Lesson 16 – Additional Angular Attributes

**Such alliteration about attributes!**

In regards to AngularJS there are some attributes that you can add to an element in order to give it more capabilities. (You may be saying “But I forget what an attribute is!”)

**Reminder of attributes**

Consider the following code. ..

<p id=”myid”> This is a paragraph with an id in it </p>

The piece **id=”myid”** is an attribute. Basically, anything that is within that opening tag that isn’t the tag name, in this case **p**, is an attribute. Some attributes have the form of **something=”something”**. The first **something** is called the **key**. The second **something** is called a value.

Some attributes don’t have the form of the **key/value** pair. The following example demonstrates this.

<iframe width="560" height="315" src="//www.youtube.com/embed/XN30uehLlqI" frameborder="0" allowfullscreen></iframe>

The **allowfullscreen** in the opening tag is only one word. Thus it doesn’t fit the **key/value** pair exactly. That is ok though. The attribute does its job.

**Ng-Cloak**

The attribute **ng-cloak** makes the angular site look better in that everything shows up at the same time. Sometimes if **ng-cloak** is not on the page the page will flash and then change a little bit. Thus we also add a class **.ng-cloak** and give it a style of **display: hidden.**

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.ng-cloak{

display: none;

}

</style>

</head>

<body ng-app="myapp" ng-cloak class="ng-cloak">

<div ng-controller="mycontroller">

<span ng-click="">The word of the day is: {{myWord}}</span>

</div>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.21/angular.min.js"></script>

<script type="text/javascript">

var app = angular.module('myapp', []);

app.controller('mycontroller', ['$scope', function($scope) {

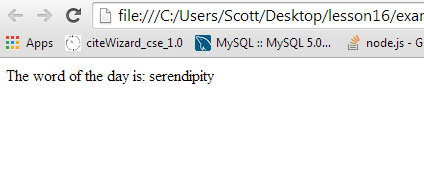
$scope.myWord = 'serendipity';

}]);

</script>

</body>

</html>



This example also has the variable **myWord** that is attached onto the **$scope**. Don’t forget, the **$scope** object is what talks to the HTML world from the JavaScript world. The variable **$scope.myWord** gets attached to the **{{myWord}}** in the HTML.

**The ng-click attribute**

The attribute **ng-click** has a function run when the element that has **ng-click** on it is clicked.

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.ng-cloak{

display: none;

}

.size{

width: 300px;

height: 300px;

}

.box-blue{

background-color: #B4CDCD;

}

</style>

</head>

<body ng-app="myapp" ng-clock class="ng-cloak">

<div class=" size box-blue" ng-controller="mycontroller">

<button ng-click="raise()"> Click Me </button> : {{count}}

</div>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.21/angular.min.js"></script>

<script type="text/javascript">

var app = angular.module('myapp', []);

app.controller('mycontroller', ['$scope', function($scope) {

$scope.count = 0;

$scope.raise = function(){

$scope.count= $scope.count + 1;

}

}]);

</script>

</body>

</html>





These two screenshots show the page when it is loaded and then when the button is clicked. When the button is clicked, the number increments on the page.

**The ng-mouseenter Attribute**

This attribute makes it so that a function is ran whenever the mouse enters the element that has this attribute on it.

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.ng-cloak{

display: none;

}

.size{

width: 300px;

height: 300px;

}

.box-blue{

background-color: #B4CDCD;

}

.box-red{

background-color: #F80000 ;

}

</style>

</head>

<body ng-app="myapp" ng-clock class="ng-cloak">

<div ng-controller="mycontroller">

<div class=" size box-blue " ng-mouseenter="stuff()">{{message}}</div>

<div class=" size box-red " ng-mouseenter="stuff()"></div>

</div>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.21/angular.min.js"></script>

<script type="text/javascript">

var app = angular.module('myapp', []);

app.controller('mycontroller', ['$scope', function($scope) {

$scope.message = 'hello';

$scope.stuff = function(){

alert('hello');

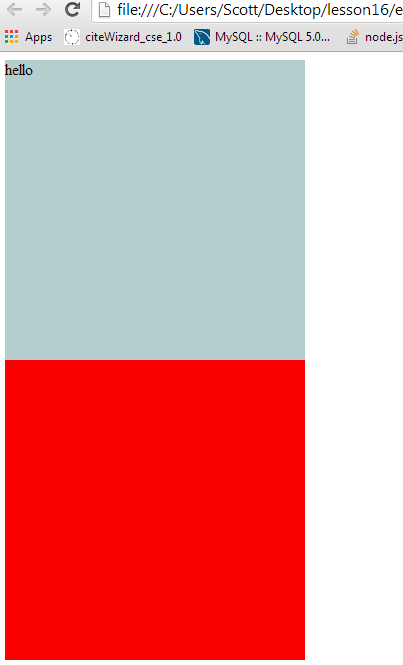
}

}]);

</script>

</body>

</html>



Both of the boxes have the function attached that make the **alert(‘Hello’)** run.

Here is another example. It is more interesting in that it changes the words in the box.

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

.ng-cloak{

display: none;

}

.size{

width: 300px;

height: 300px;

}

.box-blue{

background-color: #B4CDCD;

}

.box-red{

background-color: #F80000 ;

}

</style>

</head>

<body ng-app="myapp" ng-clock class="ng-cloak">

<div ng-controller="mycontroller">

<div class=" size box-blue " ng-mouseenter="stuff()">{{message}}</div>

<div class=" size box-red " ng-mouseenter="stuff()"></div>

</div>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.21/angular.min.js"></script>

<script type="text/javascript">

var app = angular.module('myapp', []);

app.controller('mycontroller', ['$scope', function($scope) {

$scope.message = 'hello';

$scope.stuff = function(){

$scope.message = 'You entered the box';

}

}]);

</script>

</body>

</html>