CS603 – Web Engineering

PREPARED BY: DR. REEMA PATEL

Angular JS

AngularJS

- JavaScript framework that helps build web application
- Open source framework maintained by Google and community
- MVC framework for dynamic application
- Extends HTML for two way binding for automatic synchronization of models and views

- Website builds using AngularJS
    - https://www.madewithangular.com/

AngularJS

- Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM.
- Neither of these address the root problem that HTML was not designed for dynamic views".
- Structure, Quality and Organization
- Lightweight (< 36KB compressed and minified)
- Free
- Separation of concern
- Modularity
- Extensibility & Maintainability
- Reusable Components
- " HTML? Build UI Declaratively! CSS? Animations! JavaScript? Use it the plain old way!"

Jquery to Angular JS

- Allows for DOM Manipulation
- jQuery is seen as augmentation, not infrastructure
    - Does not provide structure to your code

- Does not allow for two way binding
- Architecture focused : building applications, not web pages
- Declarative UI means view based functionality is apparent
- Distinct Model Layer – not the DOM

AngularJS Taxonomy

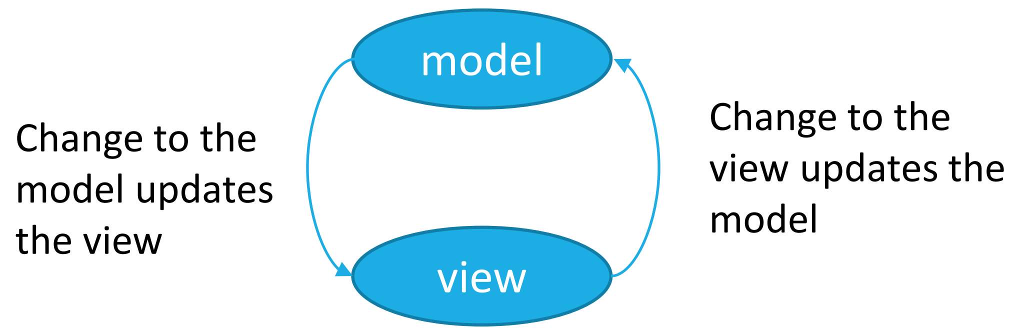
- Module: organizational and reusable container for different parts of your apps
  - Module for each feature
- Controller: business logic for views
- Directives: extend HTML used as widgets often
- Services: reusable business logic independent of views

Features of AngularJS

- Two-way Data Binding – Model as single source of truth
- Directives – Extend HTML
- MVC
- Dependency Injection
- Testing
- Deep Linking (Map URL to route Definition)
- Server-Side Communication

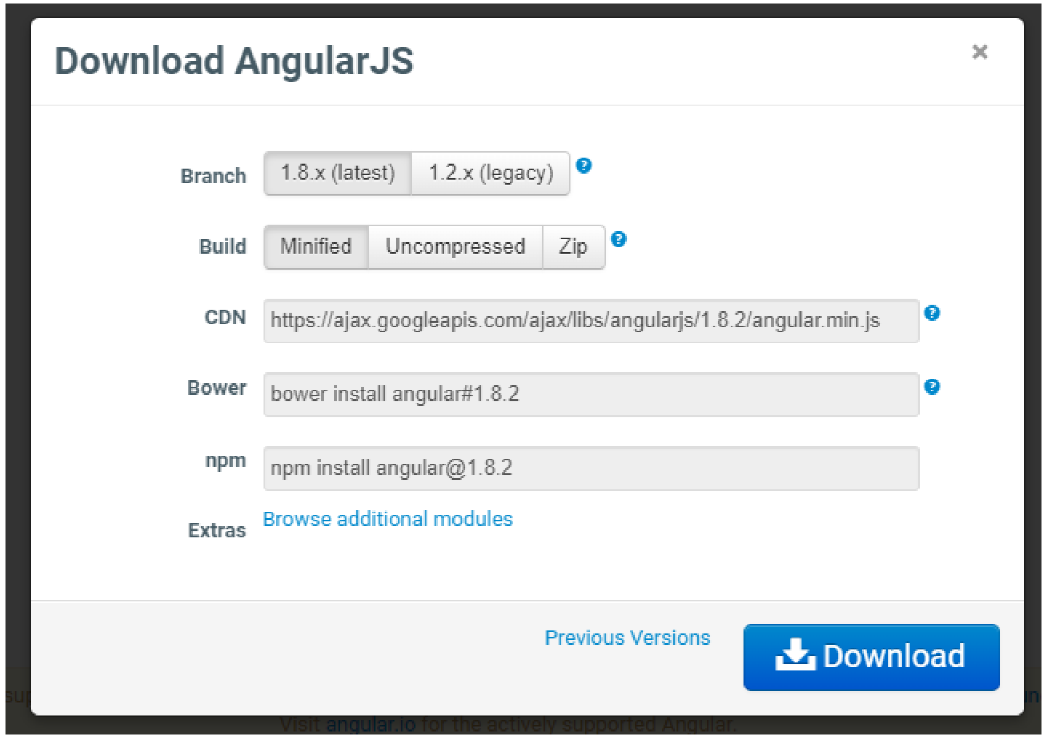
Benefits of AngularJS

- Two way data-binding


model
view
Change to the model updates the view
Change to the view updates the model

AngularJS Download

- https://angularjs.org/



Download AngularJS

Branch: [✓ 1.8.x (latest)] 1.2.x (legacy)

Build: [✓ Minified] Uncompressed Zip

CDN: https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js

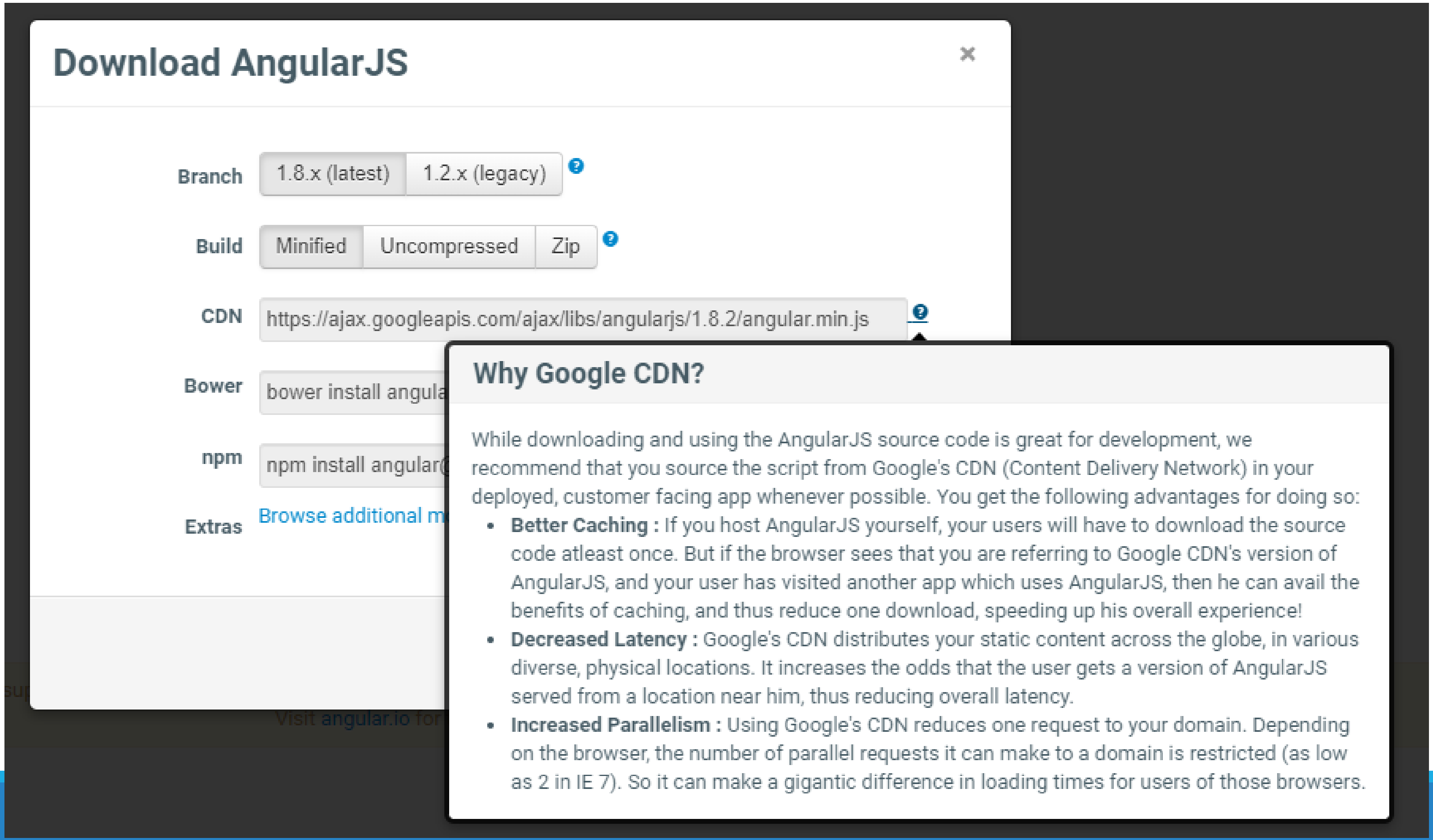
Bower: bower install angular#1.8.2

npm: npm install angular@1.8.2

Extras: Browse additional modules

Visit angular.io for the actively supported Angular.

AngularJS Download



Download AngularJS

Branch: 1.8.x (latest) | 1.2.x (legacy)

Build: Minified | Uncompressed | Zip

CDN: https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js

Bower: bower install angula

npm: npm install angular@

Extras: Browse additional m

Why Google CDN?

While downloading and using the AngularJS source code is great for development, we recommend that you source the script from Google's CDN (Content Delivery Network) in your deployed, customer facing app whenever possible. You get the following advantages for doing so:

- **Better Caching :** If you host AngularJS yourself, your users will have to download the source code atleast once. But if the browser sees that you are referring to Google CDN's version of AngularJS, and your user has visited another app which uses AngularJS, then he can avail the benefits of caching, and thus reduce one download, speeding up his overall experience!
- **Decreased Latency :** Google's CDN distributes your static content across the globe, in various diverse, physical locations. It increases the odds that the user gets a version of AngularJS served from a location near him, thus reducing overall latency.
- **Increased Parallelism :** Using Google's CDN reduces one request to your domain. Depending on the browser, the number of parallel requests it can make to a domain is restricted (as low as 2 in IE 7). So it can make a gigantic difference in loading times for users of those browsers.

Angular JS

- AngularJS applications consist of 3 components:
- View, which is the HTML of the application
- Model, which defines the data for the View
- Controller, which consists of JavaScript functions to alter the data.
- The scope is the Model that binds the View to the Controller.
- AngularJS scopes are JavaScript objects with methods and properties available for both the View and the controller. When making a controller in an AngularJS application, you can pass in a scope object as an argument.

Example

```
<html ng-app>
<head> <script type="text/javascript" src=angular.min.js></script> </head>
<body>
<div> 10 + 20 = {{ 10+20 }} </div>
<div>
        {{1==2}}
        {{1==1}}
        {{ {name: 'Reema', age: '30'}.age }}
        {{ ['David', 'Sara', 'Pam'][0] }}
</div> </body> </html>
```

Expressions

Expressions allow you to execute some computation in order to return a desired value.

- {{ 1 + 1 }}
- {{ 946757880 | date }}
- {{ user.name }}

you shouldn’t use expressions to implement any higher-level logic.

Modules and Controllers

- What is a module in AngularJS
  - A module is a container for different parts of application i.e., controllers, services, directives, filters, etc.
  - You can think of a module as a Main() method in other types of applications

- How to create a module
  - Use the angular object’s module() method to create a module
  - var myApp = angular.module(“myModule”, []); //[] dependencies

Modules and Controllers

- What is a controller in angular
  - A controller is a javascript function. Controller can build a model for the view to display

- How to create a controller in angular

```
var myController = function ($scope) {
$scope.message = “AngularJS Introdcution”;
}
```

Directives

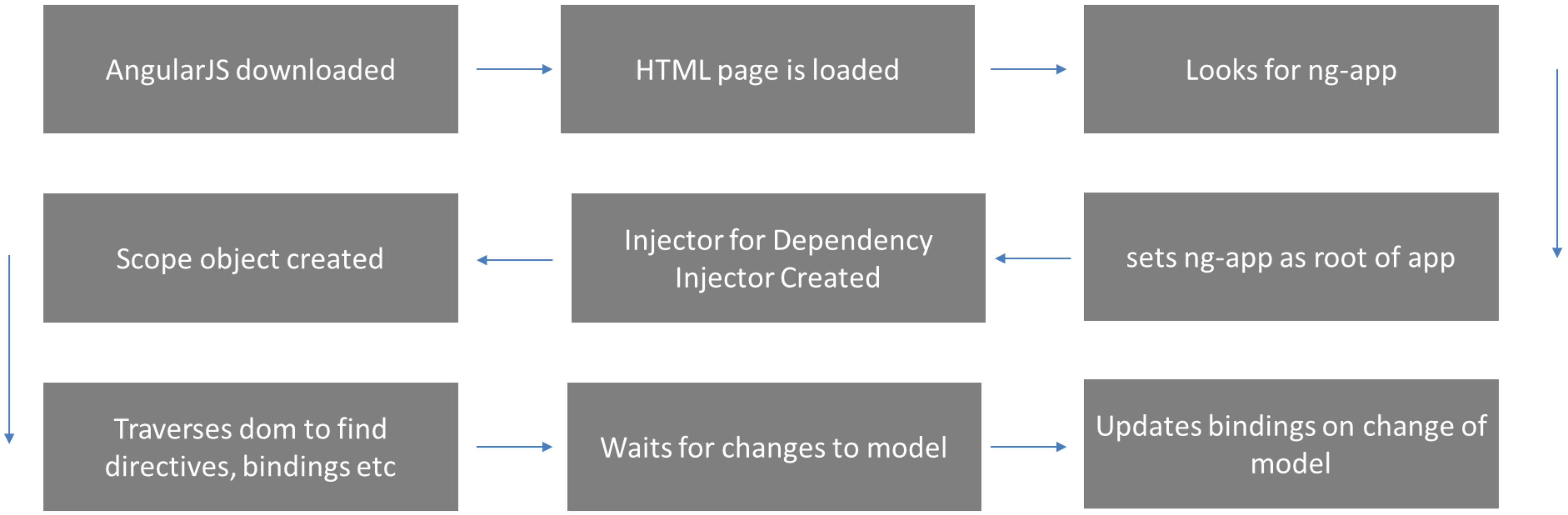
- Directives are markers (such as attributes, tags, and class names) that tell AngularJS to attach a given behaviour to a DOM element (or transform it, replace it, etc.)

- Some angular directives
- The ng-app - Bootstrapping your app and defining its scope.
- The ng-controller - defines which controller will be in charge of your view.
- The ng-repeat - Allows for looping through collections

AngularJS Taxonomy

MODULE	CONTROLLER	DIRECTIVES	SERVICES
Organizational and reusable container for different parts of your apps. Module for each feature.	Business logic for views	Extend HTML, used as widgets often	Reusable business logic independant of views

AngularJS Bootstrapping


AngularJS downloaded
HTML page is loaded
Looks for ng-app
sets ng-app as root of app
Injector for Dependency Injector Created
Scope object created
Traverses dom to find directives, bindings etc
Waits for changes to model
Updates bindings on change of model

AngularJS Scope

- The scope is the binding part between the HTML (view) and the JavaScript (controller).
- The $scope variable – Link your controllers and view
- The scope is an object with the available properties and methods.
- The scope is available for both the view and the controller.

```
app.controller('myCtrl', function($scope) {
  $scope.carname = "Volvo";
});
```

AngularJS Scope

- If we consider an AngularJS application to consist of:
  - View, which is the HTML.
  - Model, which is the data available for the current view.
  - Controller, which is the JavaScript function that makes/changes/removes/controls the data.
- Then the scope is the Model.
- The scope is a JavaScript object with properties and methods, which are available for both the view and the controller.

Attaching a complex object to the scope

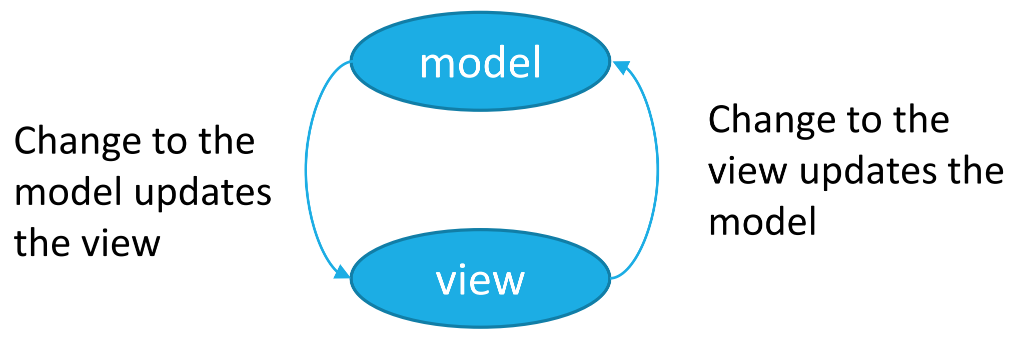
```
myApp.controller("con1", function ($scope){
        var employee = {
                firstname: 'Reema',
                lastname: 'Patel',
                gender: 'Male'
        };
        $scope.employee = employee;
});
```

```
<div ng-controller = "con1">
        <div>firstname : {{employee.firstname}}</div>
        <div>lastname : {{employee.lastname}}</div>
        <div>gender : {{employee.gender}}</div>

</div>
```

Two way data binding in AngularJS

- Keeps the model and view in sync at all times,
  - that is a change in model updates the view and
  - a change in the view updates the model


model
view
Change to the model updates the view
Change to the view updates the model

- ng-model directive can be used with
  - Text
  - Select
  - Textarea
- Binding expression updates the view when the model changes {{message}}
- ng-model directives updates the model when the view changes
- <input type="text" ng-model="message" />

Two way data binding in AngularJS

```
<div ng-controller = "modelupdate">
        <input type="text" ng-model = "message">
        <input type="text" ng-model = "greetings">
        {{ message }}
        {{greetings}}
</div>
```

```
myApp.controller("modelupdate", function ($scope) {
        $scope.message = "Hello AngularJS";
});
```

AngularJS Filter

- AngularJS filters allow users to format data in the user interface without altering the original format.
- AngularJS provides a plethora of filters to transform data:
- **currency**: Used to format a number to a currency format
- **date**: Used to format a date to some format
- **filter**: Select a subset from a set of items
- **json**: Used to format an object to a JSON string
- **limitTo**: Used to limit an array/string, into a specific number of elements/characters
- **lowercase**: Used to format a string to lower case
- **number**: Can format a numerical value to a string
- **orderBy**: Sorts an array by an expression
- **uppercase**: Used to format a string to upper case.

AngularJS Filter

- Filters in AngularJS can be added to expressions used in the pipeline | character followed by a filter. Let's look at an example:
- **div** ng-app="exampleApp" ng-controller="exampleController">
  <**p**>The name is {{ LastName | uppercase }}</**p**>
  </**div**>

Routing in AngularJS

- Routing allows your application to route different pages without reloading the entire application.
- Use the **ngRoute** module if you want to navigate between different pages within your application but also want it to be an SPA (Single Page Application) with no page reloading.
- To enable routing in your application, you need to integrate the AngularJS Route module:
- **<script** src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-route.js"> </**script**>
- After integration, you need to add the **ngRoute** dependency in your application module:
- = angular.module("myApp", ["ngRoute"]);

Components Covered in Angular JS

- Introduction of Angular JS
- Two Way data binding Example – All type of input types
- Basic Different Directives Examples
  - ng-model
  - ng-change
  - ng-click
  - ng-repeat
  - ng-init
  - ng-show
  - ng-controller
  - ng-app
- Basic filters in Angular JS
- Factory example in Angular JS
- Form in Angular JS – tags available for form fields
- Other components as per covered in lecture – Demo Program is uploaded in classroom.