Angular:

2-way databinding

filters

directives

MVC or MVW

Notes:

* while defining any parameters, if it is string put in ‘ ’ single quotation and if it’s object put in “”double quotation.
* $location , $scope, $http – pass into controller.
* ngRoute –pass into module
* <form class=”” novalidate>--- it says html5 that don’t go for validation as angular is going to validate the form.
* Filter is used to filter the data as per the condition.

e.g: <input type=”” ng-model=”search”>  
<ul><li ng-repeat=”arr in myarr” |orderBy: ‘name’ | filter:search”>

* Filter can be used with directive and angular expression({{}})
* Filter used for input element and orderBy used for select element
* Display as per your individual column value then …..ng-model=”search.name” is for search as per typing name value.
* To display header use header tag e.g:<header ng-include=”’header.html’”>
* Main tag with ng-view tells the current page is a main page.
* # is a path symbol in angular. E.g localhost/#/home
* $http service is used to read the json data .
* While creating custom tag use –restrict(where to use),scope(here define all the attributes used in destination page defined in templateUrl ),templateUrl,
* $location –defines where to send the request/response using $locationProvider.
* $locationProvider—used to send the req/res without ‘/#’(by making html5mode to true while calling .config. Then in home.html page use <head><base href =”/”/>..which says the base url starts with /home instead /#/home

Methods:

push (): this method is used to inert an element to an array.  
slice(): Used to get all elements from an array and traverse it.  
parseInt(): convert string into int.  
console.log(angular.toJson($scope.myarr));---shows myarr elements in the console.  
  
$http.get(‘data/jsondata.json’).success(function(data){

$scope.myarr=data;})---read the json data and assign to my array myarr.

Mymodule.filter is used to create custom filter.

Mymodule.config is used to set all the configuration detail before register the controller with module.  
mymodule.controller –register controller  
mymodule.directive—to define our own custom tag.

transclude = true – to display contents,paragraph…etc inside our own custom tag into view page.  
replace=true—used to change the custom tag into html based tag while inspecting.(by inspecting)  
templateUrl: used in module which defines the destination page , to which need to send the response.

Classes:

<div class=”content”>-- form looks better format, look and feel.  
input type=”text”---for text  
button and ng-click –for botton  
select ng-model and option value—for select option   
input type=”number”---for number  
Input type=”checkbox” ng-model-- for checkbox

Div class=”remove” is used to open a remove option e.g X

*Directives:*

**ng-view:**

loads the related view pages and make it available to the home page.

e.g:

<main ng-view>-- load all related pages and make it available here. And behave this page as main page.

**ng-App:**

This is used to define the module we are working for and the page behaves like as an angular page, for each home page there will be only one ng-app. This is used in html tag.

**ng-controller:**

The directive define all the data will handle by the defined controller. Outside the controller data will not available.

**ng-init:**

andl The directive used to initialize the data .

e.g: <body ng-init =”numbers=[1,2,4,7]”>

**ng-model (used with input,select and textarea)**

This tag is used to receive the user input data and hold it and pass to model obj defined in module class.

So when ever changes happened at UI , the same reflecting at model.

**ng-repeat:(can be use with directive , li …)**

used to traverse through an array.  
e.g: <div ng-repeat obj in myArr>

**ng-include:**

used to include another page into my current page , Here the outside html page should be inside single quotation inside double quotation.

e.g:

<ng-include src=”’test.html’”> instead of

<header ng-include =”’test.html’”> --- this defines a header.

**ng-required:**

This is used to defined whether the validation is required for the corresponding form field .

e.g: <input type=”text” ng-required =true>

**ng-disabled:**

Will disable the form as per the condition. here if input field is invalid the submit button will disabled.

<form name=”contactform”>  
e.g: <input type=”submit” value=”submit” ng-disabled=”contactform.$invalid”>

**ng-show:**

Show the content as per the condition.

e.g:<ul><li ng-repeat=”arr in myarr” | ng-show=”arr.available” >

the available is true in my defined array myarr. That’s why shows all the elements whose available is true from an array.

**ng-hide:**

it works reverse to ng-show means , display all elements as per the condition.

**ng-click:**

Used to fire a click event . means tell angular what to do once the click is happened.

e.g:

**ng-submit:**

It is used to submit new records or forms into an array / DB…or submit the form. mean

e.g: <form ng-submit=”add()” when the form is submitted the add () method called up.

**ng-source or ng-src:**

Defines the image from a particular location.

e.g: <img ng-src=”image\das.png”>

**ng-view and ngRoute :**

ng view is used to behave as this page is a view page and get all other view/dependency pages to the current home page using ngRoute.

**FormValidation Tags:**

**Ng-pristine:** when form/input not used yet, means no text has been typed.

**Ng-dirty:** when form/input has been used, means some text has been typed.

**Ng-touched:** we are just touching the form/input field

**Ng-untouched:** we are not yet touched the form/input field

**Ng-valid:** when a form/input field has valid data.

**Ng-invalid :** when a form/input field has not valid data.

limtTo field is used to define the limit to display

e.g

<tr ng-repeat="tech in technologies | limitTo:rowlimit:1">

select ng-model="Asc"

input--text , number

button-

HTML TAGS:

<p>--paragraph

<div>--Its defines a division or section where controller scope will be there.

<ul>--Unordered list

<li>--Defines a list

<h1>--Heading in larger size, h4 is small

<span>--used to apply styles for inline elements.