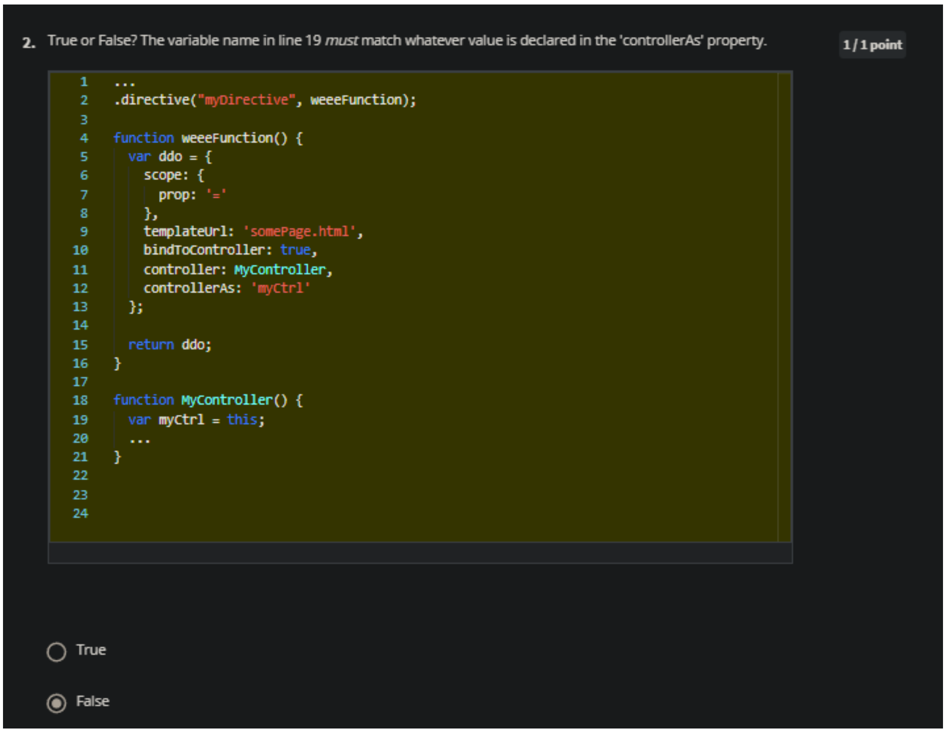
1 / 1 point

```
...
.directive("myDirective", weeeFunction);

function weeeFunction() {
  var ddo = {
    scope: {
      prop: '='
    },
    controllerAs: 'myCtrl',
    templateUrl: 'somePage.html'
  };

  return ddo;
}
```

○ bindToController

○ controller

● bindToController and controller



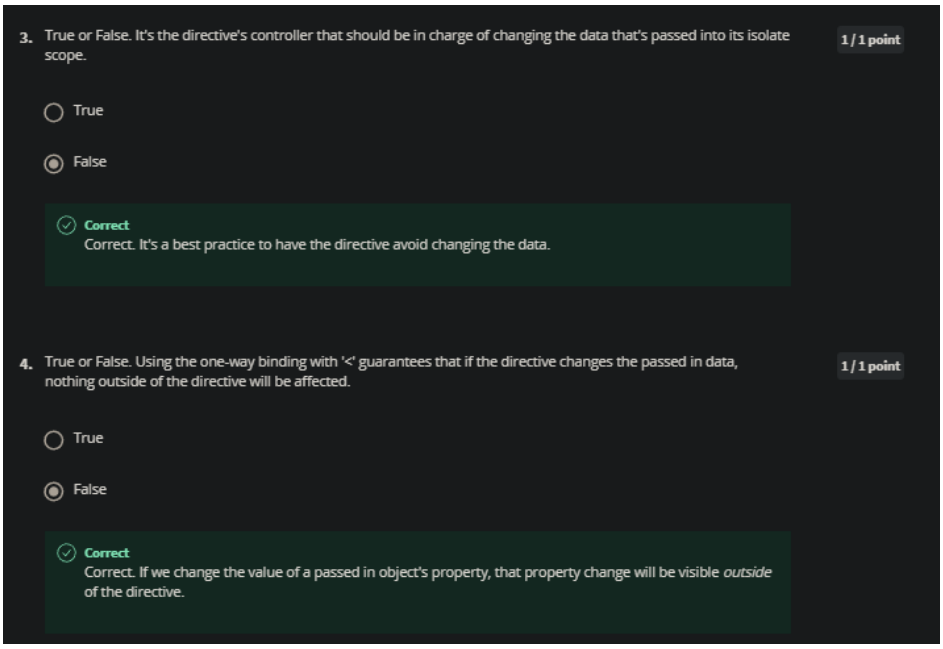
2. True or False? The variable name in line 19 *must* match whatever value is declared in the 'controllerAs' property.

1 / 1 point

```
1   ...
2   .directive("myDirective", weeeFunction);
3
4   function weeeFunction() {
5     var ddo = {
6       scope: {
7         prop: '='
8       },
9       templateUrl: 'somePage.html',
10      bindToController: true,
11      controller: MyController,
12      controllerAs: 'myCtrl'
13    };
14
15    return ddo;
16  }
17
18  function MyController() {
19    var myCtrl = this;
20    ...
21  }
22
23
24
```

○ True

● False



3. True or False. It's the directive's controller that should be in charge of changing the data that's passed into its isolate scope.

1 / 1 point

○ True

● False

Correct
Correct. It's a best practice to have the directive avoid changing the data.

4. True or False. Using the one-way binding with '<' guarantees that if the directive changes the passed in data, nothing outside of the directive will be affected.

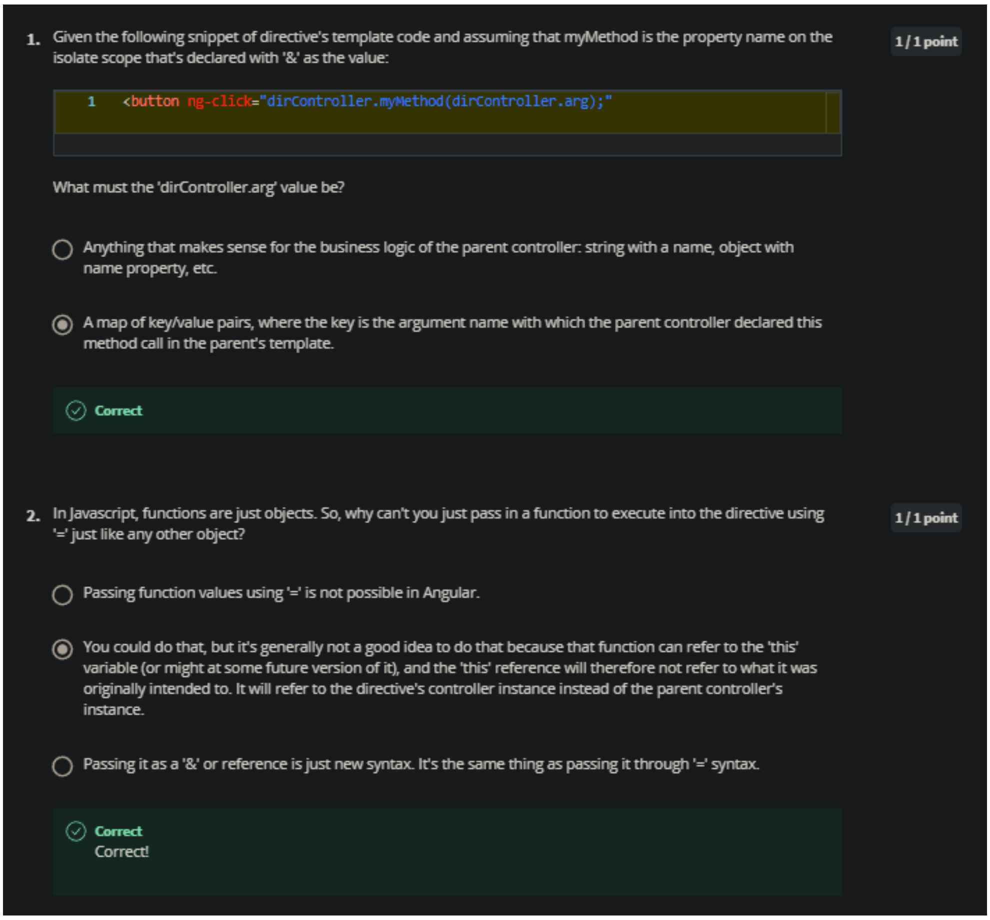
1 / 1 point

○ True

● False

Correct
Correct. If we change the value of a passed in object's property, that property change will be visible *outside* of the directive.

Quiz 29



1. Given the following snippet of directive's template code and assuming that myMethod is the property name on the isolate scope that's declared with '&' as the value:

1 / 1 point

```
<button ng-click="dirController.myMethod(dirController.arg);"
```

What must the 'dirController.arg' value be?

○ Anything that makes sense for the business logic of the parent controller: string with a name, object with name property, etc.

● A map of key/value pairs, where the key is the argument name with which the parent controller declared this method call in the parent's template.

✓ Correct

2. In Javascript, functions are just objects. So, why can't you just pass in a function to execute into the directive using '=' just like any other object?

1 / 1 point

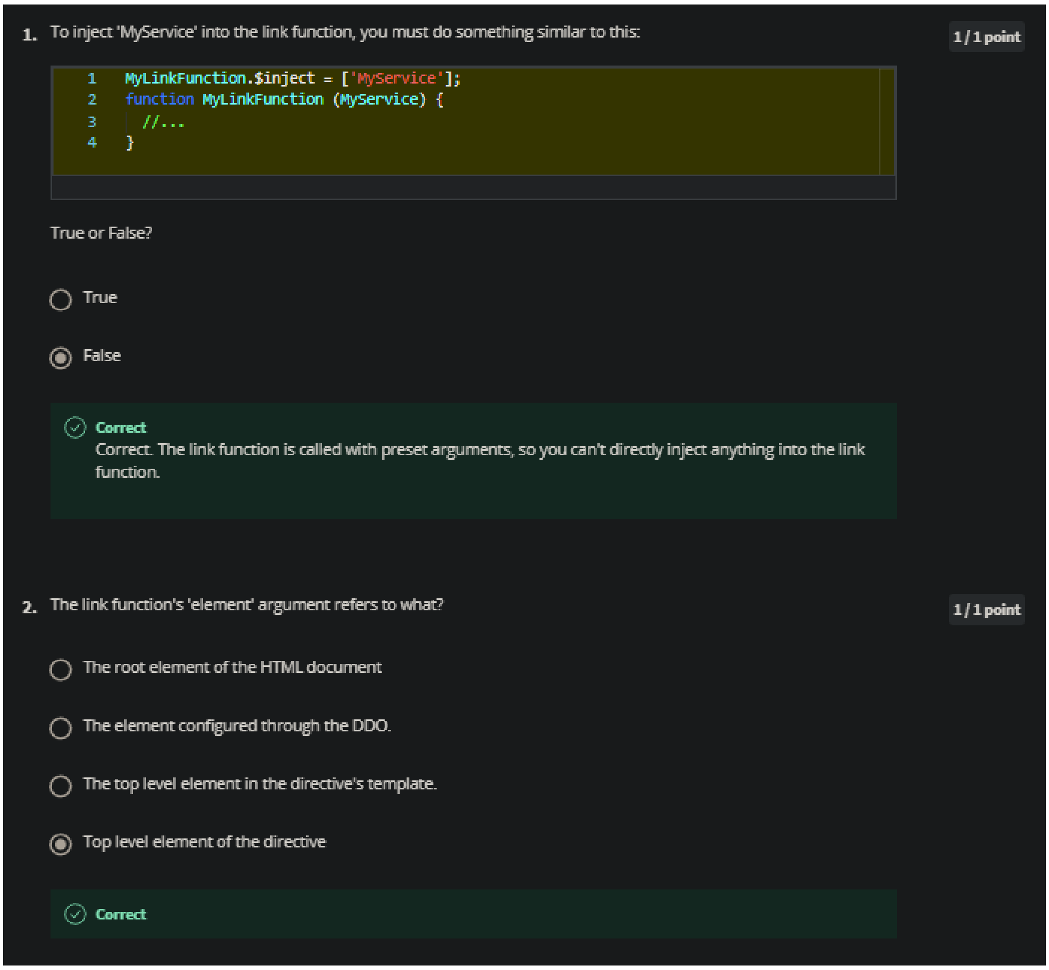
○ Passing function values using '=' is not possible in Angular.

● You could do that, but it's generally not a good idea to do that because that function can refer to the 'this' variable (or might at some future version of it), and the 'this' reference will therefore not refer to what it was originally intended to. It will refer to the directive's controller instance instead of the parent controller's instance.

○ Passing it as a '&' or reference is just new syntax. It's the same thing as passing it through '=' syntax.

✓ Correct
Correct!

Quiz 30



1. To inject 'MyService' into the link function, you must do something similar to this:

1 / 1 point

```
MyLinkFunction.$inject = ['MyService'];
function MyLinkFunction (MyService) {
  //...
}
```

True or False?

○ True

● False

✓ Correct
Correct. The link function is called with preset arguments, so you can't directly inject anything into the link function.

2. The link function's 'element' argument refers to what?

1 / 1 point

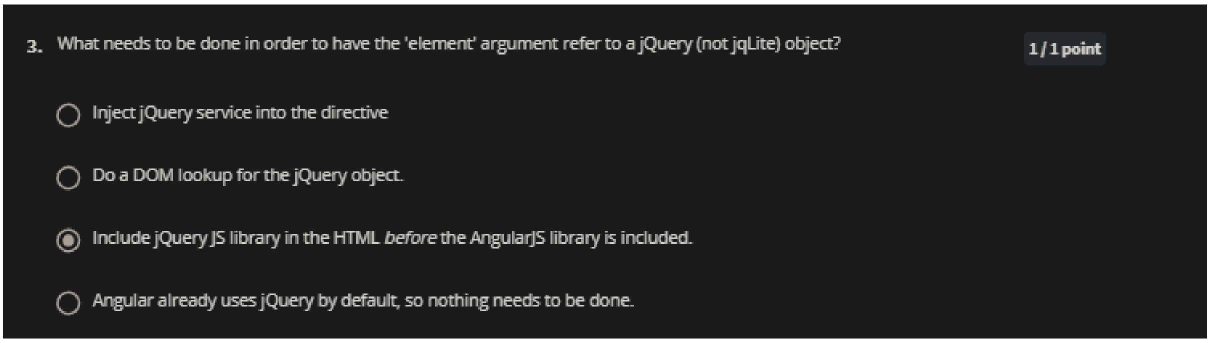
○ The root element of the HTML document

○ The element configured through the DDO.

○ The top level element in the directive's template.

● Top level element of the directive

✓ Correct



3. What needs to be done in order to have the 'element' argument refer to a jQuery (not jqLite) object?

1 / 1 point

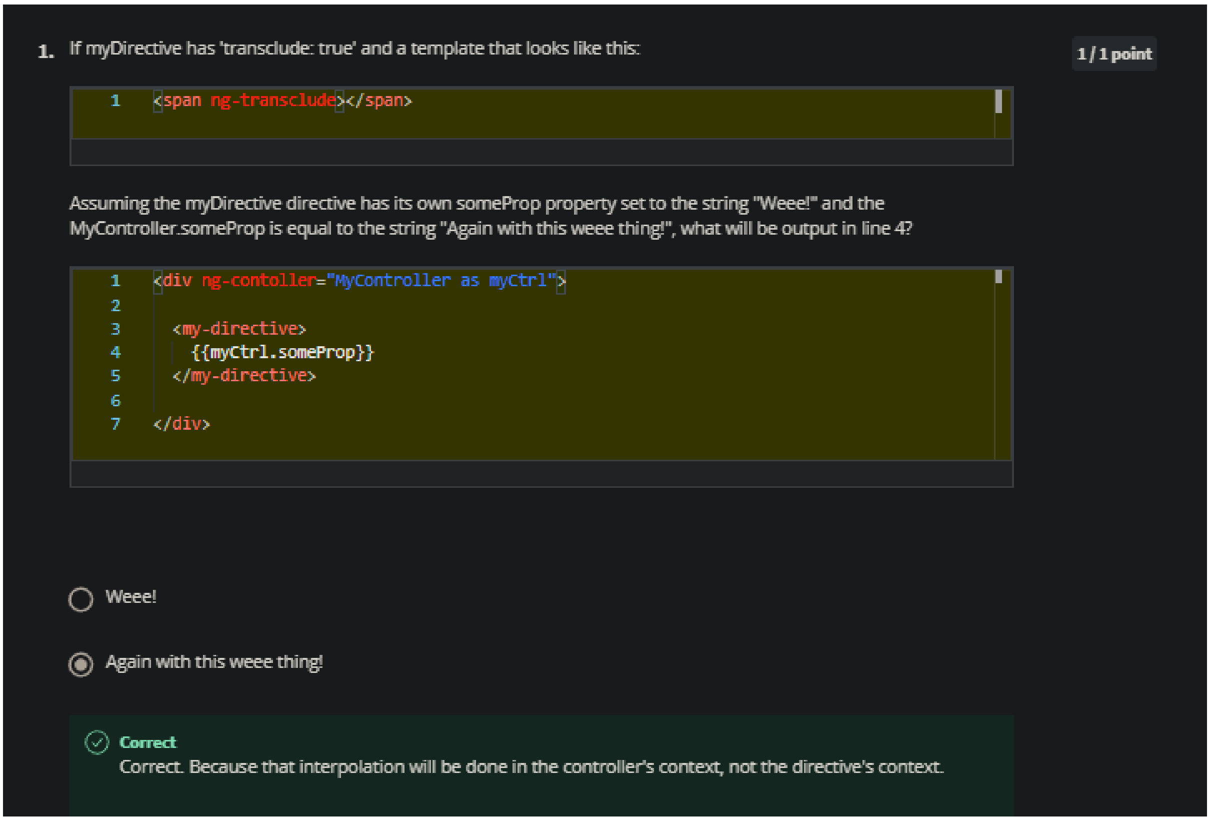
○ Inject jQuery service into the directive

○ Do a DOM lookup for the jQuery object.

● Include jQuery JS library in the HTML *before* the AngularJS library is included.

○ Angular already uses jQuery by default, so nothing needs to be done.

Quiz 31



1. If myDirective has 'transclude: true' and a template that looks like this:

1 / 1 point

```
<span ng-transclude></span>
```

Assuming the myDirective directive has its own someProp property set to the string "Weee!" and the MyController.someProp is equal to the string "Again with this weee thing!", what will be output in line 4?

```
1   <div ng-contoller="MyController as myCtrl">
2
3     <my-directive>
4       {{myCtrl.someProp}}
5     </my-directive>
6
7   </div>
```

○ Weee!

● Again with this weee thing!

Correct

Correct. Because that interpolation will be done in the controller's context, not the directive's context.