AngularJS

What is AngularJS

- Angular JS is an open source JavaScript framework that is used to build web applications. It can be freely used, changed and shared by anyone.
- Angular Js is developed by Google.
- It is an excellent framework for building single phase applications and line of business applications.
- AngularJS is entirely based on HTML and JavaScript,so there is no need to learn another syntax or language.
- AngularJS is also called just "Angular".
- AngularJS changes static HTML to dynamic HTML. Itbextends the ability of HTML by adding built-in attributes and components and also provides an ability to create custom attributes using simple JavaScript.
- AngularJS can be used to create Single Page Applications.
- AngularJS website - https://angularjs.org

Features of AngularJS:

(1) Data-binding: It is the automatic synchronization of data between model and view components.
(2) Scope: These are objects that refer to the model. They act as a glue between controller and view.
(3) Controller: These are JavaScript functions bound to a particular scope.
(4) Services: AngularJS comes with several built-in services such as $http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
(5) Filters: These select a subset of items from an array and returns a new array.
(6) Routing: It is concept of switching views.
(7) Model View Whatever: MVW is a design pattern for dividing an application into different parts called Model, View, and Controller, each with distinct responsibilities. Angular JS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-ViewView-Model). The Angular JS team refers it humorously as Model View Whatever.

Advantages of AngularJS:

The advantages of AngularJS are :

(1) It provides the capability to create Single Page Application in a very clean and maintainable way.
(2) It provides data binding capability to HTML. Thus, it gives user a rich and responsive experience.
(3) AngularJS code is unit testable.
(4) AngularJS uses dependency injection and make use of separation of concerns.
(5) AngularJS provides reusable components.
(6) With AngularJS, the developers can achieve more functionality with short code.

Disadvantages of AngularJS:

Though AngularJS comes with a lot of merits, here are some points of concern :

(1) Not Secure: Being JavaScript only framework, application written in Angular JS are not safe. Serverside authentication and authorization is must to keep an application secure.
(2) Not degradable: If the user of your application disables JavaScript, then nothing would be visible, except the basic page.

AngularJS Environment Setup

AngularJS Module

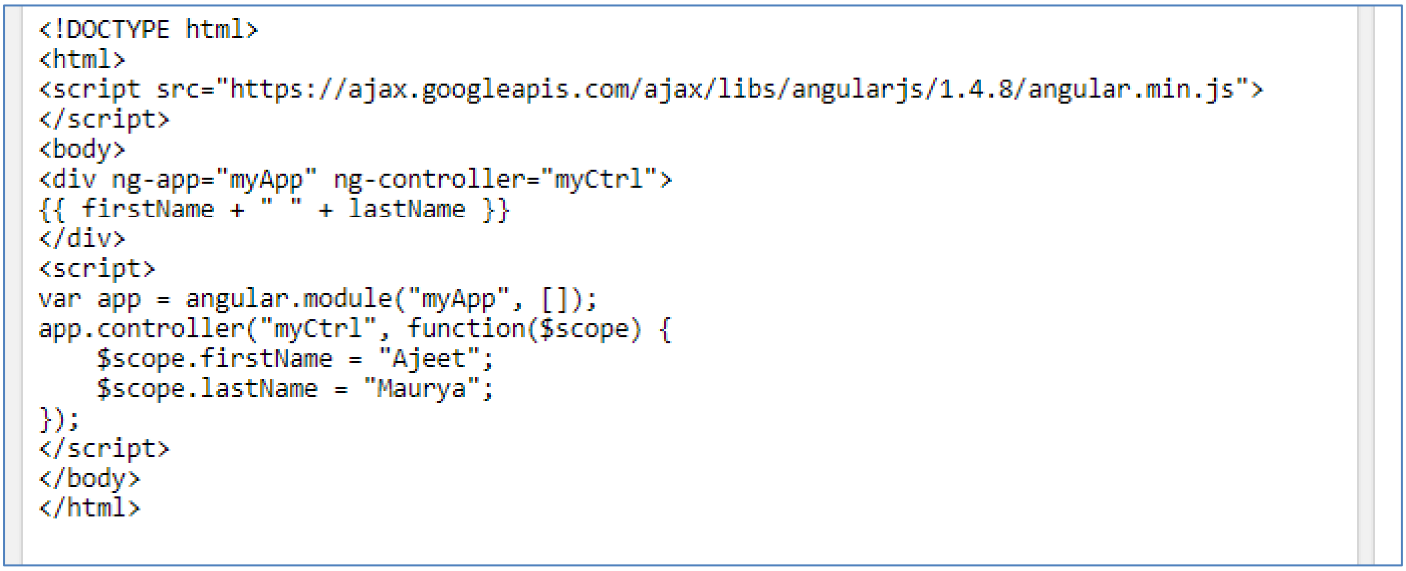
- A module in AngularJS is a container of the different parts of an application such as controller, service, filters, directives, factories etc.
- It supports separation of concern using modules.
- Modules are used to separate logic such as services, controllers, application etc. from the code and maintain the code clean.
- We define modules in separate js files. A module is used as a Main() method.

How to create a module

The angular object's module() method is used to create a module. It is also called AngularJS function angular.module

1. `<div ng-app="myApp">…</div>`
2. `<script>`
3. `var app = angular.module("myApp", []);`
4. `</script>`

Ex



```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body>
<div ng-app="myApp" ng-controller="myCtrl">
{{ firstName + " " + lastName }}
</div>
<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
    $scope.firstName = "Ajeet";
    $scope.lastName = "Maurya";
});
</script>
</body>
</html>
```

Directive in AngularJS: (ex)

- Directives are markers on the DOM element which tell AngularJS to attach a specified behaviour to that DOM element or even transform the DOM element with its children.
- Simple AngularJS allows extending HTML with new attributes called Directives. AngularJS has a set of built-in directives which offers functionality to the applications.
- It also defines its own directives. A directive can be defined using some functions which are: Element name, Attribute, Class, and Comment**.**

Why use Directive in AngularJS?

It gives support to creating custom directives for different types of elements.

A directive is activated when the same element or matching element is there in front.

It is used to give more power to HTML by helping them with the new syntax.

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 = "">
   ...
</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.

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

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

AngularJS directives Example

Let's take an example to use all the above discussed directives:

```
<!DOCTYPE html>
<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 = "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>
</body>
</html>
```

Benefits of AngularJS Directive:

- Directives are helpful in creating repeat and independent code.
- They modularize the code by clubbing requirement-specific behavioral functions in one place. It does not create objects in the central controller and manipulate them using multiple JavaScript methods.

Expressions

- Expressions in AngularJS are used to bind application data to HTML.
- The expressions are resolved by AngularJS and the result is returned back to where the expression is written.
- The expressions in AngularJS are written in double braces: {{ expression }}.
- They behave similar to ng-bind directives: ng-bind=”expression”.

Syntax:`{{ expression }}`

AngularJS Expressions Example

```
<!DOCTYPE html>
<html>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>
<div ng-app>
<p>A simple expression example: {{ 5 + 5 }}</p>
</div>
</body>
</html>
```

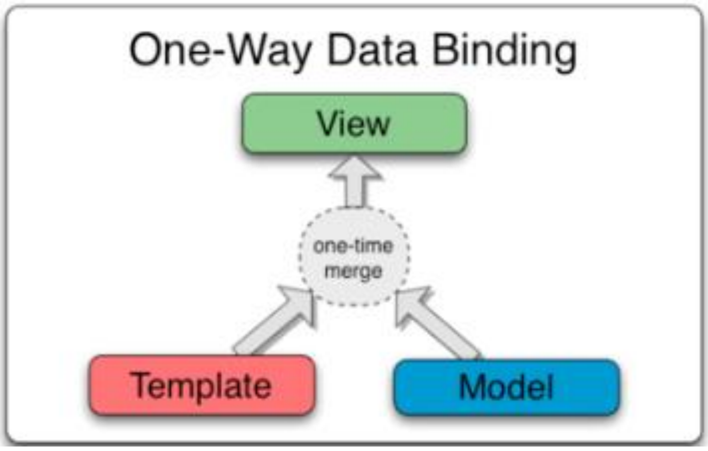
✒AngularJS Data Binding

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.

AngularJS follows Two-Way data binding model.

One-Way Data Binding

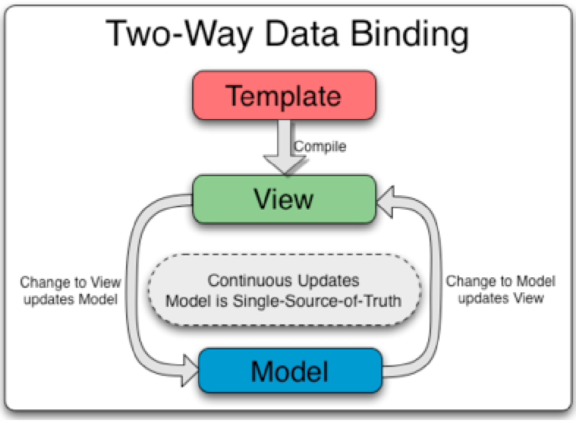
- The one-way data binding is an approach where a value is taken from the data model and inserted into an HTML element.
- There is no way to update model from view.
- It is used in classical template systems. These systems bind data in only one direction.

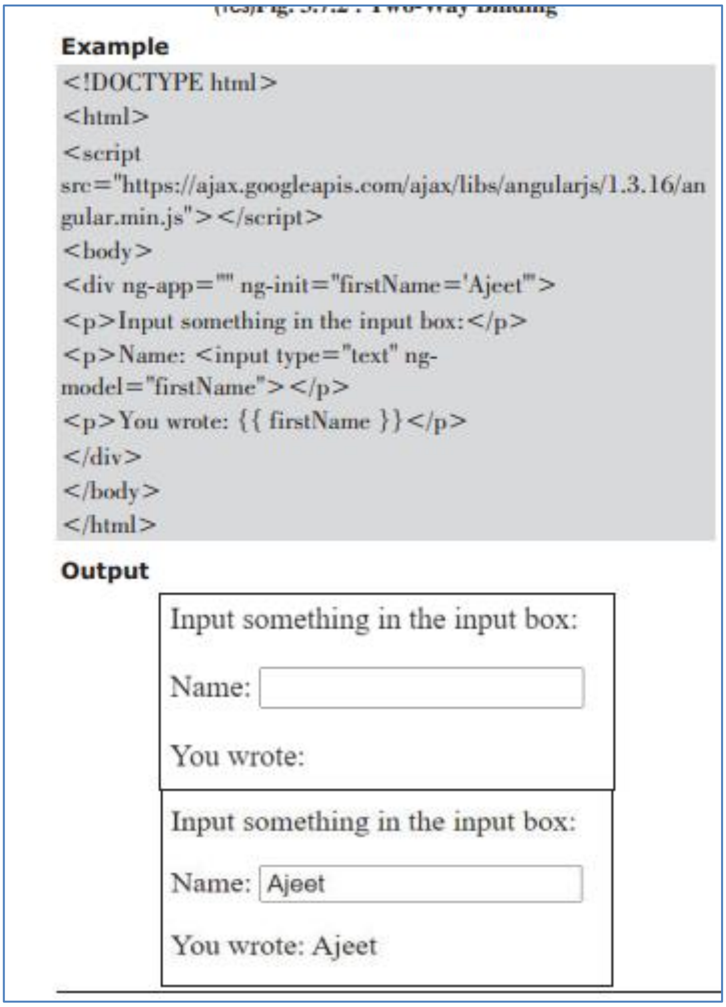


Two-Way Data Binding:

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.


Two-Way Data Binding
Template
Compile
View
Change to View updates Model
Continuous Updates Model is Single-Source-of-Truth
Change to Model updates View
Model


(ICS)Fig. 5.7.2 : Two-way Binding

Example

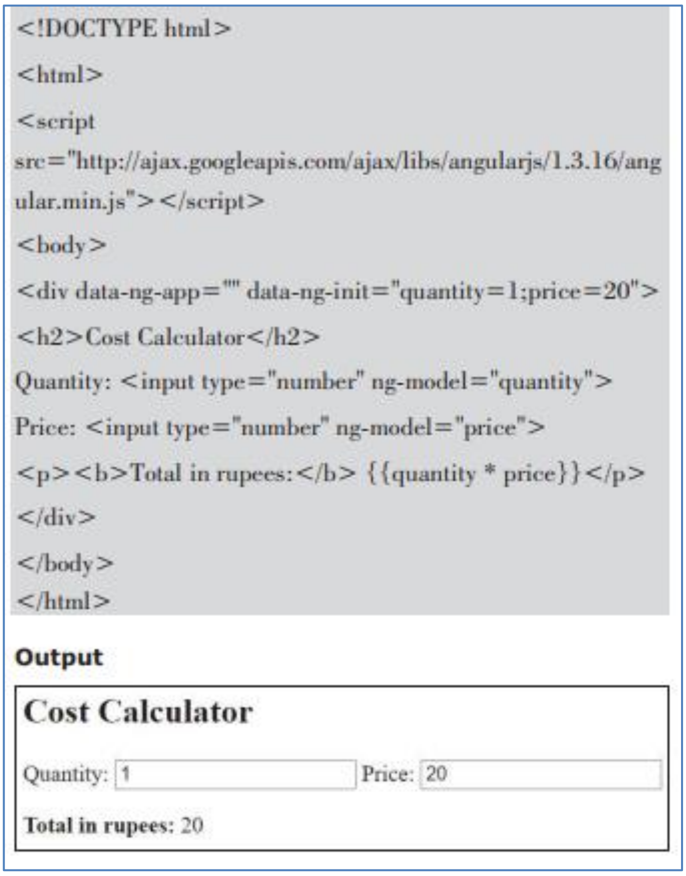
```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/an
gular.min.js"></script>
<body>
<div ng-app="" ng-init="firstName='Ajeet'">
<p>Input something in the input box:</p>
<p>Name: <input type="text" ng-
model="firstName"></p>
<p>You wrote: {{ firstName }}</p>
</div>
</body>
</html>
```

Output

```
Input something in the input box:

Name: [                    ]

You wrote:
```

```
Input something in the input box:

Name: [Ajeet               ]

You wrote: Ajeet
```

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"



```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.16/ang
ular.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>
```

Output

```
Cost Calculator

Quantity: 1                Price: 20

Total in rupees: 20
```

What Are the Differences Between Node.js and AngularJS?