



DIRECTIVES

BUILDING CUSTOMER DIRECTIVES IN ANGULAR 1.X

JON DEJONG

Chief Technologist @objectpartners

@jondejong

github.com/jondejong/directives-demo

WHO IS JON?

Chief Technologist @objectpartners

@jondejong

github/jondejong

WHAT IS ANGULARJS?

Angular is way to bind data between HTML and a JavaScript object (controller)

Angular is way to bind data between HTML and a JavaScript
object (controller)

...And share data between controllers with services

Angular is way to bind data between HTML and a JavaScript object (controller)

...And share data between controllers with services

...And extend HTML via custom directives

Angular is way to bind data between HTML and a JavaScript object (controller)

...And share data between controllers with services

...And extend HTML via custom directives

...Oh and handle routing

Angular is way to bind data between HTML and a JavaScript object (controller)

...And share data between controllers with services

...And extend HTML via custom directives

...Oh and handle routing

...and a kitchen sink

CONTROLLER

JavaScript:

```
angular
  .module('playground')
  .controller('HelloController', function () {
    var self = this;
    self.message = 'Hello MidwestJSers'
  });
```

HTML:

```
<div ng-controller="HelloController as hello">
  {{hello.message}}
</div>
```

DIRECTIVES

JAVASCRIPT

```
angular
  .module('playground')
  .directive('helloWorld', function () {
    return {
      restrict: 'E',
      template: '<p>Hello MidwestJS</p>'
    };
  });
```

DIRECTIVES

HTML

```
<hello-world></hello-world>
```

DIRECTIVES

OUTPUT

```
<hello-world class="ng-scope">  
  <p>Hello MidwestJS</p>  
</hello-world>
```

WHAT MAKES UP A DIRECTIVE?

TEMPLATE

```
template: "<div>Hello World</div>"
```

TEMPLATE URL

```
templateUrl: 'app/components/helloWorldTemplate/helloWorld.template.html'
```

RESTRICT

```
restrict: 'E' // Element  
restrict: 'A' // Attribute  
restrict: 'C' // Class  
  
restrict: 'AE' // Attribute or Element
```


SCOPE

```
scope: false //default
scope: true // New scope, inherits from the parent
scope: {} // Isolate Scope
scope: {property: '@'} // Grab the string of the attribute
scope: {property: '='} // Grab the object in the attribute
scope: {property: '&'} // Executable expression
```



Directive

```
scope: {  
  name: '@'  
},  
template: '<p>Hello {{name}}</p>'
```

HTML

```
<hello name='Jonny'></hello>
```

Output

Hello Jonny

Can be values from a controller

```
<hello name="{{ctrl.user.firstName}}"></hello>
```

=

Directive

```
scope: {  
  user: '='  
},  
template: '<p>Hello {{user.firstName}}</p>'
```

HTML

```
<hello user="{{ctrl.user}}"></hello>
```

Output

Hello Jonny

&

Directive

```
scope: {  
  callback: '&  
,  
link: function (scope) {  
  scope.callback()  
}
```

HTML

```
<hello callback={{ctrl.functionName()}}"></hello>
```

Output

Hello Jonny

LINK FUNCTION

HTML

```
<hello color='red' name='Lori'></hello>
```

JavaScript

```
scope: {  
  name: '@'  
},  
link: function (scope, el, attrs) {  
  console.log(scope.name) // Lori  
  
  element.class('fancy-class') // jqLite element  
  
  console.log(attrs.color) // red  
}
```

CONTROLLER

Directive

```
controller: DogListController,  
controllerAs: 'dlc',  
link: function (scope, el, attrs, dlc) {  
    // Controller is injected in here for  
    // use in the link function.  
    // ie, can access dlc.dogs  
}
```

Controller

```
/** @ngInject */  
function DogListController(DogService) {  
    var dlc = this;  
    dlc.dogs = [];  
  
    DogService.getDogs().then(function (dogs) {  
        dlc.dogs = dogs;  
    });  
}
```

TRANSCLUDE

Fancy sounding made up word for "wrap the content"

most of the time

Also, from the docs:

transclude makes the contents of a directive with this option have access to the scope outside of the directive rather than inside.

code now please

Well, that's all.