

# AngularJS: Lesson 3

## Custom Directives

# Custom Directives

For creating reusable UI components  
or reusable, composable UI behaviors

# Defining a directive

Prefix directive names with your own short namespace to avoid collisions

```
angular.module('angularjsTutorial')
.directive('ajstDirective', function() {
  return {
    // Directive definition object
    // https://docs.angularjs.org/api/ng/service/$compile#directive-definition-object
  };
});

// Can also dependency-inject here
.directive('ajstDirective', ['$log', function($log) {
  }]);
```

# Including a directive

JS: words in directive name are camelCase

HTML: words in directive name are dash-separated

```
// JS
.directive('ajstDirective', function() {
});

// HTML
<ajst-directive></ajst-directive>
<div data-ajst-directive></div>
<div ajst-directive></div>
```

# Directive Template

HTML can be inline or retrieved through a URL

```
// Inline. Use this for small templates
.directive('ajstDirective', function() {
  return {
    template : '<div></div>'
  }
});

// Async
.directive('ajstDirective', function() {
  return {
    templateUrl : 'components/component-name/directive-template.html'
  }
});
```

# ‘restrict’

Define how a directive should be included

```
.directive('ajstDirective', function() {
  restrict : 'E' // only as an element : <ajst-directive></ajst-directive>
});

.directive('ajstDirective', function() {
  restrict : 'A' // only as an attribute : <div ajst-directive></div>
});

.directive('ajstDirective', function() {
  restrict : 'C' // only as a classname: <div class="ajst-directive"></div>
});
```

# ‘scope’

2 options:  
Inherited Scope  
Isolate Scope

```
// inherit scope from parent
.directive('ajstDirective', function() {
  // omit the 'scope' property
});

// create isolated scope
.directive('ajstDirective', function() {
  scope : {}
});
```

# Passing information to a directive

## Done through isolate scope & HTML attributes

```
.directive('ajstDirective', function() {  
  scope : {  
    string : '@' // match an attribute of the same name, expect string  
    anotherStr : '@differentName, // match an attribute named 'different-name'  
    boundVariable : '=', // two-way binding. Can also have a different name like above  
    expression : '&' // evaluable expression. Can also have a different name like above  
  }  
});
```

```
// HTML  
<ajst-directive  
  string="Title"  
  different-name="Subtitle"  
  bound-variable="mainCtrl.newTodoTitle"  
  expression="mainCtrl.doSomething()"  
></ajst-directive>
```



# ‘controller’

Custom injections come after

```
.directive('ajstDirective', function() {  
  return {  
    controller :function ($scope){  
  
    }  
  }  
})
```

# ‘link’

Called once template has been instantiated.

At this point, you have access to do DOM manipulation,  
add event listeners, etc.

Passed the following 4 parameters:  
`scope element attrs controller`

```
.directive('ajstDirective', function() {  
  link : function (scope, element, attrs, controller){  
    // element is the angular-wrapped DOM element of your template  
    element.on('click', ...);  
  }  
});
```

# ‘transclude’

When you want to pass HTML content to a  
directive

```
.directive('ajstDirective', function() {  
  transclude: true  
});
```

```
// Directive template
```

```
<div>  
  <p>some directive content</p>  
  <ng-transclude></ng-transclude>  
</div>
```

# Animation

Can take advantage of animation by using the \$animate service inside the 'link' function.

<https://docs.angularjs.org/guide/animations>

```
.directive('ajstDirective', ['$animate', function($animate) {  
  return {  
    link : function(scope, element, attrs) {  
      element.on('click', function() {  
        if(element.hasClass('clicked')) {  
          $animate.removeClass(element, 'clicked');  
        } else {  
          $animate.addClass(element, 'clicked');  
        }  
      });  
    }  
  }  
});
```

# Further Reading

Guide to directives

<https://docs.angularjs.org/guide/directive>

In-depth docs on directive options

[https://docs.angularjs.org/api/ng/service/\\$compile](https://docs.angularjs.org/api/ng/service/$compile)

How directive compilation works

<https://docs.angularjs.org/guide/compiler>