Angular JS

**OVERVIEW:**

* Extends HTML w/ new attributes
* Perfect for Single Page Applications (SPAs)
* Basics of Angular:
* Directives, Expressions, Modules, Controllers, Filters
* Other Areas:
* Events, DOM, Forms, Input, Validation, HTTP

**INTRODUCTION:**

\*Angular 🡪 a JavaScript framework – can be added to an HTML page w/ a <script> tag

\*Angular extends HTML attributes w/ Directives & binds data to HTML w/ Expressions

* AngularJS is a JS framework 🡪 a library written in JavaScript
* Starts automatically when the web page has loaded
* Is distributed as a JavaScript file & can be added w/ a script tag:

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>

**Directives:**

\*HTML attributes w/ an ng prefix

* **ng-app** directive:
* defines the application
* **creates a scope for an angular application**
* tells AngularJS that the outer most <div> element is the "owner" of an AngularJS **application**
* **ng-model** directive:
* marks the HTML element as a data model
* something else in app will use the value of this model
* binds the value of the input field to the application variable **name**
* **ng-bind** directive:
* takes the value of the model and puts it in the given element
* binds the innerHTML of this element to the value in the corresponding model
* binds the **innerHTML** of the given element to the application variable **name**

**Expressions:**

* AngularJS expressions are written inside double braces 🡪 **{{ expression }}**
* will "output" data exactly where the expression is written
* Expressions bind AngularJS data to HTML the same way as the **ng-bind** directive

**AngularJS Applications:**

* \*Modules 🡪 define
* \*Controllers 🡪 define
* ng-app directive 🡪 defines the application
* ng-controller directive 🡪 defines the controller

**EXPRESSIONS:**

* AngularJS expressions can be written inside double braces 🡪 {{ *expression* }}
* can also be written inside a directive 🡪 ng-bind="*expression*"
* AngularJS will resolve the expression & return the result exactly where the expression is written
* **AngularJS expressions** are much like **JavaScript expressions:** They can contain literals, operators, and variables 🡪 Ex: {{ 5 + 5 }} or {{ firstName + " " + lastName }}

\*Let Angular change the value of CSS properties\*

\*AngularJS 🡪 Number, Strings, Objects, Arrays

* AngularJS Expression VS JavaScript Expressions
* Like JavaScript expressions, AngularJS expressions can contain literals, operators, and variables.
* Unlike JavaScript expressions, AngularJS expressions can be written inside HTML.
* AngularJS expressions do not support conditionals, loops, and exceptions, while JavaScript expressions do.
* AngularJS expressions support filters, while JavaScript expressions do not.

**MODULES:**

\*AngularJS module defines an application

* Is a container for the different parts of an application
* Is a container for the application controllers
* Controllers always belong to a module

**DIRECTIVES:**

* Data Binding:
* Repeating HTML Elements:
* **ng-repeat** directive: repeats an HTML element
* **clones HTML elements** once for each item in a collection
* **ex: use on an array of objects**

**CONTROLLERS:**

\*Control the data of AngularJS applications

\*Is a JS Object – created by a standard JS Object Constructor

\*ng-controller 🡪 defines the application controller

<div ng-app="myApp" ng-controller="myCtrl">  
  
First Name: <input type="text" ng-model="firstName"><br>  
Last Name: <input type="text" ng-model="lastName"><br>  
<br>  
Full Name: {{firstName + " " + lastName}}  
  
</div>  
  
<script>  
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope) {  
    $scope.firstName = "John";  
    $scope.lastName = "Doe";  
});  
</script>

* The AngularJS application is defined by **ng-app="myApp"**. The application runs inside the <div>
* The **ng-controller="myCtrl"** attribute is an AngularJS directive. It defines a controller
* The **myCtrl** function is a JavaScript function
* AngularJS will invoke the controller with a **$scope** object
* In AngularJS, $scope is the application object (the owner of application variables and functions)
* The controller creates two properties (variables) in the scope (**firstName** and **lastName**).
* The **ng-model** directives bind the input fields to the controller properties (firstName and lastName)