**Fundamentals – Angular.JS**

**by Todd Bagley**

**ANGULAR.JS** – First version of Angular. “Angular 1.0”

**ANGULAR** – Second and subsequent versions of Angular. “Angular 2, Angular 4, Angular 5”

**JQUERY** – Angular plays well with jQuery.

**RESPONSIVE** – Goal here is for browser to load core HTML and JavaScript ONCE, then support the user experience with light and fast JSON client-server communications useful for DOM manipulation via JavaScript [side note: prior to discovering Angular or React, I’ve used JavaScript and jQuery to build highly-responsive ajax-enabled single-page applications user experiences since 2009 = 9 yrs experience].

**DIRECTIVE** – marker on a HTML tag that tells Angular to run or reference some JavaScript code.

**MODULES** – where developers write pieces of Angular application, enabling encapsulation of code.

**EXPRESSIONS** – use of {{ }} to insert dynamic values into HTML.

**CONTROLER** – where developers define app behavior by defining functions and values.

**CLOSURE** – an anonymous function inside which JavaScript may exist (keeps code out of global scope of the JavaScript engine’s Execution Context).

**BUILT-IN DIRECTIVES** – special HTML marker directives:

**ng-app** – attach the Application Module to the page

**ng-class** – class control

**ng-click** – updates property and re-evaluates expressions via “2-way Data Binding”

**ng-controller** – attach a Controller function to the page

**ng-include** – include an html file

**ng-init** – enables evaluation of an expression in the current scope (for prototyping)

**ng-model** – binds a form element value to a property

**ng-show / ng-hide** – display a section based on an Expression

**ng-src** – <img ng-src=”{{ product.imageUrl }}” />

**ng-repeat** – repeat a section for each item in an Array

**BUILT-IN FILTERS** – calls to special data formatting {{ data | filter:options }}

**date** – {{ ‘1388123412323’ | date:’yyyy-MM-dd h:mm:ss’ }} => “2013-12-27 12:50:00”

**uppercase & lowercase** – {{ ‘hello world’ | uppercase }} => “HELLO WORLD”

**limitTo** – {{ ‘ABC123DEF456’ | limitTo:5 }} => “ABC12”

**limitTo** – <li ng-repeat=”product in store.products | limitTo:3”> => “shows first three products”

**orderBy** – <li ng-repeat=”product in store.products | orderBy:’-price’”> => “sorts descending”

**orderBy** – <li ng-repeat=”product in store.products | orderBy:’price’”> => “sorts ascending”

**BUILT-IN VALIDATIONS** – client-side validations for use with directives:

**novalidate** – tells browser to turn off default HTML validation

**required** – marks fields as required

**ng-pristine, ng-dirty, ng-invalid, ng-valid** – default Input classes characterizing value’s state

**CUSTOM DIRECTIVES** – custom HTML tags creating the opportunity to express the behavior of the app

**<some-tag></some-tag> => someTag** – dash in HTML translates to camelCase in JavaScript

**expand / replace a custom tag or attribute**

**controller logic** – can support if needed

**expressing complex UI**

**calling events and registering event handlers**

**reusing common components**

**restrict: ‘E’** – element = return { restrict: ‘E’. templateUrl: ‘[file.html]’ }

**restrict: ‘A’** – attribute = return { restrict: ‘A’. templateUrl: ‘[file.html]’ }

**controller, controllerAs** – return { restrict: ‘E’. templateUrl: ‘[file.html]’, controller: function{}, controllerAs: ‘[alias goes here]’ }

**BUILT-IN SERVICES** – client-side validations for use with directives:

**$http** – fetch JSON data from a web service**… returns Promise object!**

**$log** – log messages to the JavaScript console

**$filter** – filter an array

**CONTROLLER SERVICE** – funky array syntax:

var store = this;

store = [];

app.controller(‘SomeController’, [ ‘$http’, function($http){

// now $http injected into the controller as an arg

$http.get(‘/products.json’).success(function(data){

store.products = data;

});

} ]);