Angular JS

AngularJS is a **JavaScript framework**. It can be added to an HTML page with a <script> tag.

AngularJS extends HTML attributes with **Directives**, and binds data to HTML with **Expressions**.

AngularJS is a JavaScript framework written in JavaScript.

AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

## AngularJS Extends HTML

AngularJS extends HTML with **ng-directives**.

The **ng-app** directive defines an AngularJS application.

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.

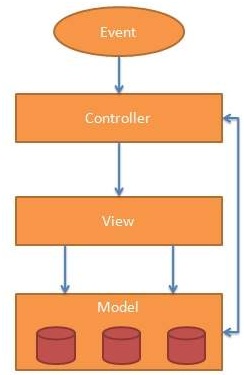
The **ng-bind** directive binds application data to the HTML view.

**Following are the advantages of AngularJS over other JavaScript frameworks:**

* **Dependency Injection:** Dependency Injection specifies a design pattern in which components are given their dependencies instead of hard coding them within the component.
* **Two way data binding:** AngularJS creates a two way data-binding between the select element and the orderProp model. orderProp is then used as the input for the orderBy filter.
* **Testing:** Angular JS is designed in a way that we can test right from the start. So, it is very easy to test any of its components through unit testing and end-to-end testing.
* **Model View Controller:** In Angular JS, it is very easy to develop application in a clean MVC way. You just have to split your application code into MVC components i.e. Model, View and the Controller.
* Directives, filters, modules, routes etc.

# AngularJS MVC Architecture

MVC stands for Model View Controller. It is a software design pattern for developing web applications. It is very popular because it isolates the application logic from the user interface layer and supports separation of concerns.



The MVC pattern is made up of the following three parts:

1. **Model:** It is responsible for managing application data. It responds to the requests from view and to the instructions from controller to update itself.
2. **View:** It is responsible for displaying all data or only a portion of data to the users. It also specifies the data in a particular format triggered by the controller's decision to present the data. They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology.
3. **Controller:** It is responsible to control the relation between models and views. It responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

EX:- <html>  
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>  
<body>  
  
<div ng-app="">  
  <p>Name: <input type="text" ng-model="name"></p>  
  <p ng-bind="name"></p>  
</div>  
  
</body>  
</html>

AngularJS starts automatically when the web page has loaded.

The **ng-app** directive tells AngularJS that the <div> element is the "owner" of an AngularJS **application**.

The **ng-model** directive binds the value of the input field to the application variable **name**.

The **ng-bind** directive binds the content of the <p> element to the application variable **name**.

Ex2:-

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="">

<p>Input something in the input box:</p>

<p>Name : <input type="text" ng-model="name" placeholder="Enter name here"></p>

<h1>Hello {{name}}</h1>

</div>

</body>

</html>

**Lab Programs**

## 1. Write a AngularJS Program to print your details

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="">

<p>Input something in the input box:</p>

<p>Name : <input type="text" ng-model="name" placeholder="Enter name here"></p>

<h1>Hello {{name}}</h1>

</div>

</body>

</html>

## 2.Write a angularJS program to binding data and perform Expressions using ng-bind

## Viva Questions

## Data binding is a very useful and powerful feature used in software development technologies. It acts as a bridge between the view and business logic of the application

## One-Way Data Binding

The one-way data binding is an approach where a value is taken from the data model and inserted into an HTML element. There is no way to update model from view. It is used in classical template systems. These systems bind data in only one direction.

## Two-Way Data Binding

Data-binding in Angular apps is the automatic synchronization of data between the model and view components.Data binding lets you treat the model as the single-source-of-truth in your application. The view is a projection of the model at all times. If the model is changed, the view reflects the change and vice versa.

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<body>

<div ng-app="" ng-init="firstName='Ajeet'">

<p>Input something in the input box:</p>

<p>Name: <input type="text" ng-model="firstName"></p>

<p>You wrote: {{ firstName }}</p>

</div>

</body>

</html>

In the above example, the {{ firstName }} expression is an AngularJS data binding expression. Data binding in AngularJS binds AngularJS expressions with AngularJS data.

{{ firstName }} is bound with ng-model="firstName".

Input something in the input box:

Output:-Name: 

You wrote: {{ firstName }}

Ex2; <!DOCTYPE html>

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<body>

<div data-ng-app="" data-ng-init="quantity=1;price=20">

<h2>Cost Calculator</h2>

Quantity: <input type="number" ng-model="quantity">

Price: <input type="number" ng-model="price">

<p><b>Total in rupees:</b> {{quantity \* price}}</p>

</div>

</body>

</html>

**Cost Calculator**

Quantity: Price:

**Total in rupees:** 20

## AngularJS Expressions

AngularJS expressions can be written inside double braces: {{ expression }}.

AngularJS expressions can also be written inside a directive: ng-bind="expression".

AngularJS will resolve the expression, and return the result exactly where the expression is written.

**AngularJS expressions** are much like **JavaScript expressions:** They can contain literals, operators, and variables.

Example {{ 5 + 5 }} or {{ firstName + " " + lastName }}

<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div ng-app="">

<p>My first expression: {{ 5 + 5 }}</p>

</div>

</body>

</html>

Ex2;<html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<p>Without the ng-app directive, HTML will display the expression as it is, without solving it.</p>

<div>

<p>My first expression: {{ 5 + 5 }}</p>

</div>

</body>

</html>

7. <html>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<p>Try to change the names.</p>

<div ng-app="myApp" ng-controller="myCtrl">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{firstName + " " + lastName}}

</div>

<script>

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

app.controller('myCtrl', function($scope) {

$scope.firstName= "John";

$scope.lastName= "Doe";

});

</script>

</body>

3.Write a angularJS program to using Angular directives

# Viva questions

# AngularJS Directives

AngularJS facilitates you to extend HTML with new attributes. These attributes are called directives.

There is a set of built-in directive in AngularJS which offers functionality to your applications. You can also define your own directives.

Directives are special attributes starting with ng- prefix. Following are the most common directives:

* ng-app: This directive starts an AngularJS Application.
* ng-init: This directive initializes application data.
* ng-model: This directive defines the model that is variable to be used in AngularJS.
* ng-repeat: This directive repeats html elements for each item in a collection.

## ng-app directive

ng-app directive defines the root element. It starts an AngularJS Application and automatically initializes or bootstraps the application when web page containing AngularJS Application is loaded. It is also used to load various AngularJS modules in AngularJS Application.

**See this example:**

In following example, we've defined a default AngularJS application using ng-app attribute of a div element.

<div ng-app = "">

1. ...
2. </div>

**ng-init directive**

ng-init directive initializes an AngularJS Application data. It defines the initial values for an AngularJS application.

In following example, we'll initialize an array of countries. We're using JSON syntax to define array of countries.

1. <div ng-app = "" ng-init = "countries = [{locale:'en-IND',name:'India'}, {locale:'en-PAK',name:'Pakistan'}, {locale:'en-AUS',name:'Australia'}]">
2. ...
3. </div>

**ng-model directive:**

ng-model directive defines the model/variable to be used in AngularJS Application.

In following example, we've defined a model named "name".

1. <div ng-app = "">
2. ...
3. <p>Enter your Name: <input type = "text" ng-model = "name"></p>
4. </div>

**ng-repeat directive**

ng-repeat directive repeats html elements for each item in a collection. In following example, we've iterated over array of countries.

1. <div ng-app = "">
2. ...
3. <p>List of Countries with locale:</p>
5. <ol>
6. <li ng-repeat = "country in countries">
7. {{ 'Country: ' + country.name + ', Locale: ' + country.locale }}
8. </li>
9. </ol>

Ex:- <html>

<head>

<title>AngularJS Directives</title>

</head>

<body>

<h1>Sample Application</h1>

<div ng-app = "" ng-init = "countries = [{locale:'en-IND',name:'India'}, {locale:'en-PAK',name:'Pakistan'}, {locale:'en-AUS',name:'Australia'}]">

<p>Enter your Name: <input type = "text" ng-model = "name"></p>

<p>Hello <span ng-bind = "name"></span>!</p>

<p>List of Countries with locale:</p>

<ol>

<li ng-repeat = "country in countries">

{{ 'Country: ' + country.name + ', Locale: ' + country.locale }}

</li>

</ol>

</div>

<script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

</body>

</html>

OUTPUT:-

# Sample Application

Enter your Name: 

Hello !

List of Countries with locale:

1. Country: India, Locale: en-IND
2. Country: Pakistan, Locale: en-PAK
3. Country: Australia, Locale: en-AUS