AngularJS 1.X

This is simple framework built on JavaScript to create SPAs

Helloworld Program

We need to include a angular lib to get the features of angular.
By add the angular lib, we got access to angular object

```
<html ng-app="">
```
→ starting point of angular app

```
<body>
    <input type="text" ng-model="name">value
```
→ directive takes the value from field & displays where you mention attribute

```
    <div> Hello {{name}} </div>
</body>
</html>
```

The attributs starts with ng-* are directives of angulars

```
<div ng-bind="name"> .... </div>
```

Introduction to directives

docs.angularjs.org

At a high level, directive are markers on a Dom element that tell AngularJS's HTML Compiler to attach a specified behaviour to that Dom element or even transform the Dom element and its children.

data-ng:model
x-ng-model
ng-model
} same result only writing method is different

Introduction to expressions

Expressions are nothing but the curly braces "{{ }}" we use to output our ng-model. Inside the braces we can directly do arithematic operations directly

10 * 10 = {{10*10}}

ng-init - It is used initisalize a intial value to a perti variable, array, object.

```
<div ng-init = "language = 'AngularJS' ">
    {{ language }} // AngularJS
</div>

<input type="text" ng-model="language" /> // AngularJS
```

ng-repeat

→ It is another directive, which, we use it for looping a data.

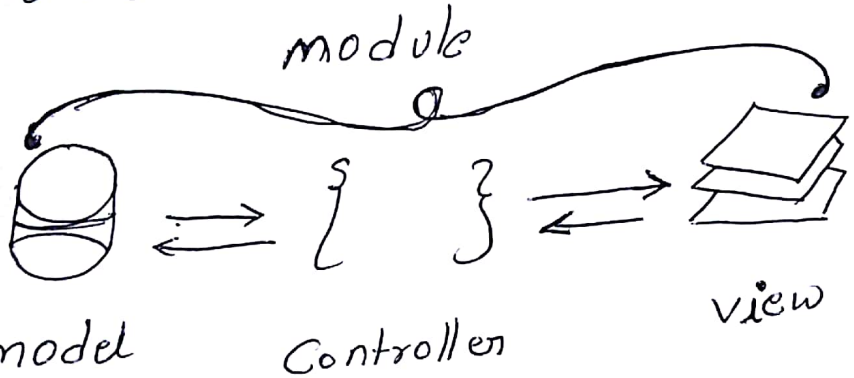
```
<div ng-init = "myFavLan = [{name: 'PHP', extention: '.PHP'},----]">

<p ng-repeat = "language in myFavLan" >
    Name : {{ language.name }} <br>
Extention : {{ language.extention }}
</p>
</div>
```

Introduction to MVC – model - view - Controller

model - How the applications behaves with data

view - ~~Controller~~ whatever is visible to the user

Controllers - The coding which would act between model and view.



we can create serease of module & make them to handle specfic part of your application

Introduction to Controller

Controller are nothing more than traditional JavaScript objects created with angularJS.

It actually control data of your application you have written in JavaScript with angularJS

```
<div ng-controller="languages">
select your Favoorite language: <button ng-click="PHP()">PHP</button> ---
- - - - <button>Java</button>

<P> You have selected : {{ myFavLanguage }} </P>

<SCRIPT>
Var app = angular.module("app", []);

app.Controller("languages", function ($scope) {
        $scope.myFavlanguage = 'None';
});

$scope.PHP = function() {
        $scope.myFavlanguage="PHP";
}

</SCRIPT>
```

Introduction to Fillers

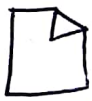
Fillers are nothing but simply a way of formating your output nicely on the screen.

```
<body ng-app>
  Your Name : <input type="text" ng-model="name"><br>
    {{ 'welcome' +name | uppercase }}
                       | currency : 'Rs.' }}
                       | number : 5}} :0
```

ng-include

We can embed Html pages within a HTML Page using ng-include directive.

```
<ng-include src="" ></ng-include>

<body ng-app>
  <div ng-include = "'Helloworld.html'" > </div>

</body>
```



Helloworld.ht

ng-routing

Routing is nothing but a way to controlle URS Contents in your website.

```
<body ng-app="app">

    <div ng-view></div>

</body>

<script>
var app = angular.module("app", ['ngRoute']);

app.config(function($routeProvider) {
    $routeProvider
    .when('/', {
        template : 'welcome user!'
    })
    .when('/anotherPage', {
        template : 'welcome another user'
    })
    .otherwise({
        redirectTo : '/'
    });
});
```

Consolen

↓

document.location.hash // "#/"

↓

document.location.hash = '!#/anothenPage';

"//!#/anothenPage"

more

```
$routeProvider
  .when('/', {
      templateUrl: 'page.html'
  })
  .when('/helloUser', {
      templateUrl: 'hello.html'
  })
  .otherwise({
      redirectTo: '/'
  });
```

Conditional routing

Create a login.html inside that create a form

```
<div ng-controller="myCtrl">
 <form action="/" id="myLogin">
Username <input type="text" name="username" id="username"><br>
Password <input type="text" name="password" id="password"><br>
    <button type="button" ng-click="submit()">Login</button>
 </form>
</div>

<script

    $routeProvider
    .when('/', {
        templateUrl : 'login.html'
    })
    .when('/dashboard', {
        templateUrl : 'dashboard.html'
    })
    .otherwise({
        redirectTo : '/'
    });
```

ng-model

ng-model

```
app.controller('loginCtrl', function($scope, $location) {
    $scope.submit = function() {
        var uname = $scope.username;
        var password = $scope.password;
    if ($scope.username == 'admin' && $scope.password == 'admin') {
        $location.path('/dashboard');
    } else {
        alert('wrong stuff')
    }
};
});
```

```
. when('/dashboard', {
        resolve: {
            "check": function($location, $rootScope) {
                if(!$rootScope.loggedIn) {
                    $location.path('/');
                }
            }
        },
        templateUrl: 'dashboard.html'
});
```

```
app.Controller('loginCtrl', function ($scope, $location, $rootScope) {
    $scope.submit = function() {

if ($scope.username == 'admin' && $scope.Password == "admin") {
        $rootScope.loggedIn = true;
        $location.path('/dashboard');
    } else {
        alert('wrong stuff');
    }
```

http service

it's nothing but retrevirng data from server and display in view.

```
<div ng-controller = "myCtrl" >
    <h2> Names and Age of Programmers In the Local Area :</h
    <ul>
        <li ng-repeat = "person in persons">
            {{person.Name + ' : ' + Person.Age }}
        </li>
    </ul>
</div>
```

[record ! {
    { name :
      age :
    },
    { name :
      age :
    }

```
var app = angular.module("mainApp", []);

app.controller('people', function ($scope, $http) {

    $http.get("file location").success(function(response){

        $scope.persons = response.records;

    });

});
```

```
vax filex = {
    "records": [

        {
        "name": "mehul",
        "Age": "18",
        "Fav_color": "orange"
        },

        {name:
         age:
         Fav_Color:
        },

        {name:
         age:
         Fav_Color:
        }
    ]
```

creating simple Search Box

Let create a simple search box using ng-model

```
<input type = "text" ng-model = "search_Box">
```

now write filter in the ng-reapet using pipe & write searchBo
that's it filter is Created.

```
<li' ng-repeat = "person in Persons | filter : searchBox" >
```

Advance search Box

```
<div ng-controller = "People">
  <ul> <h2> Names and Age of Programmers in the local Area; </h2>
  <li ng-repeat = "Person in Persons | filter ; globalSearch">
  {{ "Name: " + Person.Name + " Age : " + Person.Age + " Fav Color:"
        + Person.Fav_Color }}
  </li>
</ul>
</div>
```

```
<div id="searchboxes">
  Global search Filter : <input type="text" ng-model="globalSearch.$">
  Name Search Filter: <input type="text" ng-model="globalSearch.Name">
  Age search filter: <input type="text" ng-model="globalSearch.Age">
  Fav Color            :       "           "           "  globalSearch.FavColor)

</div>
```

Passing Data between different scopes

```
<input type="text" ng-model="message">
```

ng automatically Creates local scope inside each model.

```
<span ng-repeat = "i in [1,2]">
    {{i}}
</span>
    {{i}} // no scope
```

```
<input type = "text" ng-model="message.Property">
<p ng-repeat = "i in [1]">
    <input type="text" ng-model="message">
    you wrote : {{message}}Property
</p>

[outer] you wrote : {{message}}Property
```

if you just write message it becomes local scope instead user.Property which pass message as a global & Property

Range based loops

script

```
var app = angular.module("APP", []);
app.controller('app', function($scope) {
    var range = 10;
    var myRange = [];
    for (i=0; i<range; i++) {
      myRange.push(i);
    }
    $scope.myRange = myRange;
});
```

short cut →

```
var range = new Array(100);
$scope.myRange = range
```

```
<body ng-app="APP">
  <div ng-controller="app">
    <li ng-repeat="i in myRange">
      {{i}} <br>
    </li>
  </div>
</body>
```

short cut →

```
i in range track by $index
```

$Sope.$watch()

Its used to watch over a variable ourseleaves & get a hold of it in JavaScript

For example the input field i want to know how many tymes it is modified to do that we can use watch

```
<input type='text' ng-model="myText">
{{myText}}

<br>
You update text box {{counter}} times

Var app= angular.module("app",[]);
app.Controller("myctrl", function($scope){
    $scope.counter=-1;
    $scope.$watch('myText', function(newValue, oldValue){
        console.log("newValue:"+newValue);
        Console.log("Old value:"+oldValue);
    $scope.counter++;
    }
```

$scope.$digest

Just like $watch over the variable, and $digest also create a life cycle of digist iterate over all the watches

HTML

```
<span> {{myRandomNumber}} </span>
<br>
<input type="button" value="Create New Random number">
```

```
Var app = angular.module('mainApp', []);
app.controller('app', function($scope) {
    $scope.myRandomNumber = Math.random();
    document.querrySelector('input').addEventListener('click', function(
        Console.log("button clicked");
        $scope.myRandomNumber = Math.random();
    }, false);
    $scope.$digest();
```

Services

A service in Angular is simply an object that Provide some sort of service that can be revsed with in an angular application.

services encapsulate reusable logic that does not belong any where else

Directives, filters, views, models & Controllers)

what are the benfits of using services

- Reusability
- Dependency injection
- Testability

```
<button ng-click="generateRandom()">
  Generate num </button>
  br
  {{ randomNumber }}
```

```
var app = angular.module("mainApp", [ ]);

app.Service('random', function(
  var num = Math.floor(Math.random() * 10);
  this.generate = function() {
      return num;
  };

app.Controller('app', function($scope, random) {
  $scope.generateRandom = function() {
    $scope.randomNumber = random.generate();
  };
```

Factories

The services & factories a pretty much similar ones.

```
app.factory('random', function() {
    var randomObject = {};
    var num = Math.floor(Math.random()*10);
    randomObject.generate = function() {
        return num;
    };
    return randomObject;
});
```

Providers

They are basically head of services, factories can be done in providers.

{{ greatmessage }}

```
app.controller('app', function($scope, date) {
    $scope.greatmessage = date.showDate();
});
```

```
var app = angular.module('mainApp', []);
app.provider('date', function() {
    return {
        $get: function() {
            return {
                showDate: function() {
                    var date = new Date();
                    return date.getHours();
                }
            }
        }
    }
});
```

ngShow Directive

```
<body ng-app="mainApp" ng-controller="app">

    <button ng-click="toggle()">Toggle message</button>

    <div id="message" ng-show="message=='show'">
    This is my message 1 </div>
    <div id="message" ng-show="show=='msg2'">
    This is my message 2 </div>

var app = angular.module('mainApp',[]);
    app.controller('app', function($scope) {
        $scope.show = 'msg1';
        $scope.toggle = function() {
            $scope.show = $scope.show = 'msg1' ? 'msg2' : 'msg1';
        }
    });

</body>
```