**Introduction**

AngularJS is a JavaScript framework for dynamic web application. It is a powerful library of JavaScript. It can be added to an HTML page with a <script> tag. AngularJS, is an open-source web application framework which is maintained by Google.

AngularJS is quite new Technology its first version 1.0 was released in 2012. It is developed by Misko Hevery, a Google employee, he started to work on AngularJS in 2009. Its latest version is 1.2.21.

**AngularJS Directives**

The AngularJS framework can be divided into three major parts;

* ng-app : The ng-app directive initializes an AngularJS application.
* ng-model : ng-model directive binds the value of HTML controls (input, select, textarea) to application data.
* ng-bind : ng-bind directive binds the AngularJS application data to HTML tags.

**Why called AngularJS**

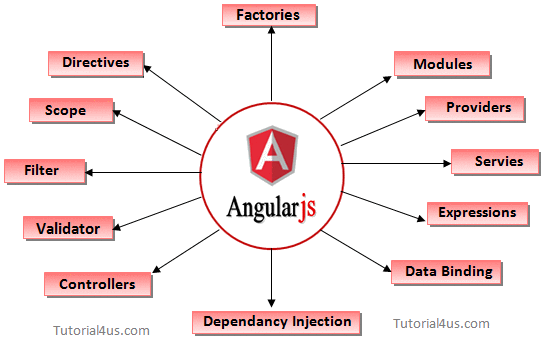
Because HTML has Angular brackets and "ng" sounds like "Angular".

**AngularJS a library, framework, plugin or a browser extension?**

AngularJS is the framework of JavaScript. AngularJS is 100% JavaScript, 100% client-side and compatible with both desktop and mobile browsers. So it's definitely not a plugin or some other native browser extension.

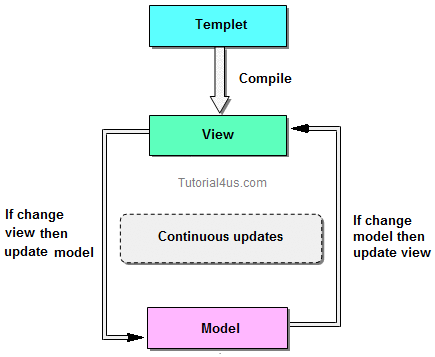
## Angularjs Features

It is framework of JavaScript and it have own event handing directives. It is open source and it is free for every one. AngularJS have a lot of new features which is given below;



## Two Way Data-Binding

It Support two way data binding features.



## MVC

AngularJS is a framework of JavaScript which work on MVC model.

## Dependency Injection

AngularJS has a built-in dependency injection subsystem that helps the developer by making the application easier to develop, understand, and test.

## Validation

AngularJS provides client side validation same like JavaScript. Using AngularJS you can create your own validation.

## Filter

Filter are mainly used for modify the data. Filters can be added to expressions and directives using a pipe (|) character.

## Directives

A directive is something that introduces new syntax. It improve the feature or functionality of html elements. Directives are markers on a DOM element which attach a special behavior to it. For example, static HTML does not know how to create and display a date picker widget. To teach HTML this new syntax we need a directive. AngularJS directives are extended HTML attributes with the prefix ng-.

## Forms

An AngularJS form is a collection of input controls like button, input elements. AngularJS has some features for binding data of HTML form input fields to the model object ($scope). You bind an input field to a model property using the ng-model directive.

**Advantage of Angularjs**

AngularJS is a framework of JavaScript. It improve the properties of Html elements. Some advantage of of angularjs are;

* **Less Code:**It help to write less code. Using angularjs developers can achieve more functionality with writing less code.
* **Testable:**It make your application testable. AngularJS code is unit testable.
* It increase the level of abstraction.
* It provides reusable components.
* Using this you can design more responsive web pages.
* It work well with other Libraries.
* It help to make and design a big applications.
* AngularJS support most of all new web browser and all mobiles OS.
* In AngularJS view are pure html code and controller are written in JavaScript to achieve business logic.

**Dis-advantage of Angularjs**

AngularJS have lot of demerits here we will discuss some important dis-advantages;

* **Not Secure:**It is a framework of JavaScript not a server side programming. Only server side programs provides security, server side authentication and authorization provides security for application.
* **Depends on User:**If any user disable JavaScript in their own browser then nothing will be apply on your application related to JavaScript and AngularJS, Only basic Html page will display.

## Angularjs Directives

A **directive** is something that introduces new syntax. It improve the feature or functionality of html elements. Directives are markers on a DOM element which attach a special behavior to it. For example, static Html does not know how to create and display a date picker widget. To teach Html this new syntax we need a directive. AngularJS directives are extended Html attributes with the prefix ng-.

Here we discuss following directives;

## ng-app

The **ng-app** directive initializes an AngularJS application. The ng-app directive also tells AngularJS that the <div> element is the "owner" of the AngularJS application. Using this directive you can tell which part of html contains Angularjs app. Below we use ng-app with <div> tag.

## Example

**<div** ng-app=""**>**

Enter text **<input** type="text" ng-model="name"**>**

**<p** ng-bind="name"**></p>**

**</div>**

## Result

Enter text 

## ng-init

The **ng-init** directive initialize application data same like variable initialization in C language, In c language you initialize int a=10;.

The ng-init directive initializes an AngularJS Application data. It is used to assign values to the variables.

## Example

**<div** ng-app="" ng-init="name='Porter'"**>** // initialize name="Porter"

Name: **<input** type="text" ng-model="name"**>**

**<p>**name: {{ name }}**</p>**

[Try it yourself](http://www.tutorial4us.com/tools/html-editor/html-editor.htm)

## ng-model

The **ng-model** directive binds the value of Html controls (input, select, textarea) to application data. The ng-model directive defines the model/variable to be used in AngularJS Application. In the following example, we define a model named name.

## Example

**<div** data-ng-app=""**>**

1st number **<input** type="number" ng-model="num1"**>**

2nd number **<input** type="number" ng-model="num2"**>**

**<p><b>**Sum:**</b>** {{num1 + num2}}**</p>**

**</div>**

[Try it yourself](http://www.tutorial4us.com/tools/html-editor/html-editor.htm)

## ng-repeat

The ng-repeat directive repeats an Html element. ng-repeat directive repeats a specific element.

## Example

**<div** data-ng-app="" data-ng-init="num=[1, 2, 3]"**>**

**<p** data-ng-repeat="x in num"**>**

{{ x }}

**</p>**

**</div>**

[Try it yourself](http://www.tutorial4us.com/tools/html-editor/html-editor.htm)

## Angularjs Controller

AngularJS applications are controlled by **Controller**. AngularJS controllers are used for control the data of AngularJS applications. AngularJS controllers are regular JavaScript Objects.

For define controller in AngularJS application we need **ng-controller** directive names

## Example

<!DOCTYPE html>

**<html>**

**<head>**

**<script** src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.2.26/angular.min.js"**></script>**

**</head>**

**<body>**

**<div** ng-app="" ng-controller="AJSController"**>**

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>**

**function** AJSController($scope) {

$scope.firstName = "Harry",

$scope.lastName = "Porter"

}

**</script>**

**</body>**

**</html>**

## Result

First Name:   
Last Name:   
  
Full Name: Harry Porter

In above example controller are use for conrtoller firstName and lastName. AngularJS will invoke personController with a $scope object.

## Expressions

AngularJS Expressions used to binds application data to HTML. AngularJS expressions are written inside double curly braces such as : {{ expression }}. Angularjs Expressions behave same like to ng-bind directives. AngularJS expressions are pure JavaScript expressions and output of the data display where they are used.

### Sample Example of AngularJS Expressions

## Result

## Calculator

First Number   
Second Number

**Sum:** 11

**Sub:** 9

**Mul:** 10

**Div:** 10

## Example

<!DOCTYPE html>

**<html** lang="en"**>**

**<div** ng-app="" ng-init="num1=10; num2=1"**>**

**<h2>**Calculator**</h2>**

First Number **<input** type="number" ng-model="num1"**><br>**

Second Number **<input** type="number" ng-model="num2"**>**

**<p><b>**Sum:**</b>** {{num1 + num2}}**</p>**

**<p><b>**Sub:**</b>** {{num1 - num2}}**</p>**

**<p><b>**Mul:**</b>** {{num1 \* num2}}**</p>**

**<p><b>**Div:**</b>** {{num1 / num2}}**</p>**

**<script** src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"**></script>**

**</body>**

**</html>**

**Note:**when you remove the ng-app directive, Html will display the expression as it is, without solving it.

### Use Expression for String

## Example

**<div** ng-app="" ng-init="firstName='Harry'; lastName='Porter';"**>**

**<p>**Full Name: {{firstName + " " + lastName }}**</p>**

### Use Expression for Array

## Example

**<div** ng-app="" ng-init="array=[10,15,20,25,30]"**>**

**<p>**Second Element is {{ array[1] }}**</p>**

**</div>**

### Example using ng-bind for String

## Example

<!DOCTYPE html>

**<html>**

**<head>**

**<script** src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.2.26/angular.min.js"**></script>**

**</head>**

**<body>**

**<div** ng-app="" ng-init="firstName='Harry'; lastName='Porter';"**>**

**<p>**Full Name: **<span** ng-bind="firstName + ' ' + lastName"**></span></p>**

**</div>**

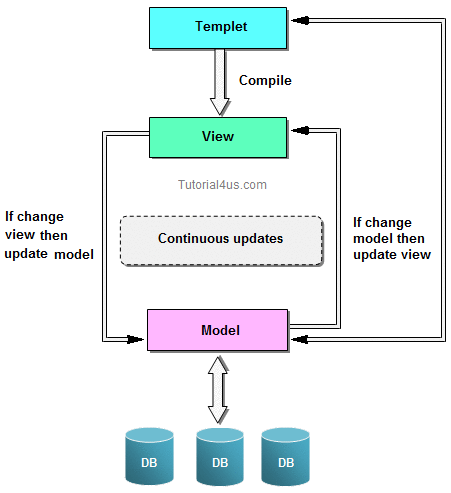
**</body>**

**</html>**

**MVC Architecture**

MVC stand for Model View Controller. Model View Controller is a software design pattern for developing web application. It given software application into three interconnected parts.

* **Model :**It is responsible for maintaining data.
* **View :**It is responsible for displaying output data to the user.
* **Controller :**It is responsible for controls the interactions between the Model and View.



**Model**

**Model**is responsible for maintaining data. Model retrieve data form database and also store data in database.

**View**

**View**It is responsible for displaying output data to the user.

**Controller**

**Controller**It is responsible for controls the interactions between the Model and View. Controllers can read data from a view, control user input, and send input data to the model.

## Html DOM

**AngularJS** has directives for binding application data to the attributes of HTML DOM elements. The following directives are used to bind application data to the attributes of HTML DOM elements;

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Name** | **Description** |
| 1 | ng-disabled | To Disables a given control of Html elements. |
| 2 | ng-show | To Shows a given control of Html elements. |
| 3 | ng-hide | To Hides a given control of Html elements. |
| 4 | ng-click | To perform AngularJS click event. |

## ng-disable Directive

**ng-disable** directive are use for binds AngularJS application data to the disabled attribute of HTML elements.

## Example

**<div** ng-app=""**>**

**<input** type="checkbox" ng-model="enableDisableButton"**>**Disable Button

**<button** ng-disabled="enableDisableButton"**>**Click Me!**</button>**

**</div>**

## Result

Disable Button  
Click Me!

## ng-show Directive

**ng-show** directive are use for show or hide Html elements.

**<div** ng-app=""**>**

**<input** type="checkbox" ng-model="hideme"**>**Show Button

**<button** ng-show="hideme"**>**Click Me!**</button>**

**</div>**

## ng-hide Directive

**ng-hide** directive are use for hide or show an Html elements.

## Example

**<div** ng-app=""**>**

**<input** type="checkbox" ng-model="hideme"**>**Hide Button

**<button** ng-hide="hideme"**>**Click Me!**</button>**

**</div>**

## ng-click Directive

**ng-click** directive are use for perform an AngularJS click event.

## Example

**<div** ng-app="" ng-controller="clickController"**>**

**<button** ng-click="count = count + 1"**>**Click Me!**</button>**

**<p>**{{ count }}**</p>**

**</div>**

**<script>**

**function** clickController($scope) {

$scope.count = 0;

}

**</script>**

## Angularjs Filters

Filter are mainly used for modify the data. Filters can be added to expressions and directives using a pipe (|) character. Following Filter are use in AngularJs;

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Name** | **Description** |
| 1 | uppercase | converts a text to upper case text. |
| 2 | lowercase | converts a text to lower case text. |
| 3 | currency | formats text in a currency format. |
| 4 | filter | filter the array to a subset of it based on provided criteria. |
| 5 | orderBy | orders the array based on provided criteria. |

## uppercase

uppercase filter are used for converting lower case text to upper case text. It is use to expression with pipe character.

## Example

**<div** ng-app="" ng-controller="personController"**>**

Enter first name:**<input** type="text" ng-model="name"**><br/>**

**<p>**The name is {{ name | uppercase }}**</p>**

**</div>**

**<script>**

**function** personController($scope) {

$scope.name="Porter"

}

**</script>**

## Example

Enter first name:

The name is PORTER

## lowercase

lowercase filter are used for converting upper case text to lower case text. It is use to expression with pipe character.

## Example

**<div** ng-app="" ng-controller="personController"**>**

Enter first name:**<input** type="text" ng-model="name"**><br/>**

**<p>**The name is {{ name | lowecase }}**</p>**

**</div>**

**<script>**

**function** personController($scope) {

$scope.name="PORTER"

}

**</script>**

## currency

Add currency filter to an expression returning number using pipe character. Below we print total cost of items.

## Example

**<div** ng-app="" ng-controller="personController"**>**

**<input** type="number" ng-model="quantity"**><br** **/>**

**<input** type="number" ng-model="price"**>**

**<p>**Total = {{ (quantity \* price) | currency }}**</p>**

**</div>**

**<script>**

**function** personController($scope) {

$scope.quantity=10;

$scope.price=1;

}

**</script>**

**Angularjs Forms**

An AngularJS form is a collection of input controls like button, input elements. AngularJS has some features for binding data of HTML form input fields to the model object ($scope). You bind an input field to a model property using the ng-model directive

**Html Controls**

HTML input elements are called HTML controls:

* input elements
* select elements
* button elements
* textarea elements

**Example of AngularJS Forms**

**Example**

**<div** ng-app="" ng-controller="formController"**>**

**<form** novalidate**>**

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

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

**<br>**

**</form>**

**<p>** Full Name: {{ firstName }} {{ lastName}}**</p>**

**</div>**

**<script>**

**function** formController ($scope) {

$scope.firstName="Harry";

$scope.lastName="Porter";

}

**</script>**

**Output**

Top of Form

First Name:  
Last Name:

Bottom of Form

Full Name: Harry Porter

**Angularjs Include**

Using AngularJS, we can include or embed Html pages within an Html page using ng-include directive.

Html does not support include or embed Html pages within the html page. To achieve this functionality, you need to use following technologies.

* **Ajax :**Use Ajax to fetch data from a server, and then write the data to the inner HTML of an HTML element.
* **Server Side Technologies:**JSP, PHP and some other server side technologies are use for include or add Html page within a dynamic page.

**AngularJS Side Includes**

Using AngularJS, we can include or embed Html pages within an Html page using ng-include directive.

**Example**

**<div** class="container"**>**

**<div** ng-include="'header.html'"**></div>**

**<div** ng-include="'footer.html'"**></div>**

**</div>**

## Angularjs Table

The ng-repeat directive is use for draw tables easily. Table data are generally repeatable. Following code is design a table using ng-repeat directive;

## Example

**<table>**

**<tr** ng-repeat="x in names"**>**

**<td>**{{ x.Student}}**</td>**

**<td>**{{ x.Marks}}**</td>**

**</tr>**

**</table>**

### Style table Using CSS

You can also style your table using CSS same like below.

## Example

**<style>**

table, th , td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f1f1f1;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

**</style>**

### Example of AngularJS using Table

## Example

**<html>**

**<head>**

**<title>**AngularJS Table**</title>**

**<script** src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"**></script>**

**<style>**

table, th , td {

border: 1px solid grey;

border-collapse: collapse;

padding: 5px;

}

table tr:nth-child(odd) {

background-color: #f1f1f1;

}

table tr:nth-child(even) {

background-color: #ffffff;

}

**</style>**

**</head>**

**<body>**

**<h2>**AngularJS Application Using Table**</h2>**

**<div** ng-app="" ng-controller="studentController"**>**

**<table>**

**<tr>**

**<th>**Name**</th>**

**<th>**Marks**</th>**

**</tr>**

**<tr** ng-repeat="subject in student.subjects"**>**

**<td>**{{ subject.name }}**</td>**

**<td>**{{ subject.marks }}**</td>**

**</tr>**

**</table>**

**</div>**

**<script>**

**function** studentController($scope) {

$scope.student = {

subjects:[

{name:'Physics',marks:60},

{name:'Chemistry',marks:70},

{name:'Math',marks:65},

{name:'English',marks:62},

{name:'Hindi',marks:67}

],

};

}

**</script>**

**</body>**

**</html>**

## Result

## AngularJS Application Using Table

|  |  |
| --- | --- |
| **Name** | **Marks** |
| Physics | 60 |
| Chemistry | 70 |
| Math | 65 |
| English | 62 |
| Hindi | 67 |

## Angularjs Validation

An AngularJS forms and input control can validate input data. It is 100% JavaScript so AngularJS are also use for client side validation.

### Example of AngularJS for client side validation

## Example

<!DOCTYPE html>

**<html>**

**<head>**

**<script** src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.2.26/angular.min.js"**></script>**

**</head>**

**<body>**

**<h2>**Client side Validation Example**</h2>**

**<form** ng-app="" ng-controller="ajvalidater"

name="myForm" novalidate**>**

**<p>**Username:**<br>**

**<input** type="text" name="user" ng-model="user" required**>**

**<span** style="color:red" ng-show="myForm.user.$dirty && myForm.user.$invalid"**>**

**<span** ng-show="myForm.user.$error.required"**>**Username is required.**</span>**

**</span>**

**</p>**

**<p>**Email:**<br>**

**<input** type="email" name="email" ng-model="email" required**>**

**<span** style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid"**>**

**<span** ng-show="myForm.email.$error.required"**>**Email is required.**</span>**

**<span** ng-show="myForm.email.$error.email"**>**Invalid email address.**</span>**

**</span>**

**</p>**

**<p>**

**<input** type="submit"

ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||

myForm.email.$dirty && myForm.email.$invalid"**>**

**</p>**

**</form>**

**<script>**

**function** ajvalidater($scope) {

$scope.user = 'Harry Porter';

$scope.email = 'h.porter@gmail.com';

}

**</script>**

**</body>**

**</html>**

## Output

Top of Form

Username:  


Email:



Bottom of Form

**Note:**The HTML form attribute novalidate is used to disable default browser validation.

**Interview Question**

**Angularjs Interview Question**

**What is AngularJS ?**

AngularJS is a javascript framework used for creating single web page applications.

**What are the key features of Angularjs ?**

* Scope
* Controller
* Model
* View
* Services
* Data Binding
* Directives
* Filters
* Testable

**What is scope in Angularjs ?**

scope is an object that refers to the application model. It is an execution context for expressions. Scopes are arranged in hierarchical structure which mimic the DOM structure of the application. Scopes can watch expressions and propagate events.

**What is controller in Angularjs ?**

In Angular, a Controller is a JavaScript constructor function that is used to augment the Angular Scope. When a Controller is attached to the DOM via the ng-controller directive, Angular will instantiate a new Controller object, using the specified Controller's constructor function.

**What is data binding in Angularjs ?**

Data-binding in Angular apps is the automatic synchronization of data between the model and view components.

**What are the Advantages of using Angularjs**

* Two way data-binding
* MVC pattern
* Provides Static template and angular template
* Can add custom directive
* Provides REST full services
* Provides form validations
* Provides both client and server communication
* Provides dependency injection
* Applying Animations
* Event Handlers

[Read more......](http://www.tutorial4us.com/angularjs/angularjs-features)

**Why called AngularJS ?**

Because HTML has Angular brackets and "ng" sounds like "Angular".

**AngularJS a library, framework, plugin or a browser extension?**

AngularJS is the framework of JavaScript. AngularJS is 100% JavaScript, 100% client-side and compatible with both desktop and mobile browsers. So it's definitely not a plugin or some other native browser extension.

**What are Directives?**

At a high level, directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler ($compile) to attach a specified behavior to that DOM element or even transform the DOM element and its children.

**Angular use the jQuery library?**

Yes, Angular can use jQuery if it's present in your app when the application is being bootstrapped. If jQuery is not present in your script path, Angular falls back to its own implementation of the subset of jQuery that we call jQLite.

**Why is this project called "AngularJS" ?**

Because HTML has Angular brackets and "ng" sounds like "Angular".

**Which browsers support Angular ?**

AngularJS mostly support all new web browser, following browsers supported by angularJS: Safari, Chrome, Firefox, Opera, IE8, IE9 and mobile browsers (Android, Chrome Mobile, iOS Safari).

**What are limitations of Angularjs ?**

* **Not Secure:**It is a framework of JavaScript not a server side programming. Only server side programs provides security, server side authentication and authorization provides security for application.
* **Depends on User:**If any user disable JavaScript in their own browser then nothing will be apply on your application related to JavaScript and AngularJS, Only basic Html page will display.

**What is ng-init directive ?**

The ng-init directive initialize application data same like variable initialization in C language, In c language you initialize int a=10;.

**What is ng-model directive ?**

The ng-model directive binds the value of HTML controls (input, select, textarea) to application data. The ng-model directive defines the variable to be used in AngularJS Application.

**What is ng-repeat directive ?**

The ng-repeat directive repeats an Html element. ng-repeat directive repeats a specific element. [Read more......](http://www.tutorial4us.com/angularjs/angularjs-directives)

**What is expression in Ajgularjs ?**

AngularJS Expressions used to binds application data to HTML. AngularJS expressions are written inside double curly braces such as : {{ expression }}.

**What is filter in Angularjs ?**

Filter are mainly used for modify the data. Filters can be added to expressions and directives using a pipe (|) character. For example uppercase filter are used for converting lower case text to upper case text. [Read more......](http://www.tutorial4us.com/angularjs/angularjs-filters)

**What is Angularjs Include ?**

Using AngularJS, we can include or embed Html pages within an Html page using ng-include directive. [Read more......](http://www.tutorial4us.com/angularjs/angularjs-include)

**How to define controller in angularjs ?**

For define controller in AngularJS application we need **ng-controller** directive names. [Read more......](http://www.tutorial4us.com/angularjs/angularjs-controller)

**How to handel event in Angularjs ?**

AngularJS have its own event directive to handle DOM events like mouse clicks, moves, keyboard presses, change events etc. [Read more......](http://www.tutorial4us.com/angularjs/angularjs-event)

**MVC Architecture**

**What is MVC Architecture ?**

MVC stand for Model View Controller. Model View Controller is a software design pattern for developing web application. It given software application into three interconnected parts, model, view and controller.

**What is view in MVC ?**

It is responsible for displaying output data to the user.

**What is model in MVC ?**

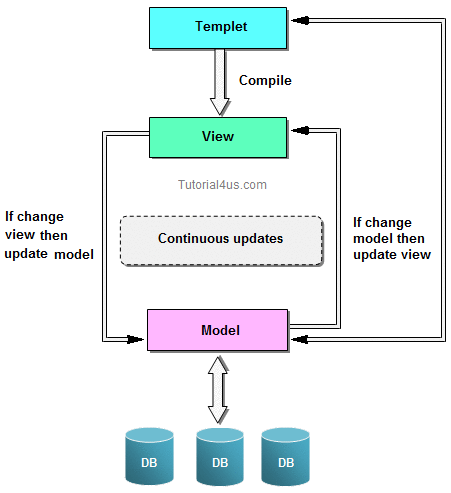
Model is responsible for maintaining data. Model retrieve data form database and also store data in database.

**What is Contoller in MVC ?**

**MVC Architecture**

MVC stand for Model View Controller. Model View Controller is a software design pattern for developing web application. It given software application into three interconnected parts.

* **Model :**It is responsible for maintaining data.
* **View :**It is responsible for displaying output data to the user.
* **Controller :**It is responsible for controls the interactions between the Model and View.



**Model**

**Model**is responsible for maintaining data. Model retrieve data form database and also store data in database.

**View**

**View**It is responsible for displaying output data to the user.

**Controller**

**Controller**It is responsible for controls the interactions between the Model and View. Controllers can read data from a view, control user input, and send input data to the model.