

Deep dive into Angular.js directives



Jesús Rodríguez Rodríguez

By the author of angular-tips.com



Deep dive into Angular.js directives

Jesús Rodríguez Rodríguez

This book is for sale at <http://leanpub.com/angularjsdirectives>

This version was published on 2015-05-07



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2014 - 2015 Jesús Rodríguez Rodríguez

Tweet This Book!

Please help Jesús Rodríguez Rodríguez by spreading the word about this book on [Twitter](#)!

The suggested hashtag for this book is [#angularjs](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#angularjs>

Contents

1. Transclusion	1
1.1 transclude: true	2

1. Transclusion

So here it comes the transclusion. We heard a lot about transclusion in the different media (forums, twitter, IRC) thinking that transclusion is something really really complex so they had to use a weird word to describe it. In reality, transclusion is a pretty easy concept and we are going to tackle it here.

So what is all this transclusion about?

Imagine you have this html:

```
1 <div my-directive>
2   <span>Hello there my friend</span>
3 </div>
```

And the directive is like:

```
1 angular.module('app').directive('myDirective', function() {
2   return {
3     template: 'This is my-directive'
4   };
5 });
```

So we know that when we put a template inside our directive, it will be placed inside the element which contains our directive (the div in our case). Good, but what happen when have some html content already inside that element? Are we going to see the template one? The one that is already there? Both? None?

This is my-directive

If you inspect the element, you can see:

```
1 <body ng-controller="MainCtrl" class="ng-scope">
2   <div my-directive="">This is my-directive</div>
3 </body>
```

So there the `` was replaced with our directive's template. That can be good for some cases, but for other cases, we would love to grab that existent html and use it on our directive's template. Can we do that? Yeah, we can **transclude** it!

So basically transclusion is a way of grabbing that html and be able to work with it inside our directive. Keep reading!

1.1 transclude: true

Imagine we want to create a directive to show some card elements. Those cards are composed of an `<h3>` as a header as well as some content like a couple of `<p>`. We could do something like:

```
1 $scope.content = '<h3>Card header</h3><p>This is my </p><p>lovely card</p>';

1 <card body="content"></card>
```

Yeah we could, but do you like that? We shouldn't put any DOM stuff on our controller and to be honest, it is a real pain. What if instead of that, we do:

```
1 <card>
2   <h3>Card header</h3>
3   <p>This is my </p>
4   <p>lovely card</p>
5 </card>
```

Isn't that way cleaner? I would say so.

So, in this kind of cases, where you want to use that html inside the directive's element for your own purposes, you need transclusion. Right, how do we use it?

We need two things: use the `transclude` property on the directive's DDO and also specify **where** we want to transclude (insert) that html in our directive's template.

Alright, how could we specify that? There is a directive called `ng-transclude` that we can put on an element of our template and then, the transcluded html (the one we are grabbing from "outside") is going to be inserted as a child of that element. Let's see our card directive:

```
1 angular.module('app').directive('card', function() {
2   return {
3     scope: {
4       color: '@'
5     },
6     transclude: true,
7     template: '<div class="card {{color}}" ng-transclude></div>'
8   };
9 });
```

We set the `transclude` property to `true` and then on our template we created a simple `div` where we put the `ng-transclude` directive. That means that our transcluded html is going to sit **inside** that `div`.

Let's see it in action:


```
1 <card color="red">
2   <h3>Card number 1</h3>
3   <p>We like our first card</p>
4   <p>Simple but useful</p>
5 </card>
```

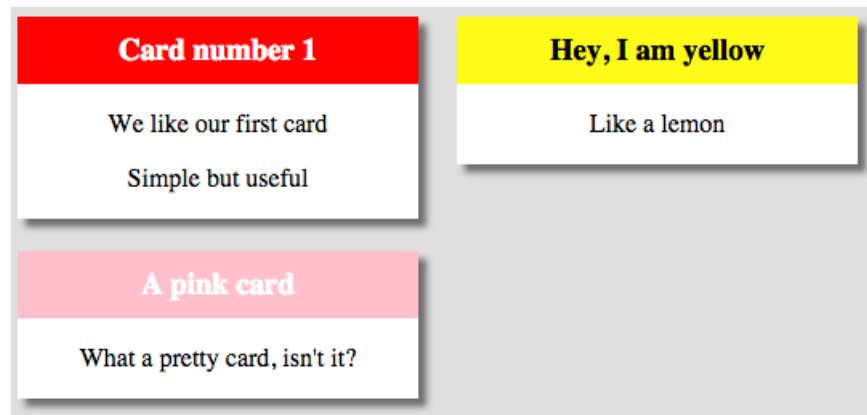
That will output an html like:

```
1 <card color="red" class="ng-isolate-scope">
2   <div class="card red" ng-transclude="">
3     <h3 class="ng-scope">Card number 1</h3>
4     <p class="ng-scope">We like our first card</p>
5     <p class="ng-scope">Simple but useful</p>
6   </div>
7 </card>
```

You see? It wasn't any complicated :)



There is another way of inserting the transcluded html into our template which we are going to see later on this book.



A couple of cards

See them [working](http://plnkr.co/edit/UDWxE01FfDHUcoDdRJgS?p=preview)¹

The tests

Time to test it!

Let's setup our test:

¹<http://plnkr.co/edit/UDWxE01FfDHUcoDdRJgS?p=preview>

```
1 describe('directive: card', function() {
2   var element;
3
4   beforeEach(module('app'));
5
6   beforeEach(inject(function($rootScope, $compile) {
7     element = angular.element(
8       '<card color="red"><h3>Hello world</h3><p>content</p></card>'
9     );
10
11     $compile(element)($rootScope);
12     $rootScope.$digest();
13   }));
14 });
```

Nothing that will surprise us now.

The tests are easy as well:

```
1 it('should contain a red card', function() {
2   var card = element.find('div.card');
3
4   expect(card.attr('class')).toContain('red');
5 });
6
7 it('should contain the header and the content', function() {
8   var header = element.find('h3').text();
9   var p = element.find('p').text();
10
11   expect(header).toBe('Hello world');
12   expect(p).toBe('content');
13 });
```

We just need to test that it sets the right color on it and also that the template contains our transcluded html.

Check the test [here](http://plnkr.co/edit/PmDuC87ncjkFgpgauWDD?p=preview)²

Summary

With `transclude: true` we can get that DOM that lies inside our directive and insert it on the directive's template where we want.

²<http://plnkr.co/edit/PmDuC87ncjkFgpgauWDD?p=preview>