

## **UNIT 4 : ANGULAR JS**

---

### 4.1 Concepts and characteristics of ANGULAR JS

4.1.1 Expressions in ANGULAR JS (Numbers, Strings, Objects, Arrays)

4.1.2 Setting up Environment, ANGULAR JS Filters

4.1.3 Understanding MVC (Model, View, Controller) Architecture

### 4.2 ANGULARJS Directives (ng-app, ng-init, ng-controller, ng-model, ng-repeat)

4.2.1 Some other Directives : ng-class, ng-animate, ng-show, ng-hide

4.2.2 Expressions and Controllers

4.2.3 Filters (Uppercase, Lowercase, Currency, order by)

# Introduction

---

- ❑ 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 extends HTML with new attributes.
  - ❑ AngularJS is perfect for Single Page Applications (SPAs).
  - ❑ AngularJS is easy to learn.
-

- 
- ❑ AngularJS was originally developed in 2009 by Miško Hevery at Brat Tech LLC.



# AngularJS is a JavaScript Framework

---

- ❑ 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.
-

# 4.1 Concepts and characteristics of ANGULAR JS

---

4.1.1 Expressions in ANGULAR JS  
(Numbers, Strings, Objects, Arrays)

4.1.2 Setting up Environment,  
ANGULAR JS Filters

4.1.3 Understanding MVC (Model,  
View, Controller) Architecture

---

## 4.1.1 Expressions in ANGULAR JS (Numbers, Strings, Objects, Arrays)

---

- ❑ 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 }}`

# Numbers

---

```
<script  
src="https://ajax.googleapis.com/ajax/libs/angular  
js/1.6.9/angular.min.js"></script>  
<body>  
  
<div ng-app="" ng-init="a=5;b=10">  
<p>Total in dollar: <span ng-bind="a *  
b"></span></p>  
</div>  
  
</body>  
</html>
```

---



# Strings

---

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body>

<div ng-app="" ng-init="firstName='20';lastName='BCA'">

<p>The full name is: {{ firstName + " " + lastName }}</p>

</div>

</body>
</html>
```

---

# Objects

---

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js">
</script>
<body>

<div ng-app="" ng-init="person={firstName:'20',lastName:'BCA'}">

<p>The name is {{ person.lastName }}</p>

</div>

</body>
</html>
```

---

# Arrays

---

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular
.min.js"></script>
<body>

<div ng-app="" ng-init="points=[1,2,19,2,40]">

<p>The third result is {{ points[2] }}</p>

</div>

</body>
</html>
```

---

## 4.1.2 Setting up Environment, ANGULAR JS Filters

---

- When you open the link <https://angularjs.org/>, you will see there are two options to download AngularJS library
    - **View on GitHub** – By clicking on this button, you are diverted to GitHub and get all the latest scripts.
    - **Download AngularJS** – By clicking on this button, a screen you get to see a dialog box
-

---

- **Downloading and hosting files locally**

- There are two different options : Legacy and Latest. The names themselves are self-descriptive. The Legacy has version less than 1.2.x and the Latest come with version 1.3.x.
  - We can also go with the minimized, uncompressed, or zipped version.
-

- 
- **CDN access** – You also have access to a CDN. The CDN gives you access to regional data centers.
-

# Include AngularJS

---

```
<head>
```

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

```
  </script>
```

```
</head>
```

---

# ANGULAR JS Filters

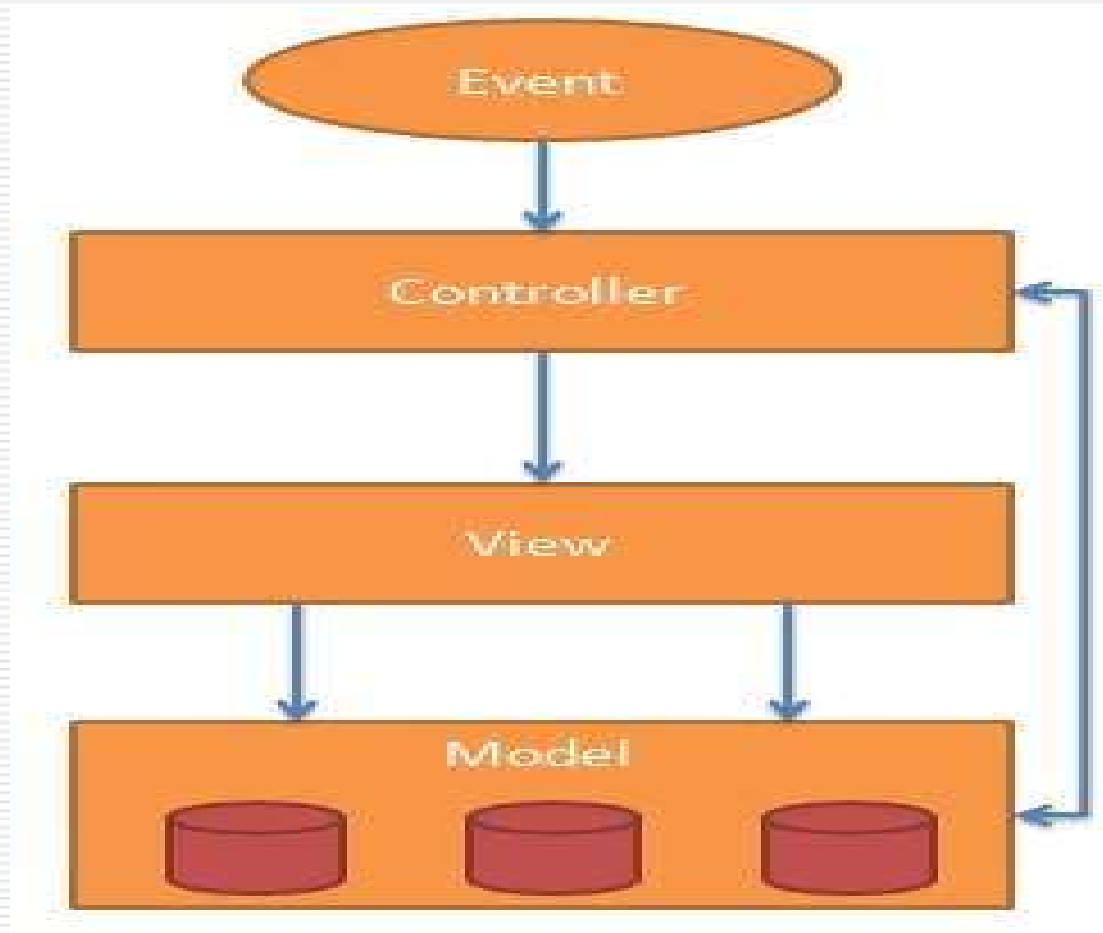
---

- ❑ **currency**
    - Format a number to a currency format.
  - ❑ **date**
    - Format a date to a specified format.
  - ❑ **filter**
    - Select a subset of items from an array.
  - ❑ **json**
    - Format an object to a JSON string.
  - ❑ **limitTo**
    - Limits an array/string, into a specified number of elements/characters.
  - ❑ **lowercase**
    - Format a string to lower case.
  - ❑ **number**
    - Format a number to a string.
  - ❑ **orderBy**
    - Orders an array by an expression.
  - ❑ **uppercase**
    - Format a string to upper case.
-



## 4.1.3 Understanding MVC (Model, View, Controller) Architecture

---



# The Model

---

- ❑ The model is responsible for managing application data.
  - ❑ It responds to the request from view and to the instructions from controller to update itself.
-

# The View

---

- ❑ A presentation of 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.
-

# The Controller

---

- ❑ The controller 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.
-

## 4.2 ANGULARJS Directives (ng-app, ng-init, ng-controller, ng-model, ng-repeat)

---

4.2.1 Some other Directives : ng-class, ng-animate, ng-show, ng-hide

4.2.2 Expressions and Controllers

4.2.3 Filters (Uppercase, Lowercase, Currency, order by)

---

# ng-app

---

- ❑ The ng-app directive tells AngularJS that this is the **root element** of the AngularJS application.
  - ❑ All AngularJS applications **must have a root element**.
  - ❑ You can only have **one ng-app** directive in your HTML document. If more than one ng-app directive appears, the first appearance will be used.
-

# ng-init

---

- ❑ The ng-init Directive is **used to initialize AngularJS Application data.**
  - ❑ It defines the initial value for an AngularJS application and assigns values to the variables.
-

# ng-controller

---

- ❑ AngularJS applications are controlled by controllers.
  - ❑ The ng-controller directive defines the application controller.
  - ❑ A controller is a JavaScript Object, created by a standard JavaScript object constructor.
-



# Example:

---

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

<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 = "TYBCA";
    $scope.lastName = "The Great Class";
});
</script>

</body>
</html>
```

---

# ng-model

---

- With the ng-model directive you can bind the value of an input field to a variable created in AngularJS.
-

# Example:

---

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

<div ng-app="myApp" ng-controller="myCtrl">
Name: <input ng-model="name">
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.name = "20BCA";
});
</script>

</body>
</html>
```

# ng-repeat

---

- ❑ The ng-repeat directive repeats a set of HTML, a given number of times.
  - ❑ The set of HTML will be repeated once per item in a collection.
  - ❑ The collection must be an array or an object.
-

# Example:

---

```
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body ng-app="myApp" ng-controller="myCtrl">
<h1 ng-repeat="x in records">{{x}}</h1>
<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
  $scope.records = [
    "AWD",
    "UNIX",
    "ASP",
    "NT",
  ]
});
</script>

</body>
</html>
```

---

## 4.2.1 Some other Directives : ng-class, ng-animate, ng-show, ng-hide

---

# ng-class

---

- ❑ The ng-class directive dynamically binds one or more CSS classes to an HTML element.
  - ❑ The value of the ng-class directive can be a string, an object, or an array.
  - ❑ If it is a string, it should contain one or more, space-separated class names.
-

# Example:

---

```
<html>
<script
src="https://ajax.googleapis.com/ajax
/libs/angularjs/1.6.9/angular.min.js"
></script>
<style>
.black {
  color:white;
  background-color:black;
  padding:20px;
  font-family:"Courier New";
}
.pink {
  background-color:pink;
  padding:40px;
  font-family:Verdana;
}
</style>
```

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

<p>Choose a class:</p>

<select ng-model="home">
<option value="pink">Pink</option>
<option value="black">Black</option>
</select>

<div ng-class="home">
  <h1>20BCA The Great Class</h1>
  <p>VTCBCSR</p>
</div>

</body>
</html>
```

---



# ng-animate

---

❑ To apply animation use the following CDN:

■ `<script  
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-animate.js"></script>`

- ❑ ng-show
  - ❑ ng-hide
  - ❑ ng-class
  - ❑ ng-view
  - ❑ ng-include
  - ❑ ng-repeat
  - ❑ ng-if
  - ❑ ng-switch
-

# ng-show

```
<!DOCTYPE html>
<html>
<style>
div {
  transition: all linear 0.5s;
  background-color: lightblue;
  height: 100px;
  width: 100%;
  position: relative;
  top: 0;
  left: 0;
}

.ng-hide {
  height: 0;
  width: 0;
  background-color: transparent;
  top: -200px;
  left: 200px;
}

</style>
```

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

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

<body ng-app="ngAnimate">

<h1>Hide the DIV: <input type="checkbox"
ng-model="myCheck" checked></h1>

<div ng-show="myCheck"></div>

</body>
</html>
```

# ng-hide

```
<!DOCTYPE html>
<html>
<style>
div {
  transition: all linear 0.5s;
  background-color: lightblue;
  height: 100px;
  width: 100%;
  position: relative;
  top: 0;
  left: 0;
}

.ng-hide {
  height: 0;
  width: 0;
  background-color: transparent;
  top: -200px;
  left: 200px;
}

</style>
```

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

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

<body ng-app="ngAnimate">

<h1>Hide the DIV: <input type="checkbox"
ng-model="myCheck"></h1>

<div ng-hide="myCheck"></div>

</body>
</html>
```

## 4.2.2 Expressions and Controllers

---

# Controllers

---

- ❑ AngularJS controllers **control the data** of AngularJS applications.
  - ❑ AngularJS controllers are regular **JavaScript Objects**.
  - ❑ AngularJS applications are controlled by controllers.
  - ❑ The **ng-controller** directive defines the application controller.
  - ❑ A controller is a **JavaScript Object**, created by a standard JavaScript **object constructor**.
-

## 4.2.3 Filters (Uppercase, Lowercase, Currency, order by)

---

# Uppercase

---

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

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

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

</div>

<script>
angular.module('myApp', []).controller('personCtrl', function($scope) {
    $scope.firstName = "Tybca",
});
</script>
</body>
</html>
```

---

# Lowercase

---

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

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

<p>The name is {{ firstName | lowercase }}</p>

</div>

<script>
angular.module('myApp', []).controller('personCtrl', function($scope) {
    $scope.firstName = "TyBca",
});
</script>
</body>
</html>
```

---



# Currency

---

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

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

<h1>Price: {{ price | currency }}</h1>

</div>

<script>
var app = angular.module('myApp', []);
app.controller('costCtrl', function($scope) {
    $scope.price = 58;
});
</script>
</body>
</html>
```

---

# order by

---

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6
.9/angular.min.js"></script>
<body>

<p>Click the table headers to change the sorting
order:</p>

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

<table border="1" width="100%">
<tr>
<th ng-click="orderByMe('name')">Name</th>
<th ng-click="orderByMe('type')">Type</th>
</tr>
<tr ng-repeat="x in names | orderBy:myOrderBy">
<td>{{x.name}}</td>
<td>{{x.type}}</td>
</tr>
</table>

</div>
```

```
<script>
angular.module('myApp', []).controller('namesCtrl',
function($scope) {
    $scope.names = [
        {name:'Aksyay',type:'Bollywood'},
        {name:'NTR',type:'South'},
        {name:'James Bond',type:'Hollywood'},
        {name:'Mahesh Babu',type:'South'},
        {name:'Hritik',type:'Bollywood'},
        {name:'Allu Arjun',type:'South'}
    ];
    $scope.orderByMe = function(x) {
        $scope.myOrderBy = x;
    }
});
</script>

</body>
</html>
```