* What is AngularJs
* a client side js framework.
* executed only by the browser.
* **Why AngularJs?**
* Rich Internet Application
* Single Page Application
* MVC Pattern
* Cross Browser Compatibility
* Open Source & Free.
* Disadvantages
* No security for source
* JavaScript must be enabled to work.

Concepts/Module

* **Data binding**

**1.Two Way Data Binding**

* keeps the model and view in sync at all the times
* a change in the model updates the view
* a change in view updates the model.
* Binding expression updates the view when the model changes-{{message}}
* Binding expression updates the model when the view changes ng-model
* **ng-model**
  + input
  + select
  + text area
* **Directives**

* Extends the HTML element attributes.
* Mostly start with ng-
* Contains Many Built-in directives.
* Able to create and customize directives
* also start with x-,data-,:,\_.

* **built In Directives**
* **ng-model** 
  + Use to 2Way data binding.
* **ng-bind** 
  + through ng-bind also can display the expression value in html control.<span >
* **ng-repeat**
* mostly used in arrays, or collection of elements
* **ng-read Only**
  + cannot edit the element ,can read only.
* **ng-disabled** 
  + can fix the html element. Also based on the expression result.
* **ng-if**
  + to take the element or Recreate it based on the expression.

* **Controllers**

* It is a JavaScript object.
* Contains properties and functions
* Application data's are controlled by controllers
* The job of the controller is to build a model for the view to display
* $scope is used as a parameter ,refers the module controlled by the controller.
* ng-controller directive is used
* Controllers can be stored as a external files
* can define many controllers

**var my Controller = function ($scope){**

**$scope. Message="Angular js";}**

* **Model**

* One of the component of mvc Architecture
* Managing the data used in application
* Used to Handle user interactions such as clicking
* $scope object is used by the model, which contain data and function’s.
* ng-model is used for data binding.connecting model in view.
* **Modules**
* Angular js identified the file where the module created.
* It is a main method for angular JS application.
* container for different parts of application.
* Used to define
  + Functionalities
  + add controllers
  + filters
  + services
  + directives to the applications.
* Easy to reduce the code in different applications.
* Module is attached to the controller to define the logical boundary for the controller.
* It is easily testable and maintainable component.
* Used to organize application easily.

**Module Types.**

* Application module
  + ng-app used to specify the module.

<script>

var mod=angular. Module("sample",[]); []-it does not dependent .

mod.servive('service name’, function)

mod. Factory('factory name’, function)

mod. Provider('provider name’, function)

</script>

* **Controller module**
  + ng-controller is used to specify.

**<script>**

**var mod=angular. Module("sample",[]);**

**mod. Controller('controller name’, function)**

**</script>**

* **Expression**
* Angular js evaluates expression and bind the result with
* Angular JS binds data to HTML using **Expressions**.
* HTML similar to java script.
* Contains Literals,Operators,Variables.
* {{Expression}}, ng-bind=expression.
* Numbers,Strings,Objects,Arrays,are used.
* ng-app,ng-app =""attribute is must to know AngularJs,
* <body><div>
* ng-init --->Used to initialize the variables.

* **AngularJs vs Js Expressions**

* Cannot contain loops,conditions,exceptions

ex-while loop,for loop,if else.

* Angular JS support filters, JS not
* Cannot contain functions
* Cannot contain comma
* **Sorting Data in Angular Js**

**Use order By Filter**

* {{orderBy\_expression | order By : expression : reverse}}

**Example:ng-repeat="employee in employees |** order by: ‘salary’: false"

* To sort in ascending order, set reverse to false.
* To sort in descending order, set reverse to true.
* also use + and - to sort in ascending and descending order respectively.

**Example:ng-repeat="employee in employees |** **order by:'+salary'"**

* **Filters**
* Filters can do three different things.
* Format
* Sort
* Filter data
* Filters can be used with the binding expression or a directive.
* To Apply using pipe(|) character

**{{expression | filterName:parameter}}**

**1.Filters for formatting data.**

* lowercase -> formates all character lower.
* uppercase -> formates all character uppercase.
* number -> a number as text.
* currency ->$ is default, formates a number as a currency.

custom currency and decimal places can be specified.

* date ->for date to string based on the request format.

**2.limit to filter**

* limit the number of rows or character in a string.

**{{expression | limiTo:limit:begin}}**

* **Scope**
  + Scope is an Build in object
  + It contains data and function
  + Used to bind view (HTML)part /and controller part(JavaScript)
  + View can display $scope data using an expression ,
  + ng-model,ng-bind directive.

* **Scope Types**
  + $root Scope
* Global(parent for all the scope object)
* given in ng-app directive.
* and availability of data is in the full application.
* **$scope**
* child Object
* local to that HTML Element.
* **Dependency Injection**
* It is a Software design pattern.
* Components are given their dependencies no values.
* Makes dependencies configurable.(no effect on other controller)
* Make Components
  + Reusable
  + Maintainable
  + Testable.
* **Dependency Injection in AngularJs**
  + Angular JS has built in dependency Injection mechanism.
  + Allows to divide application in smaller modules.
  + Smaller modules can be Injected into each other.
  + Make the modularization easily.
* **Implemented ways**
  + Value
  + Constant
  + Factory
  + Services
  + Provider
* **ng-init**
  + allows you to evaluate an expression in the current scope.
  + In real world application you should use a controller instead of ng-init to initialize values a scope.
  + ng-init should only be used for aliasing special properties of ng-repeat directive**.(such as $Index)**
* **ng-Include**
  + It is used to embed an HTML page into another HTML page.
  + This technique is extremely useful ,when you want to reuse a specific view in multiple pages in your application.
  + The value for in-Include directive
    - Can be the name of the HTML page that you want to reuse.
      * <div ng-include=”’Employee.html’”></div>(or)
    - A property the $scope object that points to the reusable HTML page.
      * <div ng-include=”employee”></div>
* Anchor scroll Example
  + **$anchor scroll**
    - service is used to jump to a specified element on the page.
  + **$location**
    - $location service hash method appends hash fragments to the URL.
  + **$anchor Scroll()**
    - It Reads the hash fragment in the URL and jumps to that element on the page.
  + **yOffsetproperty**
    - Specifies the vertical scroll-offset.
* **Routing**
  + It is the way to navigate between different views in your application ,without reloading the entire application.
  + This is useful for creating Single Page Application.
  + User can interact with the application without having to wait for the page reload.( **ngRoute** )
  + To use
    - ngRoute.
  + A route is a mapping between URL and View.
  + To define
    - When()method of
    - $routeProvider service.
  + $routeProvider
    - Used to configure the roots.
    - Help to define what page should display when the user click.
    - It accepts either when(), or otherwise().
  + ngRoute
    - must be added as a dependency in application module.
  + <div ng-view></div>
    - Used to render the content of the current page.
* **When()**
  + Define a route that matches a specific path.
* **Otherwise()**
  + Define a route that matches all paths that are not matched by any other routes.
* **templateUrl**
  + define the URL of the template that should be loaded when the route matches.
* **Angular JS HTML DOM**
* some directives can be used to bind application data to attributes of HTML DOM elements.
* **ng-disabled**
  + disables the given control.
* **ng-show**
  + shows the given control.
* **ng-hide**
  + hides the given control
* **ng-click**
  + Represents the click event.