1. **AngularJS**

* AngularJS is a client-side JavaScript MVC framework to develop a dynamic web application.
* entirely based on HTML and JavaScript
* The ng-app directive is a starting point of AngularJS Application. It initializes the AngularJS framework automatically. AngularJS framework will first check for ng-app directive in a HTML document after the entire document is loaded and if ng-app is found, it bootstraps itself and compiles the HTML template.
* <body ng-app="myAngularApp">

<script>

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

</script>

* {{ expression }}
  1. Directives

| Directive | Description |
| --- | --- |
| ng-app | Auto bootstrap AngularJS application. |
| ng-init | Initializes AngularJS variables |
| ng-model | Binds HTML control's value to a property on the $scope object. |
| ng-controller | Attaches the controller of MVC to the view. |
| ng-bind | Replaces the value of HTML control with the value of specified AngularJS expression. |
| ng-repeat | Repeats HTML template once per each item in the specified collection. |
| ng-show | Display HTML element based on the value of the specified expression. |
| ng-readonly | Makes HTML element read-only based on the value of the specified expression. |
| ng-disabled | Sets the disable attribute on the HTML element if specified expression evaluates to true. |
| ng-if | Removes or recreates HTML element based on an expression. |
| ng-click | Specifies custom behavior when an element is clicked. |

<div ng-controller="myController">

{{message}}

</div>

<script>

var ngApp = angular.module('myNgApp', []);

ngApp.controller('myController', function ($scope) {

$scope.message = "Hello World!";

});

</script>

* $scope – inside controller
* $rootscope – overall ngapp

<body ng-app="myNgApp">

<div ng-controller="myController">

Enter Message: <input type="text" ng-model="message" /> <br />

<button ng-click="showMsg(message)">Show Message</button>

</div>

<script>

var ngApp = angular.module('myNgApp', []);

ngApp.controller('myController', function ($scope) {

$scope.message = "Hello World!";

$scope.showMsg = function (msg) {

alert(msg);

};

});

</script>

</body>

* 1. **Scope object methods:**

The $scope object contains various methods. The following table lists important methods of $scope object.

| Method | Description |
| --- | --- |
| $new() | Creates new child scope. |
| $watch() | Register a callback to be executed whenever model property changes. |
| $watchGroup() | Register a callback to be executed whenever model properties changes. Here, specify an array of properties to be tracked. |
| $watchCollection() | Register a callback to be executed whenever model object or array property changes. |
| $digest() | Processes all of the watchers of the current scope and its children.Â |
| $destroy() | Removes the current scope (and all of its children) from the parent scope. |
| $eval() | Executes the expression on the current scope and returns the result. |
| $apply() | Executes an expression in angular outside the angular framework. |
| $on() | Register a callback for an event. |
| $emit() | Dispatches the specified event upwards till $rootScope. |
| $broadcast() | Dispatches the specified event downwards to all child scopes. |

* 1. AngularJS event directives.

| Event Directive |
| --- |
| ng-blur |
| ng-change |
| ng-click |
| ng-dblclick |
| ng-focus |
| ng-keydown |
| ng-keyup |
| ng-keypress |
| ng-mousedown |
| ng-mouseenter |
| ng-mouseleave |
| ng-mousemove |
| ng-mouseover |
| ng-mouseup |

* 1. built-in AngularJS services.

|  |  |  |  |
| --- | --- | --- | --- |
| $anchorScroll | $exceptionHandler | $interval | $rootScope |
| $animate | $filter | $locale | $sceDelegate |
| $cacheFactory | $httpParamSerializer | $location | $sce |
| $templateCache | $httpParamSerializerJQLike | $log | $templateRequest |
| $compile | $http | $parse | $timeout |
| $controller | $httpBackend | $q | $window |
| $document | $interpolate | $rootElement |  |

* + 1. **$http is a service as an object. It includes following shortcut methods.**

| Method | Description |
| --- | --- |
| $http.get() | Perform Http GET request. |
| $http.head() | Perform Http HEAD request. |
| $http.post() | Perform Http POST request. |
| $http.put() | Perform Http PUT request. |
| $http.delete() | Perform Http DELETE request. |
| $http.jsonp() | Perform Http JSONP request. |
| $http.patch() | Perform Http PATCH request. |

* + 1. **HTTP:**

var promise = $http.get("/demo/getdata").success(onSuccess).error(onError);

$http.post('/demo/submitData', { myData: 'Hello World!' })

.success(onSuccess)

.error(onError);

* + 1. **LOG:**

$log.log('This is log.');

$log.error('This is error.');

$log.info('This is info.');

$log.warn('This is warning.');

$log.debug('This is debugging.');

**9.5. Angularjs filters:**

{{expression | filterName:parameter }}

| Filter Name | Description |
| --- | --- |
| Number | Formats a numeric data as text with comma and fraction. |
| Currency | Formats numeric data into specified currency format and fraction. |
| Date | Formats date to string in specified format. |
| Uppercase | Converts string to upper case. |
| Lowercase | Converts string to lower case. |
| Filter | Filters an array based on specified criteria and returns new array. |
| orderBy | Sorts an array based on specified predicate expression. |
| Json | Converts JavaScript object into JSON string |
| limitTo | Returns new array containing specified number of elements from an existing array. |

* 1. **Modules:**

Different files:

app.js:

var myApp = angular.module("myApp", []);

myController.js:

myApp.controller("myController", function ($scope) {

$scope.message = "Hello Angular World!";

});

* 1. **Forms:**

<form **ng-submit="submitStudnetForm()"** >

<label for="firstName" >First Name: </label><br />

<input type="text" id="firstName" ng-model="student.firstName" /> <br />

<label for="gender" >Gender</label> <br />

<select id="gender" ng-model="student.gender">

<option value="male">Male</option>

<option value="female">Female</option>

</select><br /> <br />

<input type="submit" value="Submit" />

<input type="reset" ng-click="resetForm()" value="Reset" />

</form>

* 1. AngularJS includes the following validation directives.

| Directive | Description |
| --- | --- |
| ng-required | Sets required attribute on an input field. |
| ng-minlength | Sets minlength attribute on an input field. |
| ng-maxlength | Sets maxlength attribute on an input field. Setting the attribute to a negative or non-numeric value, allows view values of any length. |
| ng-pattern | Sets pattern validation error key if the ngModel value does not match the specified RegEx expression. |

* 1. **Route:**

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

app.config(function ($routeProvider) {

$routeProvider.when('/', {

templateUrl: '/login.html',

controller: 'loginController'

}).when('/student/:username', {

templateUrl: '/student.html',

controller: 'studentController'

}).otherwise({

redirectTo: "/"

});

app.controller("loginController", function ($scope, $location) {

$scope.authenticate = function (username) {

// write authentication code here..

$location.path('/student/' + username)

};

});

app.controller("studentController", function ($scope, $routeParams) {

$scope.username = $routeParams.username;

});

* 1. **Exception Handling:**

app.config(function ($provide) {

$provide.decorator('$exceptionHandler', function ($delegate) {

return function (exception, cause) {

$delegate(exception, cause);

alert('Error occurred! Please contact admin.');

};

});

});