AngularJS: Lesson 2

Diving in to Angular Directives

The New York Code + Design Academy

AngularJS by Google

Directives!

The special markup in your HTML

Behaviors, event handlers, UI components

ng-app

The start of it all

Declares to Angular that it should control the content inside

```
<html ng-app=”myApp”>
</html>
```

ng-controller

Component-level control

Best practice is now to use the “controllerAs” syntax

```
<html ng-app=”myApp”>
<body ng-controller=”myController as ctrl”>
     <div ng-bind=”ctrl.property”></div>
</body>
</html>
```

ng-controller

Use “this” rather than “$scope”

```
// Don’t do this anymore
angular.module('angularjsTutorial')
  .controller('MainCtrl', function ($scope) {
    $scope.awesomeThings = [];
  });
```

```
// Instead, do this
angular.module('angularjsTutorial')
  .controller('MainCtrl', function () {
    this.awesomeThings = [];
  });
```

Also, “this” doesn’t inherit from parent.

ng-bind and {{}}

One-way data-binding

```
<!-- These two have the same result -->

<span ng-bind="mainCtrl.message"></span>

<span>{{mainCtrl.message}}</span>
```

ng-bind avoids flash-of-unstyled-content issues

ng-model

Two-way data-binding

```
<input type="text" ng-model="mainCtrl.message"/>
```

ng-repeat

Loop your face off

With arrays

```
<div ng-repeat="awesomeThing in mainCtrl.awesomeThings">
     <h3>{{awesomeThing.title}}</h3>
</div>
```

With objects

```
<div ng-repeat="(key, value) in mainCtrl.awesomeThings">
     <h3>{{key}}</h3>
     <h3>{{value.title}}</h3>
</div>
```

ng-repeat

Creates a child scope!!!!!!!!!! use $parent

$index	number	iterator offset of the repeated element (0..length-1)
$first	boolean	true if the repeated element is first in the iterator.
$middle	boolean	true if the repeated element is between the first and last in the iterator.
$last	boolean	true if the repeated element is last in the iterator.
$even	boolean	true if the iterator position $index is even (otherwise false).
$odd	boolean	true if the iterator position $index is odd (otherwise false).

ng-show and ng-hide

```
<span ng-show=”mainCtrl.propertyIsTruthy”></span>

<span ng-hide=”mainCtrl.propertyIsTruthy”></span>
```

ng-if

Accepts an expression or function.

Completely removes element out of DOM if false.

ng-click

Accepts expression or function.

Can be placed on any element in the DOM.

Directives for UI Events

ngClick
ngDblclick
ngMousedown
ngMouseup
ngMouseover
ngMouseenter
ngMouseleave
ngMousemove

ngPaste
ngCut
ngCopy
ngFocus
ngBlur
ngSubmit
ngKeypress
ngKeyup/ngKeydown

And all the rest

https://docs.angularjs.org/api/ng/directive

Angular Built In Services

Get Auto-magically Injected into our controllers

Usually denoted with a $

$scope

Used to communicate with the view (html)

Also used to create $watch(ers).

$scope.$watch

Used to watch a single attribute on the scope

accepts a property or function and then event handler function

$scope.$watch([‘property’ | function()], function(newVal, oldVal))

$scope.$watchCollection

Used to watch arrays and objects for insertions/deletions

$scope.$watch equality

Used to watch the changes to objects in your collections

$timeout

Used to run a function after a certain delay.

Unlike javascript setTimeout it will trigger an angular digest cycle.

$interval

Similar to $timeout but will repeatedly occur every specified period.

Angular Utility Functions

Complete list https://docs.angularjs.org/api/ng/function

angular.equals

Deep comparison of objects

angular.forEach(array, function(object))

Iterates over an array passing each object to you function

Let’s talk about the homework!