**Angular**

|  |  |
| --- | --- |
| **Angularjs** | Predefined JavaScript Framework |
| **Purpose** | Used to develop Single Page Applications (SPAs). |
| **Extends** | HTML DOM with additional attributes |
| **Open Source & Licensed Under** | Yes & Apache license version 2.0 |
| **Developed & Maintained BY** | Misko Hevery, Adam Abrons & Google |

**AngularJS Expressions**

It is written inside {{expression}} / ng-bind="expression", which contains literals, operators, and variables. AngularJS will evaluate the expression, and return the result. Example {{4 + 5}} or {{ firstName + " " + lastName }}

|  |  |
| --- | --- |
| **Program** | **Output** |
| <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">  </script>  <div ng-app><p>Expression Evaluation (4+5): {{ 4 + 5 }}</p></div> | 9 |
| <p>Change the text box color</p>  <div ng-app="" ng-init="myColour='lightblue'">  <input style="background-color:{{myColour}}" ng-model="myCol">  </div> |  |

**AngularJS Concept**

|  |  |  |
| --- | --- | --- |
| Concept | Program | Output |
| Numbers | <script src=" angular.min.js"></script>  <div ng-app="" ng-init="len=1;breath=5">  <p>Rectangle Area: {{ len \* breath }}</p>  </div> | 5 |
| Strings | <div ng-app="" ng-init="firstName='wisdom';lastName='materials'">  Complete name: {{ firstName + " " + lastName }}  </div> | wisdom materials |
|  | <div ng-app="" ng-init="firstName='wisdom';lastName='materials'">  Complete name: <span ng-bind="firstName + ' ' + lastName"></span>  </div> | wisdom materials |
| Objects | <div ng-app="" ng-init="company={firstName:'wisdom',lastName:'materials'}">  Company Name : {{ company.lastName }}  </div> |  |
|  | <div ng-app="" ng-init="company={firstName:'wisdom',lastName:'materials'}">  Company Name  <span ng-bind="company.lastName"></span></div> | wisdom materials |
| Arrays | <div ng-app="" ng-init="nos=[1,2,5,4,6]">  <p>Arrays {{ nos }}</p> <p>Arrays {{ nos[2] }}</p>  </div> | Arrays [1,2,5,4,6]  Arrays 5 |

**AngularJS Modules:** It is a container for the application controllers which defines an application / it parts. Controllers always belong to a module.

|  |  |
| --- | --- |
| **Program** | **Output** |
| <script src=" angular.min.js"></script>  <div ng-app="myApp" ng-controller="myCtrl">  {{ firstName + " " + lastName }}  </div>  <script>  var app = angular.module("myApp", []);  app.controller("myCtrl", function($scope) {  $scope.firstName = "wisdom";  $scope.lastName = "materials";  });  </script> | wisdom materials |

**AngularJS for Creating New Directives**

|  |  |
| --- | --- |
| **Program** | **Output** |
| <div ng-app="myApp" abcd1234></div>  <script>  var app9 = angular.module("myApp", []);  app9.directive("abcd1234", function () {  return { template: "Wisdom Materials constructored new directive!" };  });  </script> |  |

**AngularJS Directives:** It extended HTML attributes with the prefix ng-.

|  |  |
| --- | --- |
| **Directive** | **Details** |
| ng-app | Initializes an AngularJS application |
| ng-init | Initializes application data. |
| ng-model | Binds input field value to the application variable name. |
| **ng-controller** | **Defines the controller which control AngularJS applications.** |
| **ng-bind** | **Binds application data to the HTML view.** |
| **ng-repeat** | **Repeats an HTML element** |

**Ng-app directive:** initializes an AngularJS application

|  |  |
| --- | --- |
| Program | <p>Enter Firstname in textbox:</p>  <div ng-app="" ng-init="firstName=''">  <p>Name: <input type="text" ng-model="firstName"></p>  <p>Firstname Entered: {{ firstName }}</p>  </div> |
| Output |  |

**ng-init directive :** Initializes application data.

|  |  |
| --- | --- |
| Program | <div data-ng-app="" data-ng-init="len=2;bre=3">  <h2>Rect Area Calculation</h2>  Length: <input type="number" ng-model="len">  breath: <input type="number" ng-model="bre">  <p>Rect Area: {{len \* bre}}</p>  </div> |
| Output |  |

**ng-model directive : Binds input field value to the application variable name.**

|  |  |
| --- | --- |
| Program | <div ng-app="">  <b>Addition of Two Numbers Program in AngularJS</b><br /><br />  First Number:<input type="number" ng-model="a" /><br /><br />  Second Number:<input type="number" ng-model="b" /><br /><br />  Sum: {{ a + b }}  </div> |
| Output |  |

**Component**

It is basically a directive which is reusable code. Or it is a method/function that not only contains the controller logic required for a UI element to function & HTML tags to generate the element.

**Why Use Components**

Reduce duplication: Traditionally, if the same UI component had to be present in two different places on a page (let alone two different pages in the application), the HTML code had to be duplicated. This leads to not only an increase in code duplication but also a decrease in maintainability, making it painful to keep the UI consistent. Components serve as a simple solution to this problem.

**HTML bundled with logic:** Under the MVC architecture, having separate files for HTML tags and the controller logic may seem organized but has a downside — there is no clear way of showing which HTML content is changed or generated which part of the controller logic. Bundling both these into a single component makes it cleaner to understand their relationship while also ensuring that one does not get changed without considering the other part.

**Consistent UI:** It enables reusable HTML code, this practice forces you to keep your elements uniform wherever they are used.

**Component Communication**

Suppose we are making a messenger app. The main view is a page level component with a reusable contact list component. We want to use the contact list component in other areas of our application, so it’s important to make this as reusable as possible.

The contact list component has a single purpose. To render the list of available contacts from the server and output the selections made. The messenger component should accept selections made from the contact list component and append them to its recipients list.

**How to use external data in a component**

When a single-page application is made of several different components, data is usually loaded by a service and then passed to the components that need it. We could add a parameter to pass a name to our component, which would be used as follows:

|  |  |
| --- | --- |
| Program | <div ng-app="myApp">  <hello-world name="'wisdom materials'" > </hello-world>  </div>  <script>  angular.module("myApp", [])  .component("helloWorld",{  template: 'hi.... {{$ctrl.name}}!',  bindings: { name: '@' }  });  </script> |
| Output | hi.... 'wisdom materials' |

**Events and Binding with ng-model Directive**

It is used to bind the value of an input field to a variable created in AngularJS.

|  |  |
| --- | --- |
| Program | <p>ng-model bind data to input field using controller property.</p>  <div ng-app="wm" ng-controller="myCtrl">  Name: <input ng-model="name">  </div>  <script>  var app = angular.module('wm', []);  app.controller('myCtrl', function($scope) {  $scope.name = "wisdom materials";  });  </script> |
| Output |  |

**Two Way Binding**

|  |  |
| --- | --- |
| Program | <div ng-app="myApp" ng-controller="myCtrl">  Name: <input ng-model="name">  <h1>Entered Name: {{name}}</h1>  </div>  <script>  var app = angular.module('myApp', []);  app.controller('myCtrl', function($scope) {  $scope.name = "wisdom materials";  });  </script> |
| Output |  |

**Email Verification: Event**

|  |  |
| --- | --- |
| Program | <p>Email Verification</p>  <form ng-app="" name="myForm">  Email:  <input type="email" name="myAddress" ng-model="text">  <span ng-show="myForm.myAddress.$error.email">Not a valid e-mail address</span>  </form> |
| Output |  |

**AngularJS Services**

It is a function / object which may be user defined or predefined.

|  |  |
| --- | --- |
| Program | <div ng-app="myApp" ng-controller="myCtrl">  <h3> Page url : {{pageurl99}}</h3>  </div>  <script>  var app = angular.module('myApp', []);  app.controller('myCtrl', function($scope, $location) {  $scope.pageurl99 = $location.absUrl();  });  </script> |
| Output | URL of the Page is printed |

**Fetch Data from an http service, Submit data to http service**

|  |  |
| --- | --- |
| Program | <body ng-app="sampleApp">  <script src="https://code.angularjs.org/1.6.9/angular-route.js"></script>  <script src="https://code.angularjs.org/1.6.9/angular.min.js"></script>  <script src="https://code.angularjs.org/1.6.9/angular.js"></script>  <script src="lib/bootstrap.js"></script>  <script src="lib/bootstrap.css"></script>  <h1> Add Topic Names to UL tag</h1>  <div ng-controller="AngularController">  <form ng-submit="Display()">  Enter a Topic Name  <input type="text" ng-app="sampleApp" ng-model="Topic"><br>  <input type="submit" value="Submit"/>  <ul ng-repeat="topicname in AllTopic">  <li>{{topicname}}</li>  </ul>  </form>  </div>  <script>  var sampleApp = angular.module("sampleApp",[]);  sampleApp.controller("AngularController",function($scope) {  $scope.AllTopic=[];  $scope.Display = function () {  $scope.AllTopic.push($scope.Topic);  }  });  </script> |
| Output |  |

**http module and Observables**

**$http** is an AngularJS service for reading data from remote servers and returns a response.

|  |  |
| --- | --- |
| **Program Names** | **code** |
| a.html | <div ng-app="myApp" ng-controller="myCtrl">  <p>Today's message:</p><h1>{{msg99}}</h1></div>  <p>The $http service requests a page on the server, and the response is set as the value of the "myWelcome" variable.</p>  <script>  var app = angular.module('myApp', []);  app.controller('myCtrl', function($scope, $http) {  $http.get("welcome.htm")  .then(function(response) {  $scope.msg99 = response.data;  });  });  </script> |
| welcome.htm | Hello AngularJS Students |
| Output | Today's message  Hello AngularJS Students  The $http service requests a page on the server, and the response is set as the value of the "myWelcome" variable. |

**$http service methods:** .delete(), .get(), .head(), .jsonp(), .patch(), .post(), .put()

**$http service Properties:** The response from the server is an object with these properties:

|  |  |
| --- | --- |
| **Property** | Details |
| **.config** | the object used to generate the request. |
| **.data** | a string, or an object, carrying the response from the server. |
| **.headers** | a function to use to get header information. |
| **.status** | a number defining the HTTP status. |
| **.statusText** | .statusText a string defining the HTTP status. |

**Example**

|  |  |
| --- | --- |
| Program | <div ng-app="myApp" ng-controller="myCtrl">  <p>Data : {{content}}</p>  <p>Status : {{statuscode}}</p>  <p>StatusText : {{statustext}}</p>  </div>  <script>  var app = angular.module('myApp', []);  app.controller('myCtrl', function($scope, $http) {  $http.get("welcome.htm")  .then(function(response) {  $scope.content = response.data;  $scope.statuscode = response.status;  $scope.statustext = response.statusText;  });  });  </script> |
| Output | Data : Hello AngularJS Students  Status: 200  Status Text: |

**Routing in AngularJS**

It is used when the user wants to navigate to different pages in an application but still wants it to be a single-page application. AngularJS routes enable the user to create different URLs for different content in an application.

|  |  |
| --- | --- |
| Program | <!--ng-app tells AngularJS that myApp is the root element of the application -->  <html ng-app="myApp">  <head>  <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular.min.js">  </script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular-route.min.js">  </script>  <style>  body { text-align: center; font-family: Arial, Helvetica, sans-serif; }  h1 { color: green; }  </style>  </head>  <body>  <h3>Single Page Application in AngularJS</h3>  <!--hg-template indicates the pages that get loaded as per requirement-->  <script type="text/ng-template" id="first.html">  <h1>First Page</h1>  <h3>{{message}}</h3>  </script>  <script type="text/ng-template" id="second.html">  <h1>Second Page</h1> <h3>{{message}}</h3>  </script>  <script type="text/ng-template" id="third.html">  <h1>Third Page</h1> <h3>{{message}}</h3>  </script>  <!-- Hyperlinks to load different pages dynamically -->  <a href="#/">First</a> <a href="#/second">Second</a>  <a href="#/third">Third</a>  <!--ng-view includes the rendered template of the current route into the main page-->  <div ng-view></div>  <script>  var app = angular.module('myApp', []);  var app = angular.module('myApp', ['ngRoute']);  app.config(function ($routeProvider) {  $routeProvider  .when('/', { templateUrl: 'first.html', controller: 'FirstController' })  .when('/second', { templateUrl: 'second.html', controller: 'SecondController' })  .when('/third', { templateUrl: 'third.html', controller: 'ThirdController' })  .otherwise({ redirectTo: '/' });  });  // Controller (JS function) maintains application data & behavior using $scope object  // properties & methods can be attached to the  // $scope object inside a controller function  app.controller('FirstController', function ($scope) {  $scope.message = 'Hello from FirstController';  });  app.controller('SecondController', function ($scope) {  $scope.message = 'Hello from SecondController';  });  app.controller('ThirdController', function ($scope) {  $scope.message = 'Hello from ThirdController';  });  </script>  </body>  </html> |
| Output |  |