Unit-4 Angular JS

Points..

- Introduction
- Expressions
- MVC
- Data Binding
- Directives
- Filters

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.
- It is an excellent framework for building Single Phase Applications (SPA) and line of business applications.
- AngularJS was originally developed in 2008-2009 by **Misko Hevery** and **Adam Abrons** and is now maintained by Google.
- It is easy to use all you need to know to work with AngularJS is the basics of HTML, CSS, and Javascript, not necessary to be an expert in these technologies.
- AngularJS allows us to work with components and hence we can use them again which saves time and unnecessary code.

Advantage of AngularJS

- Dependency Injection
- Two way data binding
- Testing
- Model View Controller
- Directives, filters, modules, routes etc.

Advantage of AngularJS

Depending Injection:

- Dependency Injection is a software design pattern. It works on the basis of Inversion of Control.
- Inversion control means objects do not create other objects. Instead, they get these objects from an outside source. The dependent object is not created by the primary object after that then uses its methods. Instead, an external source creates the dependent object and gives it to the source object for further usage.

Data Model Binding:

- Data Binding in AngularJS is a two-way process,i.e the view layer of the MVC architecture is an exact copy of the model layer. You don’t need to write special code to bind data to the HTML controls.

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.

Advantage of AngularJS

Model View Controller(MVC)

□ software pattern used to develop an application that consists of three components

Model

- Used to manage the application data.

View

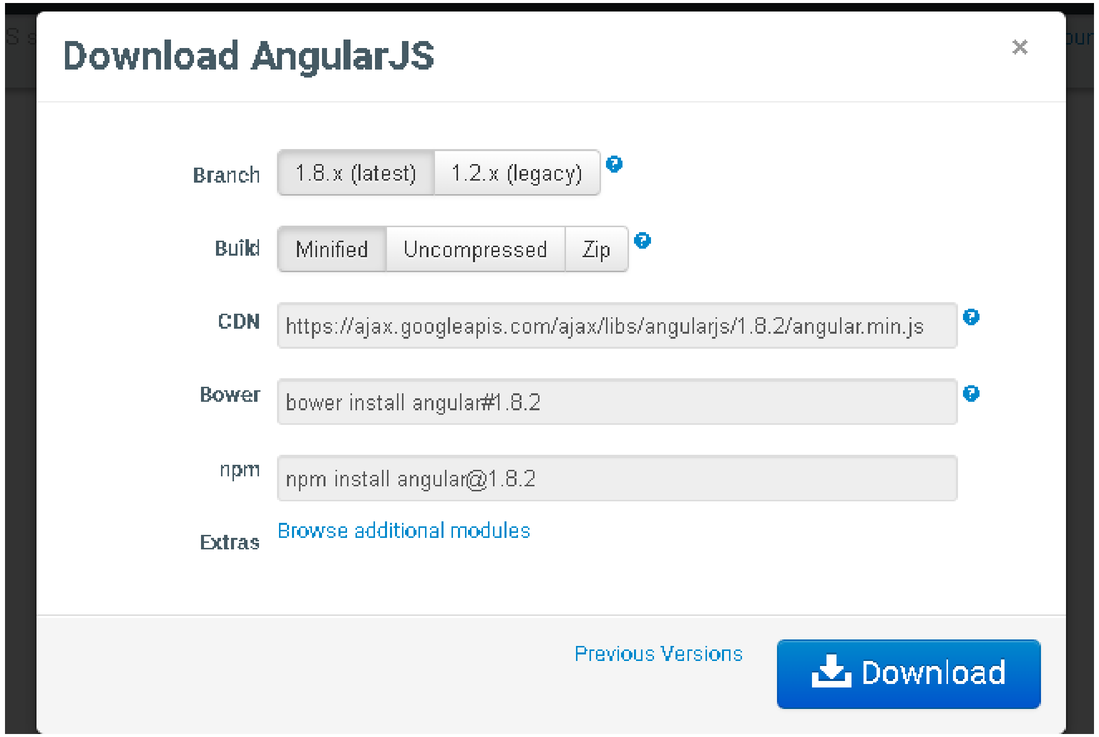
- Responsible for displaying the application data.

Controller

- The main job is to connect the model and the view component.

AngularJS - Environment Setup

- 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

- **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 1.8.x.
  - We can also go with the minimized, uncompressed, or zipped version.
- **CDN access** – You also have access to a CDN.

AngularJS - Environment Setup

Include AngularJS

```
<head>
   <script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js">
   </script>
</head>
```

```
<head>
	<script src="../Scripts/angular.min.js"></script>
</head>
```

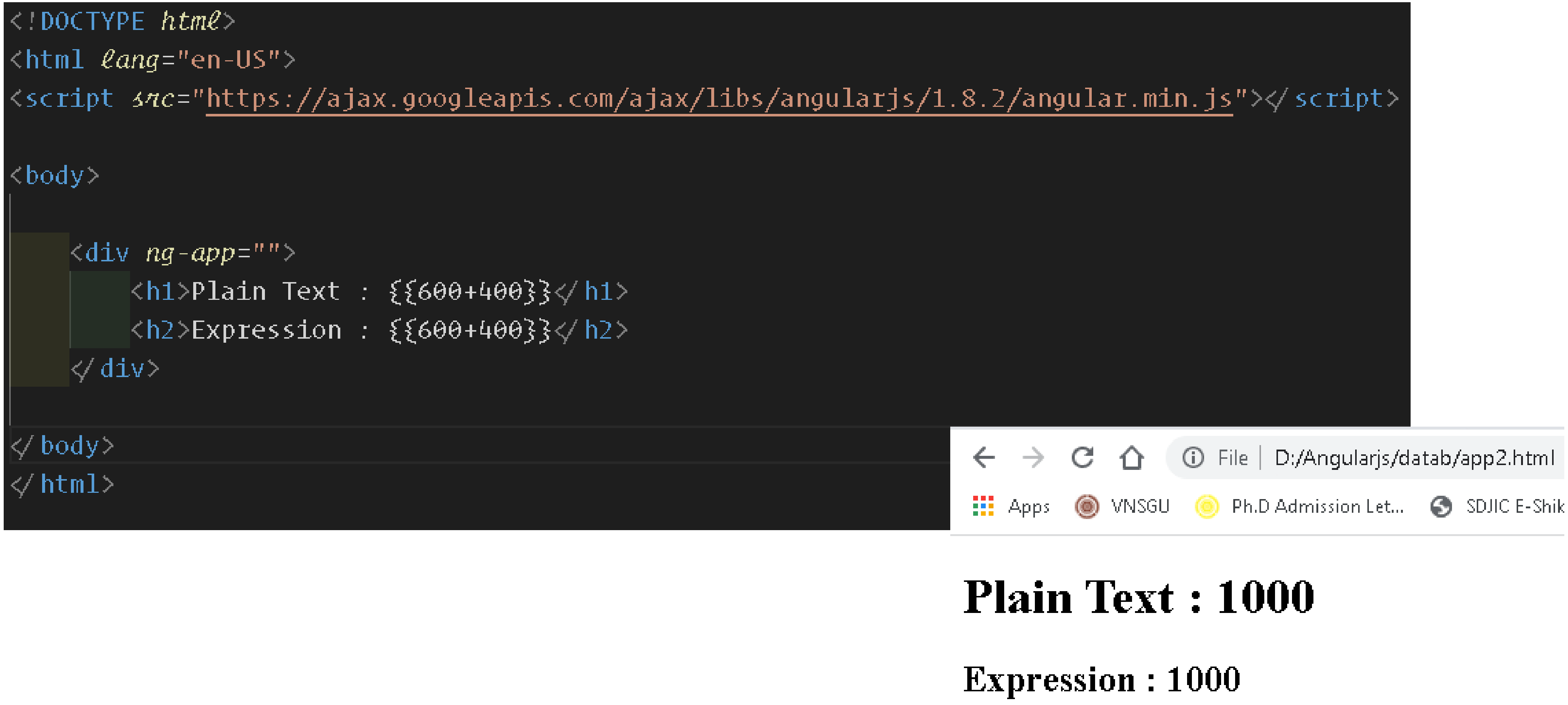
AngularJS - Environment Setup

Point to AngularJS app

- Next, it is required to tell which part of HTML contains the AngularJS app.
- You can do this by adding the ng-app attribute to the specified HTML element of the AngularJS app.
- You can either add it to the root html element or the body element as shown below or any other element as per your requirement.

```
<body ng-app = "">
  <!--AngularJS is enable inside it -->
</body>
```

AngularJS - Environment Setup



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="">
        <h1>Plain Text : {{600+400}}</h1>
        <h2>Expression : {{600+400}}</h2>
    </div>

</body>
</html>
```

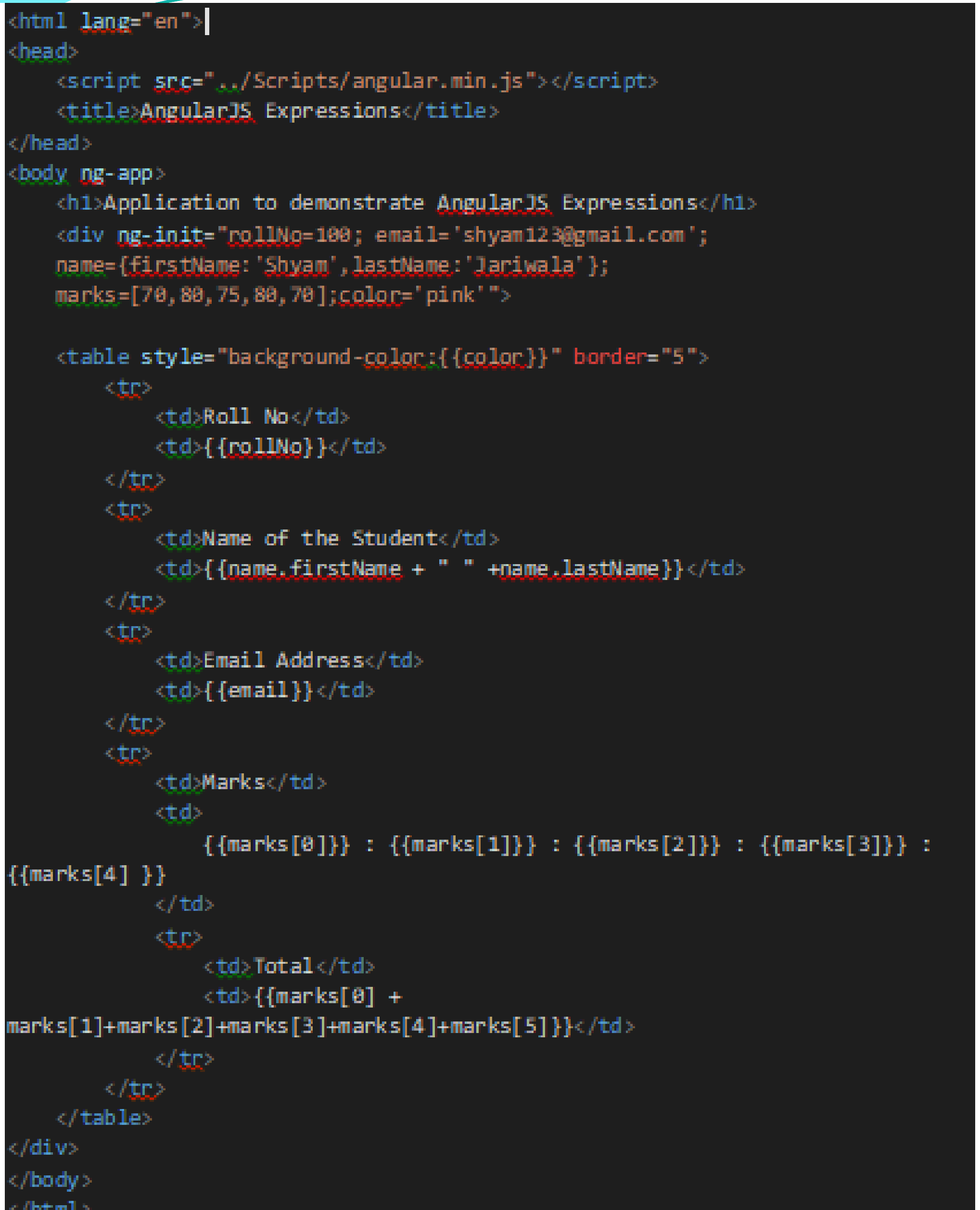
```
Plain Text : 1000

Expression : 1000
```

By:Prof. Khushbu K. Surati

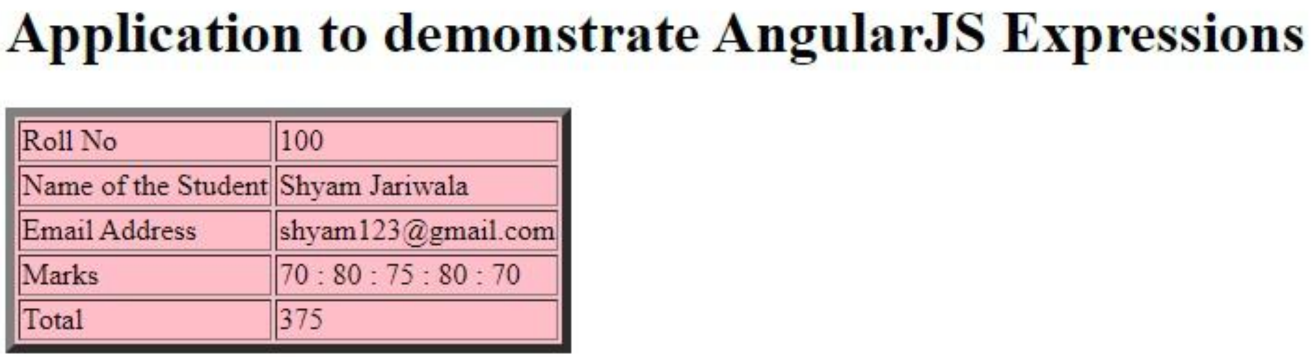
AngularJS – Expressions

- Expressions are used to bind application data to HTML.
- Written inside double curly braces such as in {{ expression}}
- Expressions behave similar to ngbind directives
- AngularJS expressions are pure JavaScript expressions and output the data where they are used
- We can use numbers, strings, arrays, objects type data and JS operators in the expressions

AngularJS – Expressions (Expressions.html)



```
<html lang="en">
<head>
    <script src="../Scripts/angular.min.js"></script>
    <title>AngularJS Expressions</title>
</head>
<body ng-app>
    <h1>Application to demonstrate AngularJS Expressions</h1>
    <div ng-init="rollNo=100; email='shyam123@gmail.com';
    name={firstName:'Shyam',lastName:'Jariwala'};
    marks=[70,80,75,80,70];color='pink'">

    <table style="background-color:{{color}}" border="5">
        <tr>
            <td>Roll No</td>
            <td>{{rollNo}}</td>
        </tr>
        <tr>
            <td>Name of the Student</td>
            <td>{{name.firstName + " " +name.lastName}}</td>
        </tr>
        <tr>
            <td>Email Address</td>
            <td>{{email}}</td>
        </tr>
        <tr>
            <td>Marks</td>
            <td>
                {{marks[0]}} : {{marks[1]}} : {{marks[2]}} : {{marks[3]}} :
{{marks[4] }}
            </td>
            <tr>
                <td>Total</td>
                <td>{{marks[0] +
marks[1]+marks[2]+marks[3]+marks[4]+marks[5]}}</td>
            </tr>
        </tr>
    </table>
</div>
</body>
</html>
```



Application to demonstrate AngularJS Expressions

Roll No	100
Name of the Student	Shyam Jariwala
Email Address	shyam123@gmail.com
Marks	70 : 80 : 75 : 80 : 70
Total	375

AngularJS - MVC Architecture:

- Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts.

Model

- It is the lowest level of the pattern responsible for maintaining data.

View

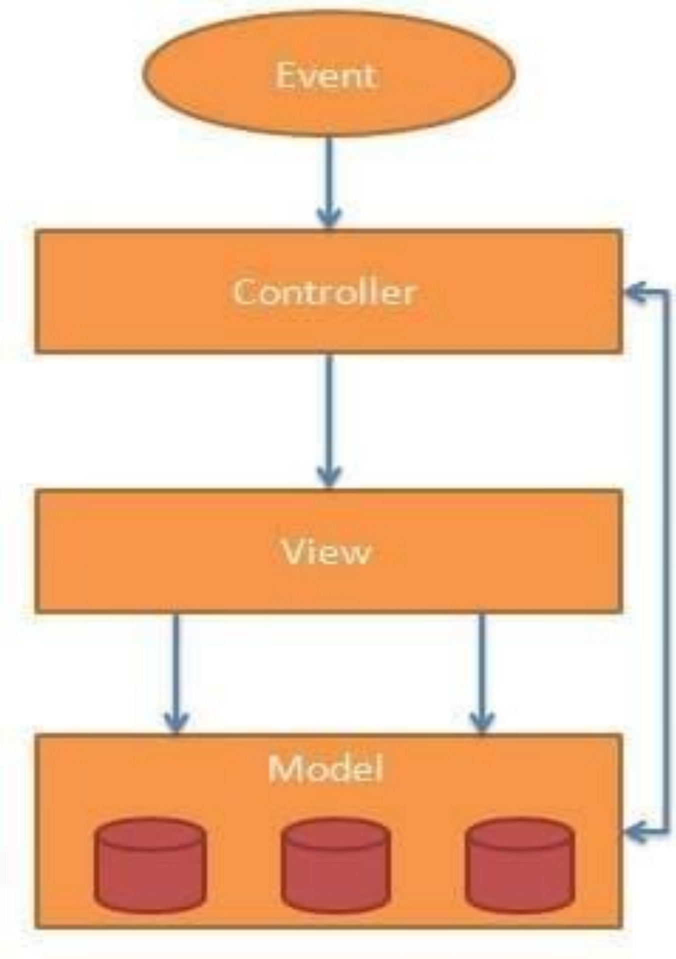
- It is responsible for displaying all or a portion of the data to the user.

Controller

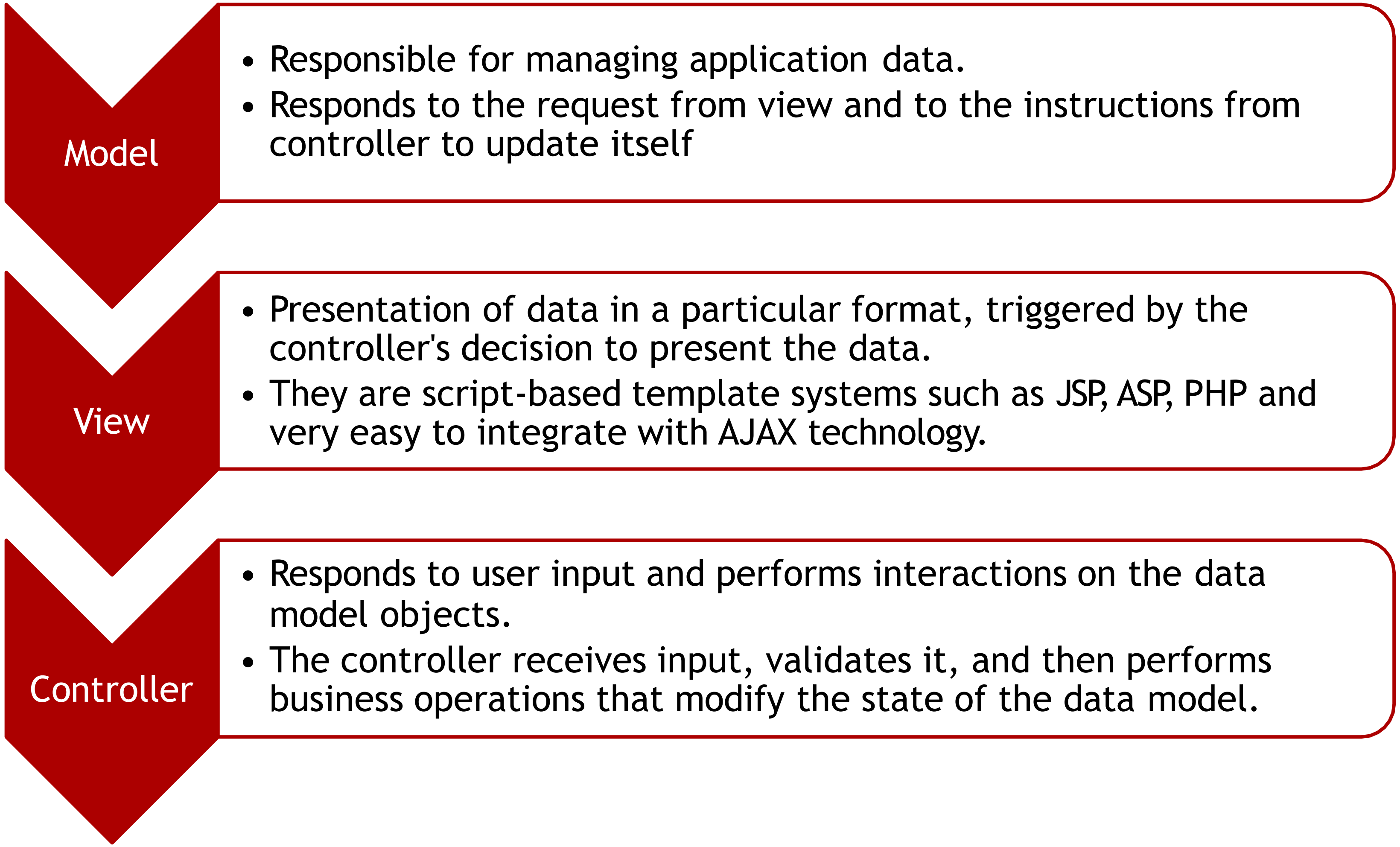
- It is a software Code that controls the interactions between the Model and View.

AngularJS - MVC Architecture:

- MVC is popular because it isolates the application logic from the user interface layer and supports separation of concerns.
- The controller receives all requests for the application and then works with the model to prepare any data needed by the view.
- The view then uses the data prepared by the controller to generate a final presentable response.



AngularJS - MVC Architecture:



Model

- Responsible for managing application data.
- Responds to the request from view and to the instructions from controller to update itself

View

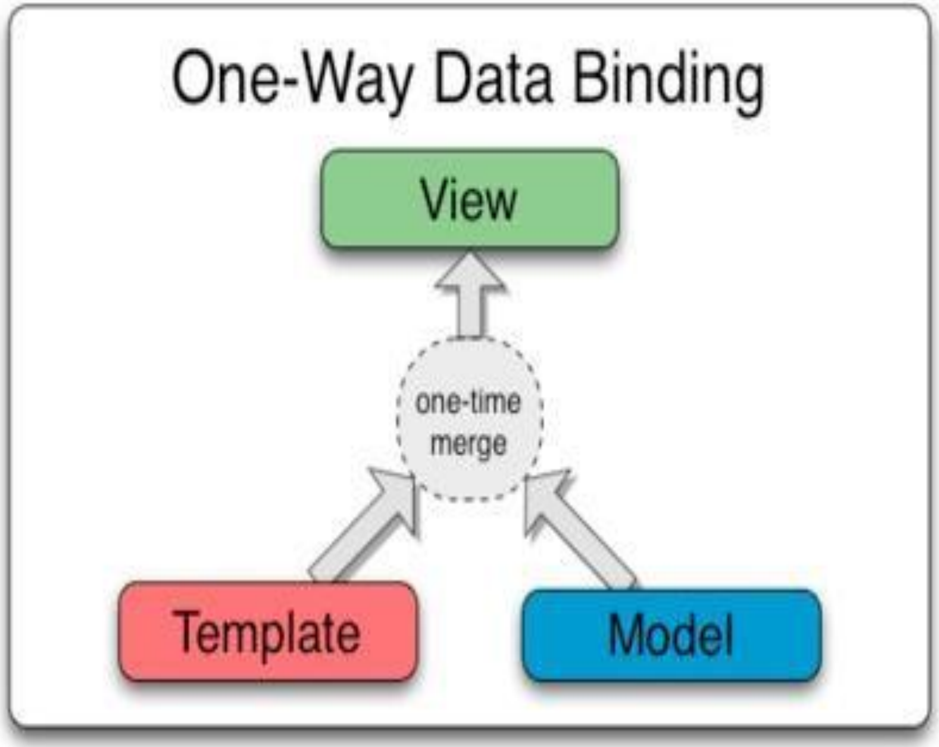
- Presentation of data in a particular format, triggered by the controller's decision to present the data.
- They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology.

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.

AngularJS Data Binding:

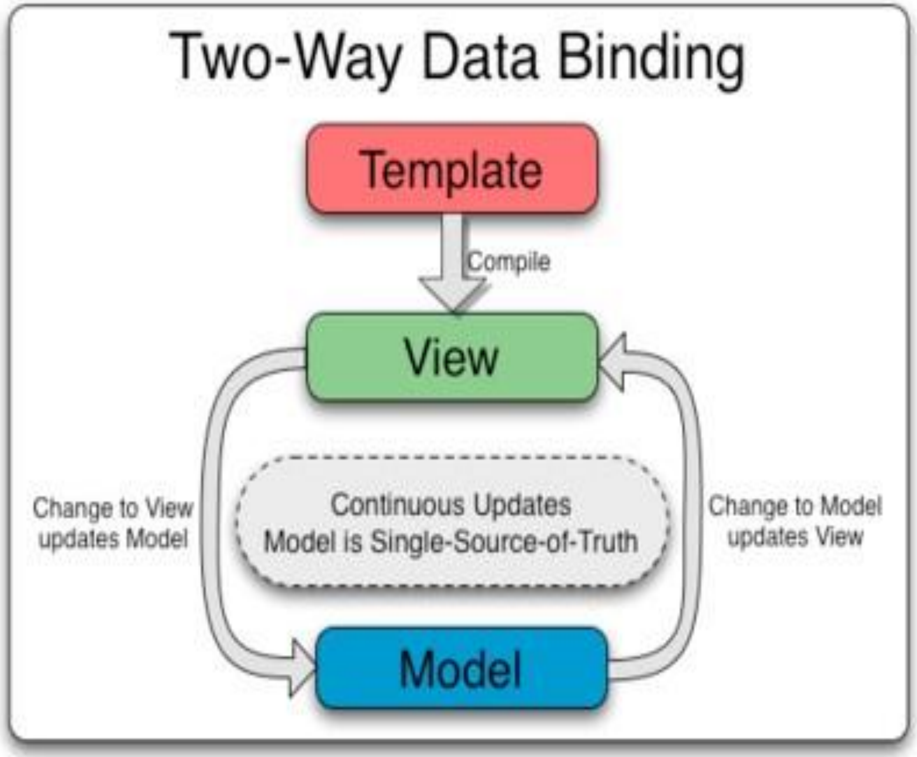
- Data binding is a very useful and powerful feature used in software development technologies. It acts as a bridge between the view and business logic of the application.
- AngularJS follows Two-Way data binding model.



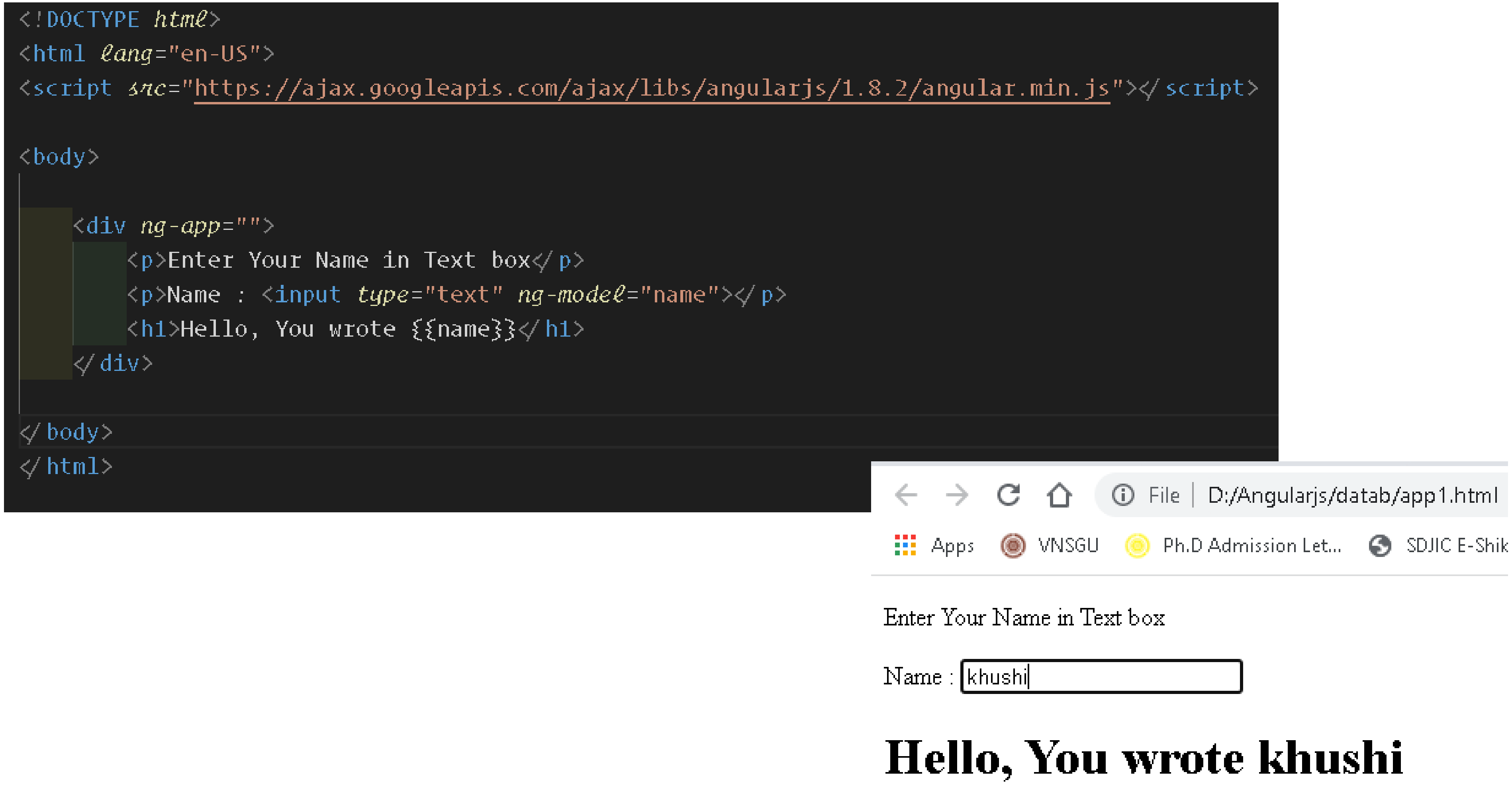
By:Prof. Khushbu K. Surati

AngularJS 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

AngularJS Data Binding:



```html
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="">
        <p>Enter Your Name in Text box</p>
        <p>Name : <input type="text" ng-model="name"></p>
        <h1>Hello, You wrote {{name}}</h1>
    </div>

</body>
</html>
```

Enter Your Name in Text box

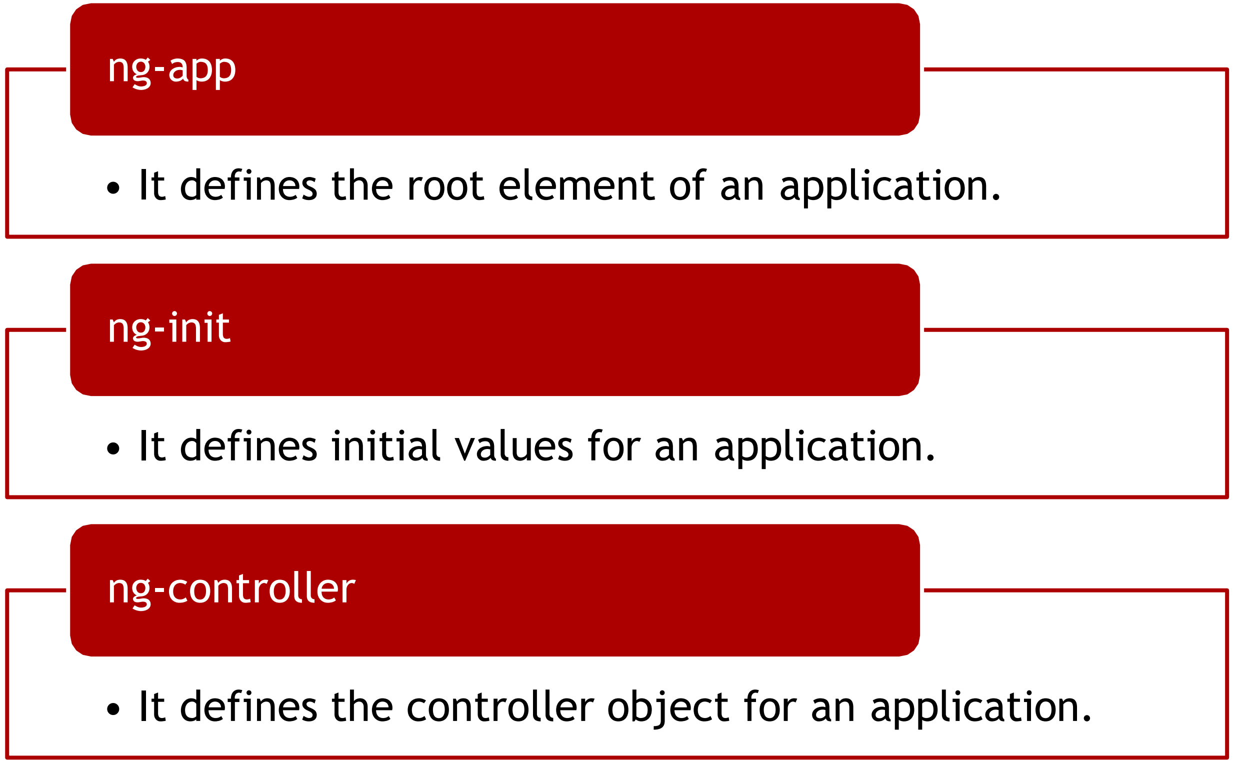
Name : khushi

Hello, You wrote khushi

By:Prof. Khushbu K. Surati

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.



ng-app

- It defines the root element of an application.

ng-init

- It defines initial values for an application.

ng-controller

- It defines the controller object for an application.

AngularJS Directives:

ng-model

- It binds the value of html controls to application data.

ng-repeat

- It defines a template for each data in a collection.

ng-class

- It specifies css classes on html elements.

ng-animate

- Provides support for CSS-based animations (keyframes and transitions) as well as JavaScript-based animations via callback hooks.

ng-hide

- It hides or shows html elements.

ng-show

- It shows or hides html elements.

AngularJS Directives:

ng-app directive

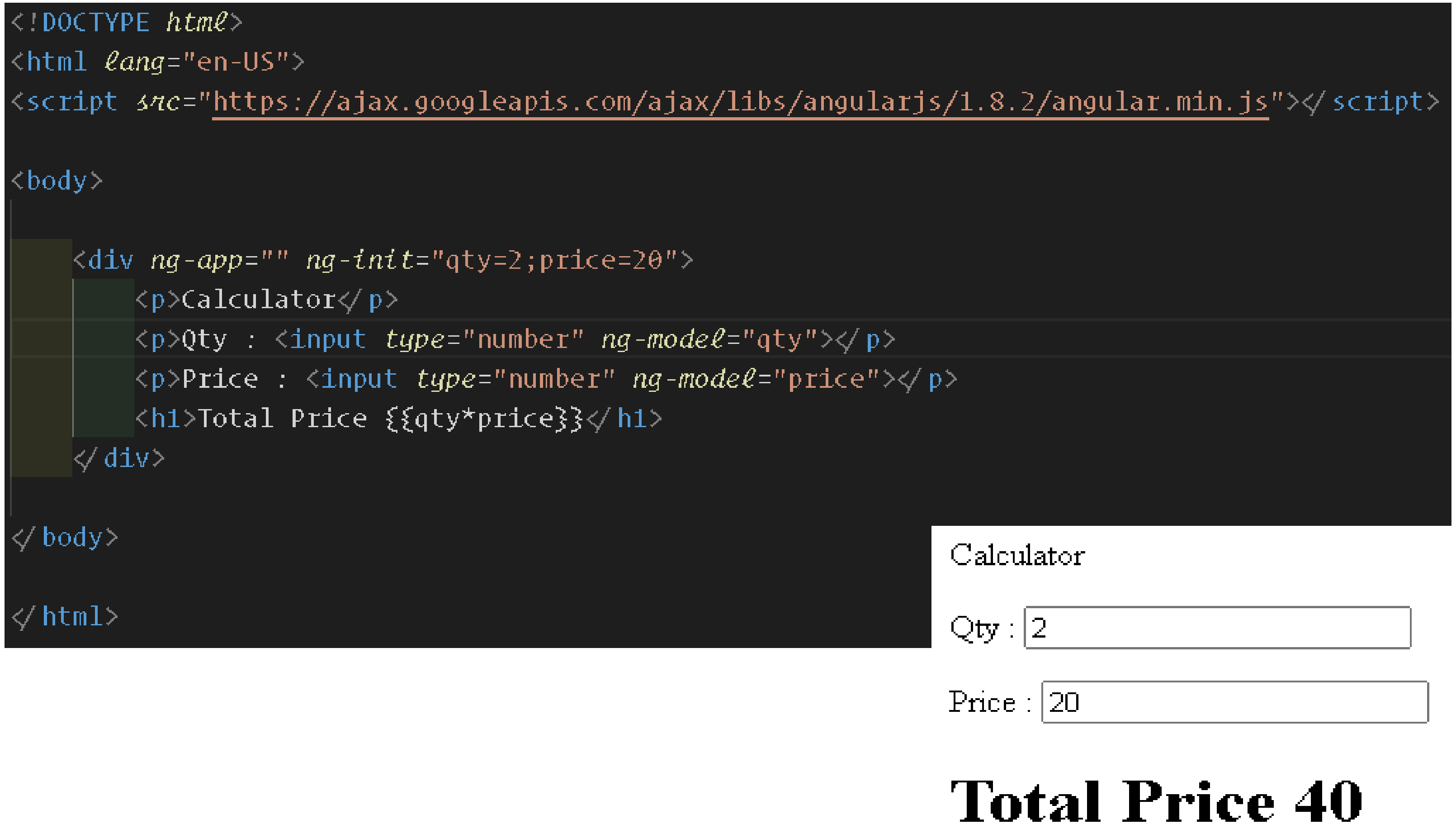
- The ng-app directive defines the root element.
- It starts an AngularJS Application and automatically initializes or bootstraps the application when web page containing AngularJS Application is loaded.
- It is also used to load various AngularJS modules in AngularJS Application.

ng-init directive

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

AngularJS Directives:

ng-init directive



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="" ng-init="qty=2;price=20">
        <p>Calculator</p>
        <p>Qty : <input type="number" ng-model="qty"></p>
        <p>Price : <input type="number" ng-model="price"></p>
        <h1>Total Price {{qty*price}}</h1>
    </div>

</body>

</html>
```

Calculator

Qty : 2

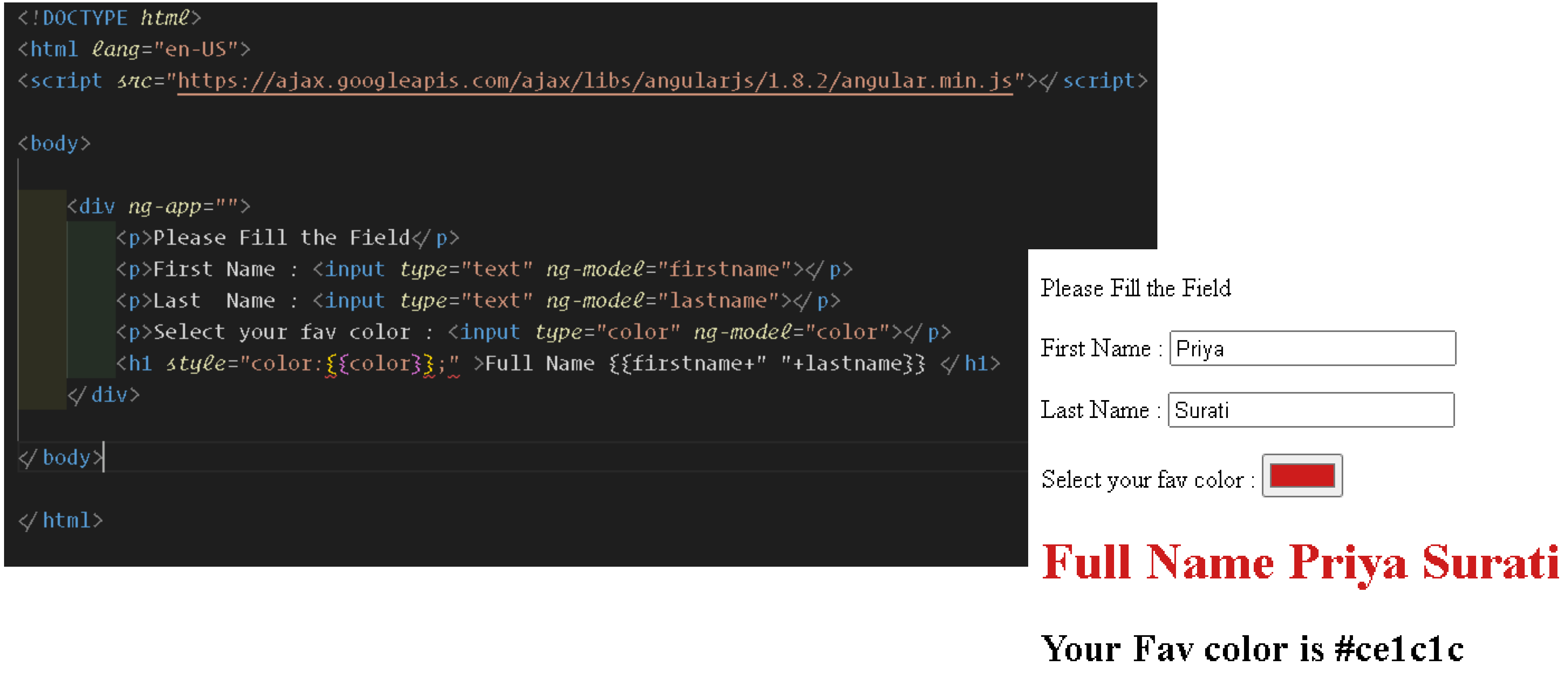
Price : 20

Total Price 40

AngularJS Directives:

ng-model directive:

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



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="">
        <p>Please Fill the Field</p>
        <p>First Name : <input type="text" ng-model="firstname"></p>
        <p>Last  Name : <input type="text" ng-model="lastname"></p>
        <p>Select your fav color : <input type="color" ng-model="color"></p>
        <h1 style="color:{{color}};" >Full Name {{firstname+" "+lastname}} </h1>
    </div>

</body>

</html>
```

Please Fill the Field

First Name : Priya

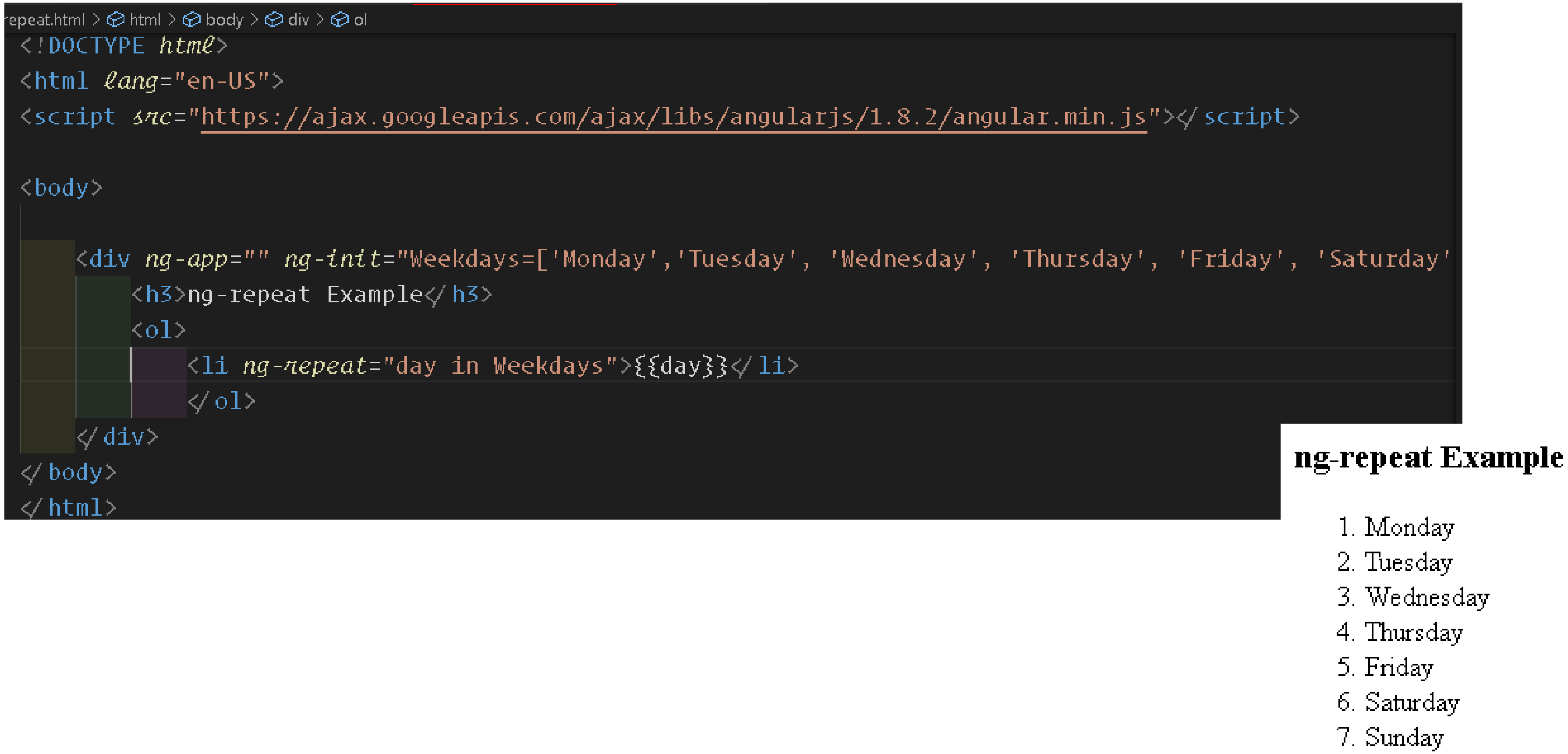
Last Name : Surati

Select your fav color : 🟥

Full Name Priya Surati

Your Fav color is #ce1c1c

AngularJS Directives:



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="" ng-init="Weekdays=['Monday','Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'
        <h3>ng-repeat Example</h3>
        <ol>
            <li ng-repeat="day in Weekdays">{{day}}</li>
            </ol>
    </div>
</body>
</html>
```

ng-repeat Example

1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday

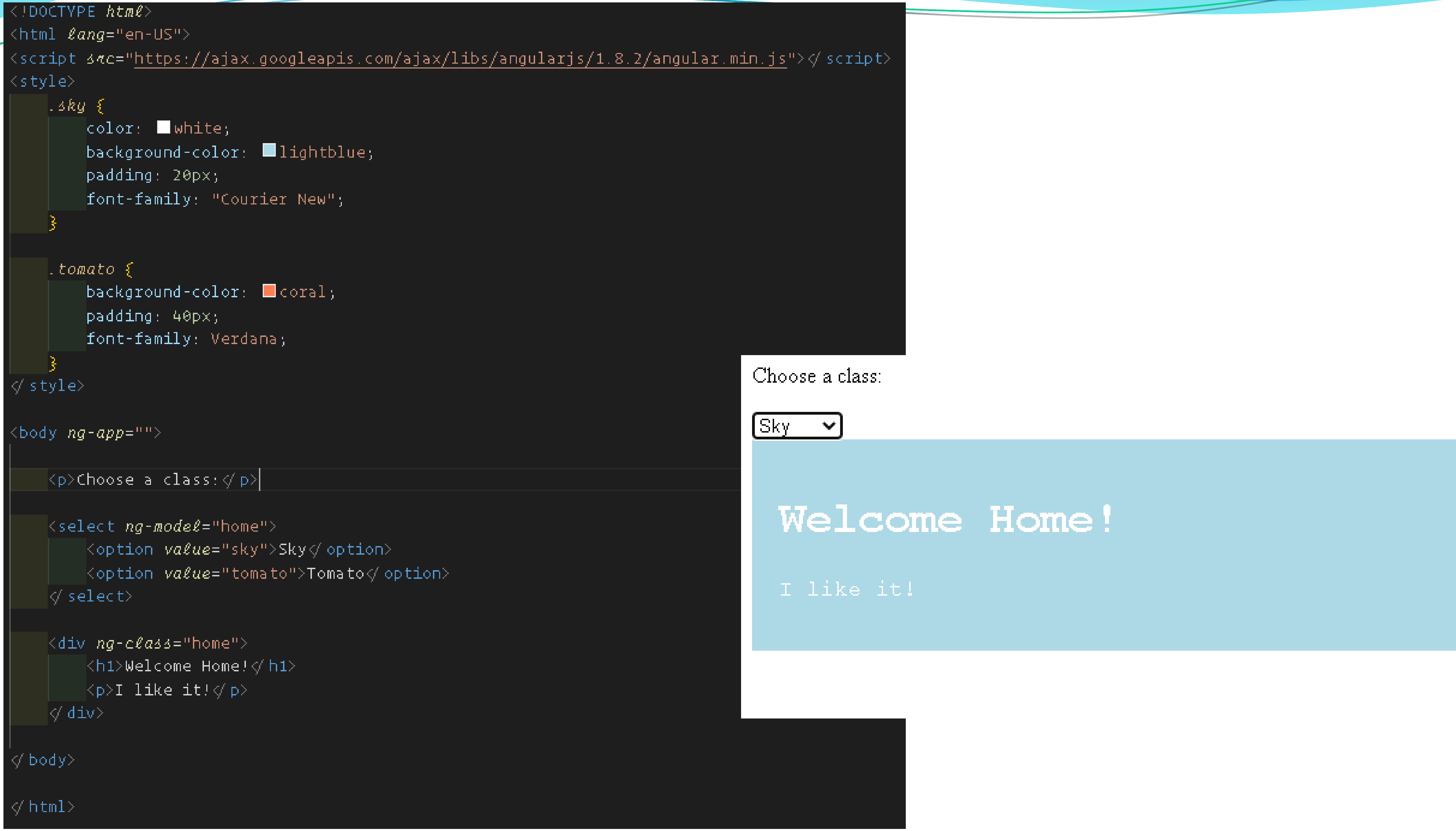
AngularJS Directives:

ng-class Directive

- The AngularJS ng-class directive facilitates you to dynamically set CSS classes on an HTML element by data binding an expression that represents all classes to be added. It may be a **String, an object or an array.**
- In case of a string, it should contain one or more, space-separated class names.
- In case of an object, it should contain key-value pairs, where the key is the class name of the class you want to add, and the value is a Boolean value.
- In the case of an array, it can be a combination of both.
- **Syntax:**

<element ng-class="expression"**></element>**

AngularJS Directives:



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<style>
    .sky {
        color: white;
        background-color: lightblue;
        padding: 20px;
        font-family: "Courier New";
    }

    .tomato {
        background-color: coral;
        padding: 40px;
        font-family: Verdana;
    }
</style>

<body ng-app="">

    <p>Choose a class:</p>

    <select ng-model="home">
        <option value="sky">Sky</option>
        <option value="tomato">Tomato</option>
    </select>

    <div ng-class="home">
        <h1>Welcome Home!</h1>
        <p>I like it!</p>
    </div>

</body>

</html>
```

Choose a class:

Sky

Welcome Home!

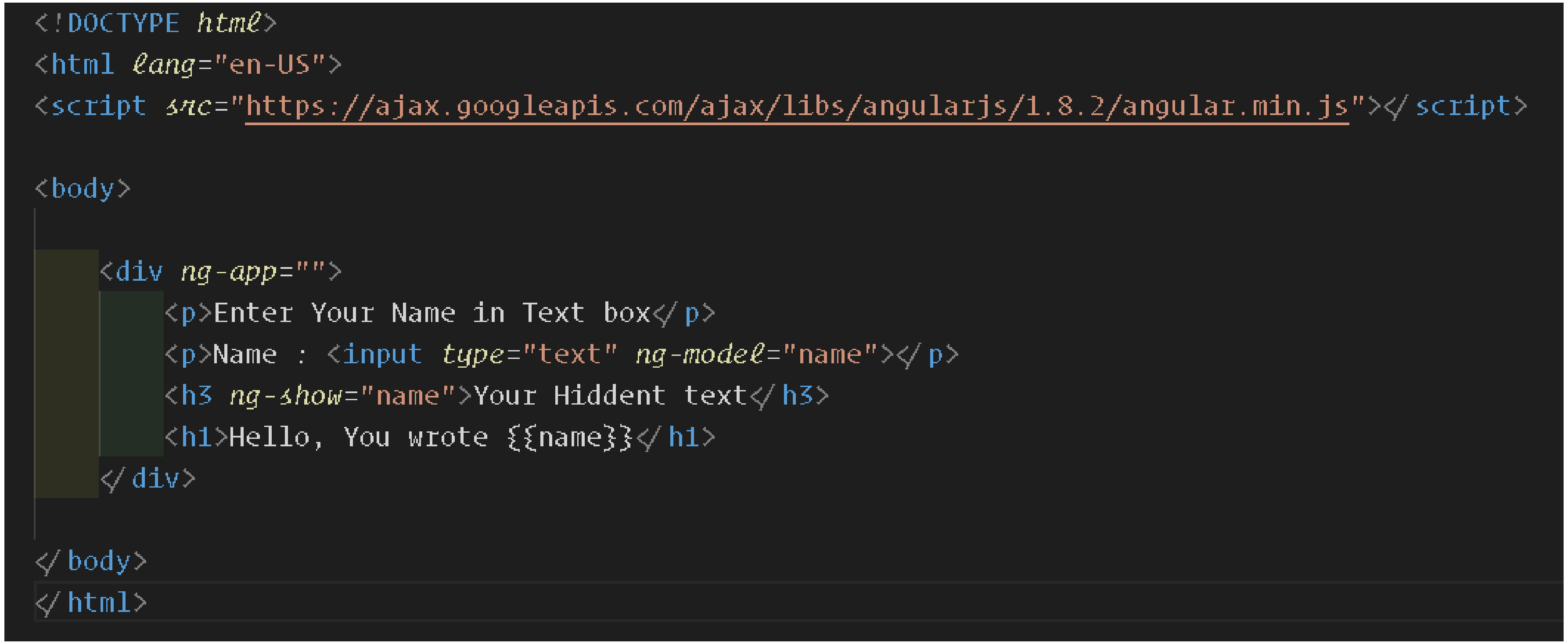
I like it!

By:Prof. Khushbu K. Surati

AngularJS Directives:

ng-show Directive

- The AngularJS ng-show directive is used to show or hide the given HTML element according to the expression given to the ng-show attribute.
- It shows the specified HTML element if the given expression is true, otherwise it hides the HTML element.



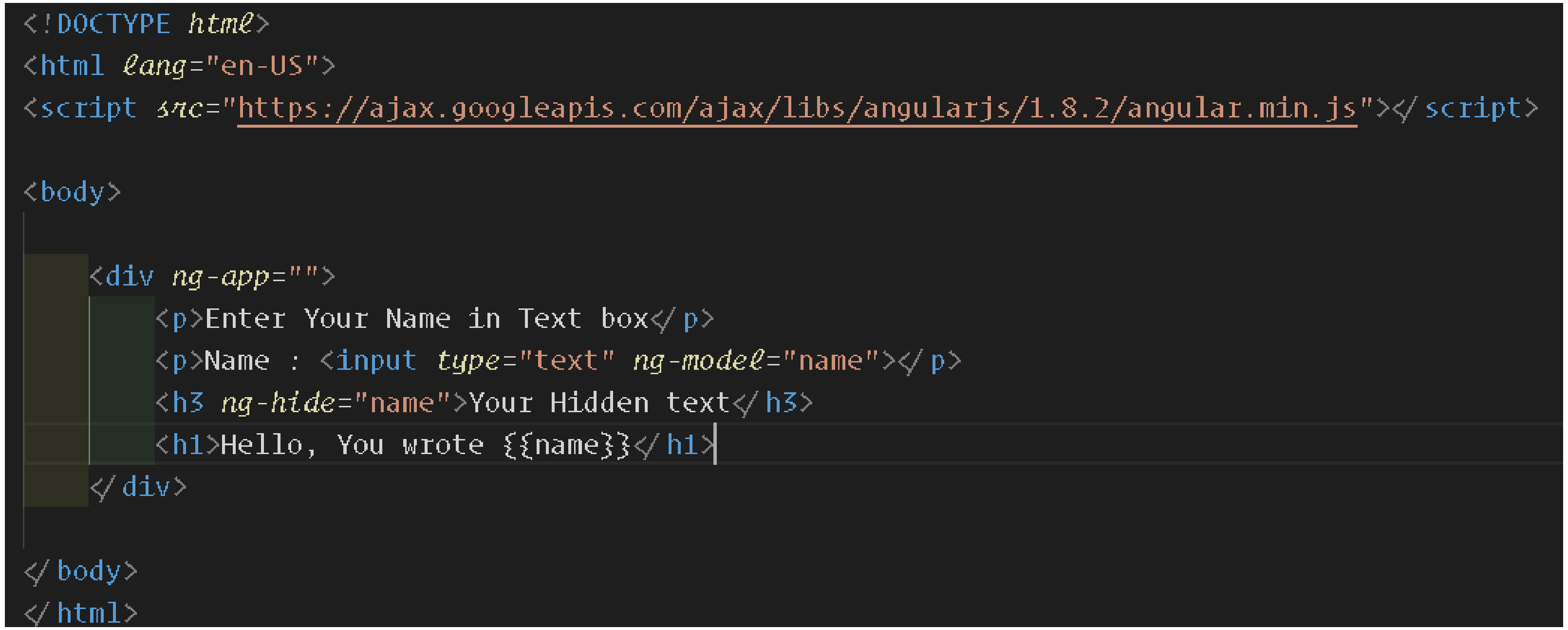
```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="">
        <p>Enter Your Name in Text box</p>
        <p>Name : <input type="text" ng-model="name"></p>
        <h3 ng-show="name">Your Hiddent text</h3>
        <h1>Hello, You wrote {{name}}</h1>
    </div>

</body>
</html>
```

AngularJS Directives:

ng-hide Directive

□ The AngularJS ng-hide directive is used to hide the HTML element if the expression is set to true.

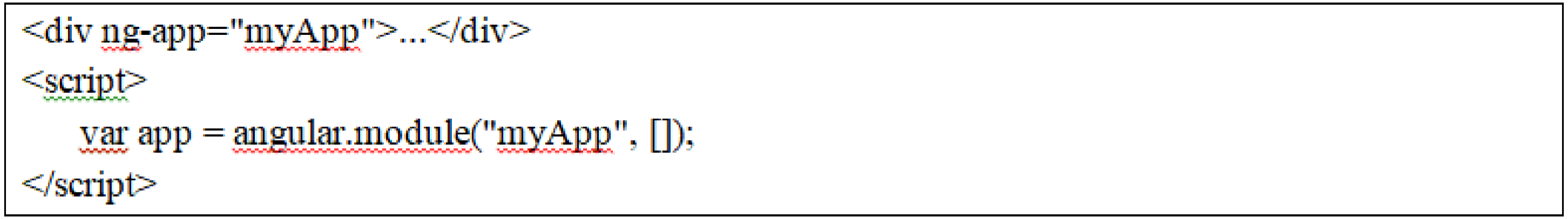
□ The element is shown if you remove the ng-hide CSS class and hidden, if you add the ng-hide CSS class onto the element. The ng-hide CSS class is predefined in AngularJS and sets the element's display to none.



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>

    <div ng-app="">
        <p>Enter Your Name in Text box</p>
        <p>Name : <input type="text" ng-model="name"></p>
        <h3 ng-hide="name">Your Hidden text</h3>
        <h1>Hello, You wrote {{name}}</h1>
    </div>

</body>
</html>
```

AngularJS Module

- In AngularJS, a module defines an application. It is a container for the different parts of your application like controller, services, filters, directives etc.
- A module is used as a Main() method. Controller always belongs to a module.
- The angular object's module() method is used to create a module.



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

- Here, "myApp" specifies an HTML element in which the application will run.

How to add controller to a module

- If you want to add a controller to your application refer to the controller with the ng-controller directive. In AngularJS applications, you can put the module and the controllers in separate JavaScript files also.

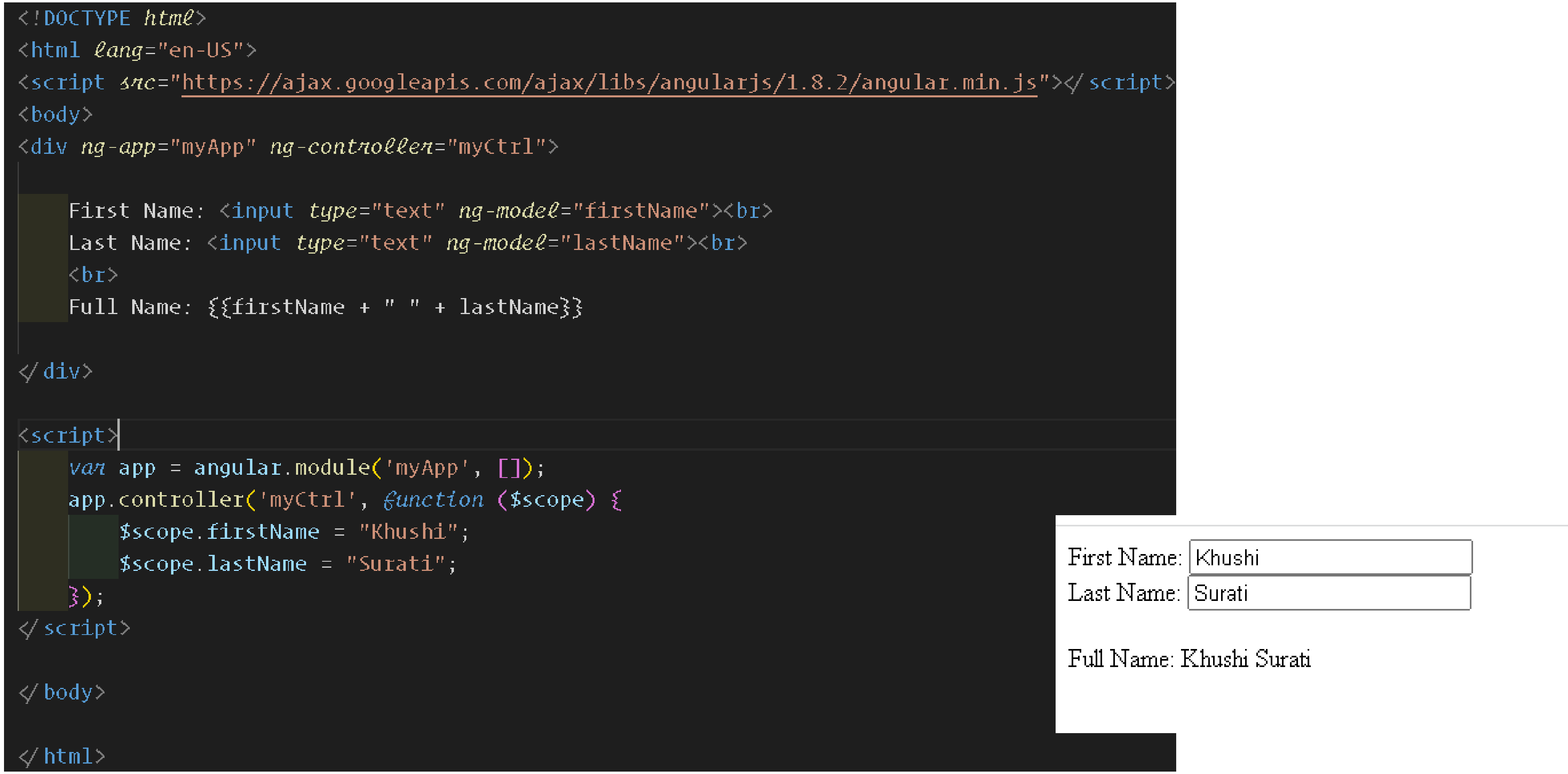
AngularJS Scopes

- The Scope is an object that is specified as a binding part between the HTML (view) and the JavaScript (controller). It plays a role of joining controller with the views. It is available for both the view and the controller.
- To make a controller in AngularJS, you have to pass the $scope object as an argument.

AngularJS Controllers

- AngularJS controllers are used to control the flow of data of AngularJS application.
- A controller is defined using ng-controller directive.
- A controller is a JavaScript object containing attributes/properties and functions.
- Each controller accepts $scope as a parameter which refers to the application/module that controller is to control.

AngularJS Controllers



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/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 = "Khushi";
        $scope.lastName = "Surati";
    });
</script>

</body>

</html>
```

```
First Name: Khushi
Last Name: Surati

Full Name: Khushi Surati
```

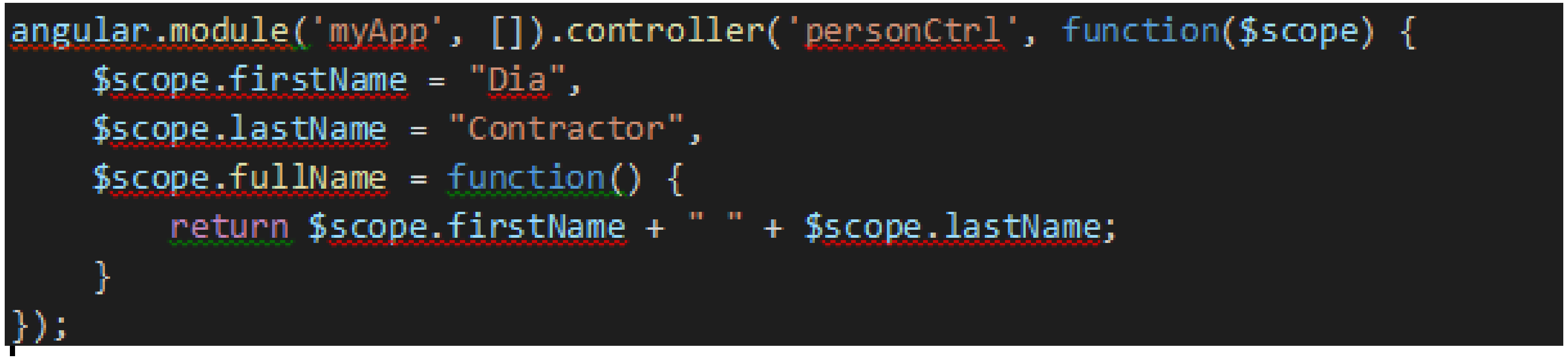
By:Prof. Khushbu K. Surati

AngularJS Controllers

- The AngularJS application runs inside the <div> is defined by ng-app="myApp".
- The AngularJS directive is ng-controller="myCtrl" attribute.
- The myCtrl function is a JavaScript function.
- AngularJS will invoke the controller with a $scope object.
- In AngularJS, $scope is the application object (the owner of application variables and functions).
- The controller creates two properties (variables) in the scope (firstName and lastName).
- The ng-model directives bind the input fields to the controller properties (firstName and lastName).

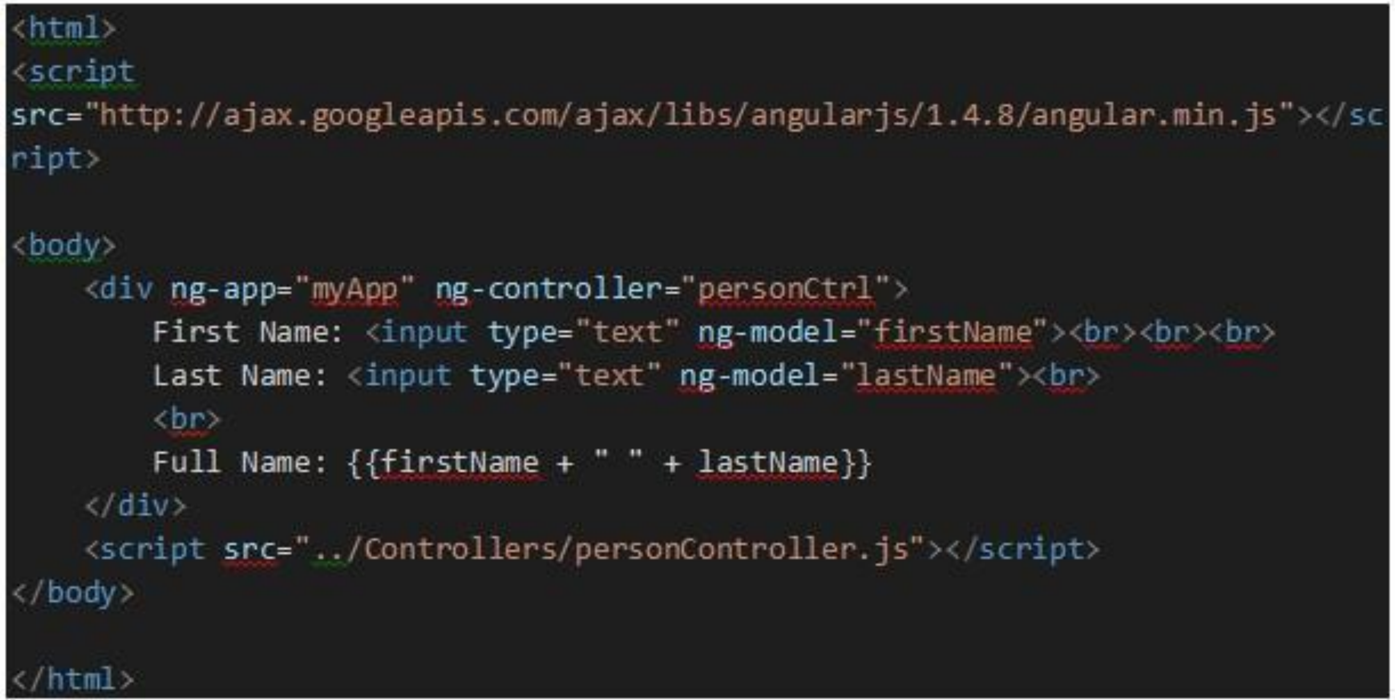
AngularJS Controllers

personController.js



```
angular.module('myApp', []).controller('personCtrl', function($scope) {
    $scope.firstName = "Dia",
    $scope.lastName = "Contractor",
    $scope.fullName = function() {
        return $scope.firstName + " " + $scope.lastName;
    }
});
```

UseController.html



```
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></sc
ript>

<body>
    <div ng-app="myApp" ng-controller="personCtrl">
        First Name: <input type="text" ng-model="firstName"><br><br><br>
        Last Name: <input type="text" ng-model="lastName"><br>
        <br>
        Full Name: {{firstName + " " + lastName}}
    </div>
    <script src="../Controllers/personController.js"></script>
</body>

</html>
```

AngularJS Filters

Filter	Description
Currency	It formats a number to a currency format.
Date	It formats a date to a specified format.
Filter	It select a subset of items from an array.
Json	It formats an object to a Json string.
Limit	It is used to limit an array/string, into a specified number of elements/characters.
Lowercase	It formats a string to lower case.
Number	It formats a number to a string.
Orderby	It orders an array by an expression.
Uppercase	It formats a string to upper case.

AngularJS Filters

How to add filters to expressions

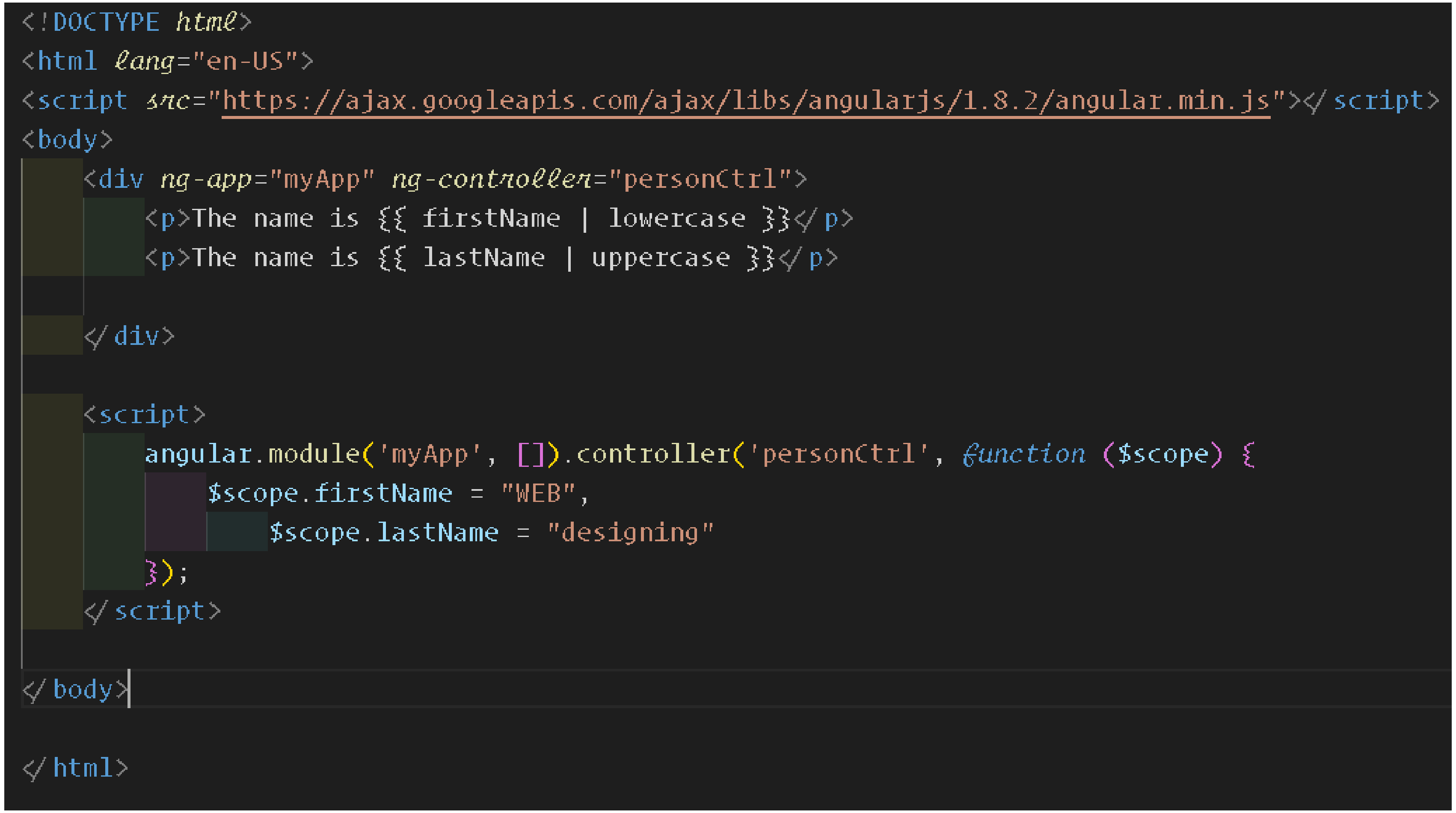
- You can add filters to expressions by using the pipe character |, followed by a filter. Filters can be applied to expressions in view templates using the following syntax:

  {{ expression | filter }}

How to add filters to directives

- Filters can be added to directives, like ng-repeat, by using the pipe character |, followed by a filter.

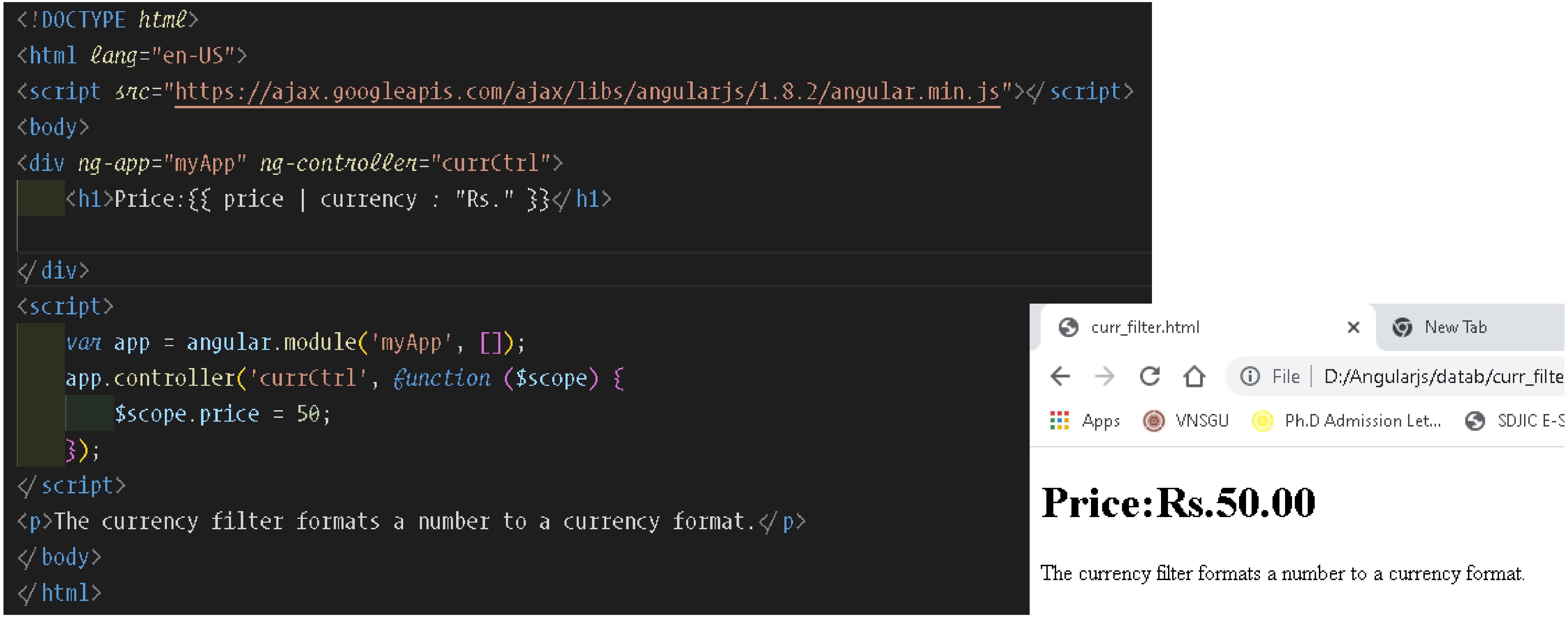
AngularJS Filter (uppercase/lowercase)



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<body>
    <div ng-app="myApp" ng-controller="personCtrl">
        <p>The name is {{ firstName | lowercase }}</p>
        <p>The name is {{ lastName | uppercase }}</p>

    </div>

    <script>
        angular.module('myApp', []).controller('personCtrl', function ($scope) {
            $scope.firstName = "WEB",
                $scope.lastName = "designing"
        });
    </script>

</body>

</html>
```

By:Prof. Khushbu K. Surati

Currency filter

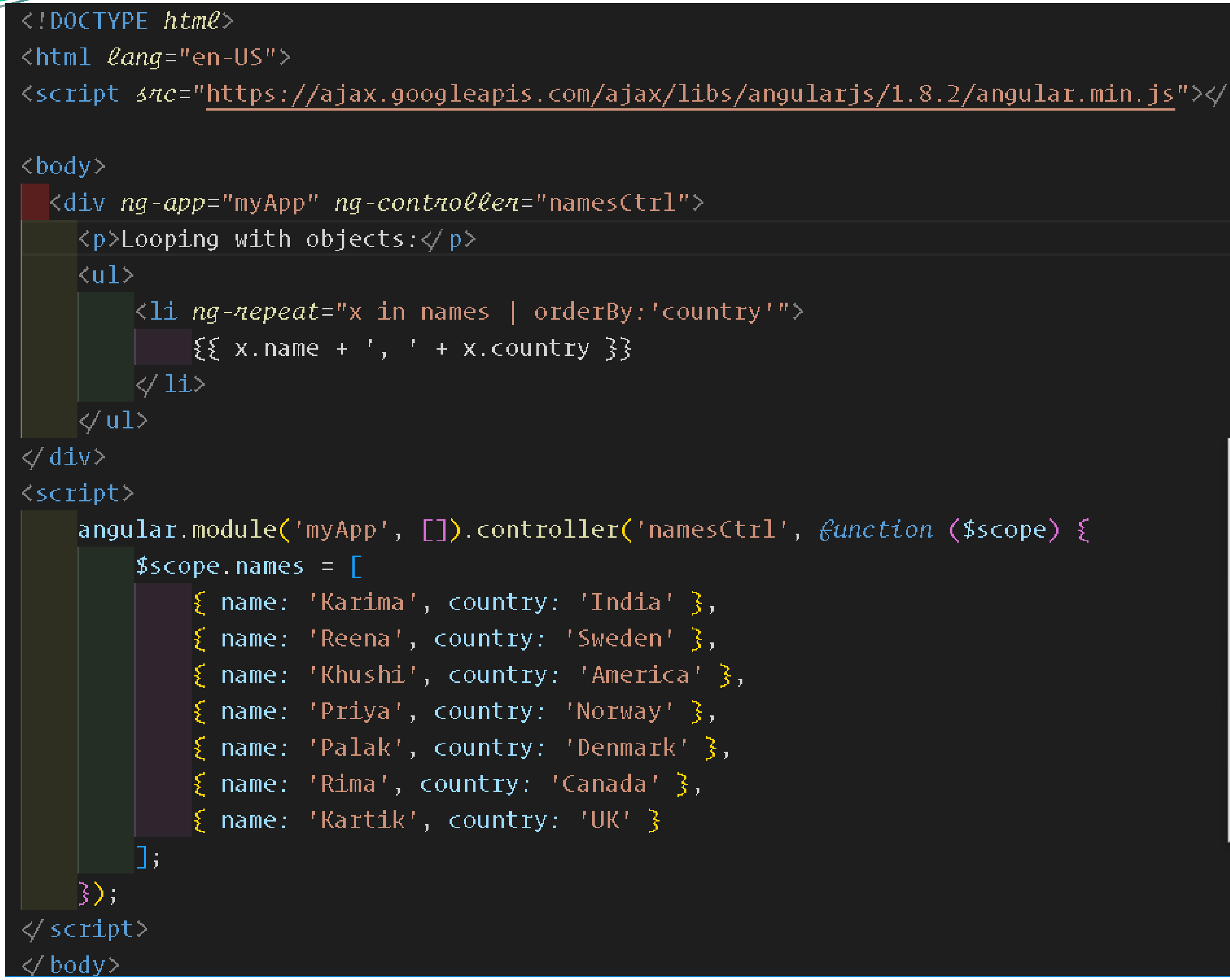


```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
<body>
<div ng-app="myApp" ng-controller="currCtrl">
    <h1>Price:{{ price | currency : "Rs." }}</h1>

</div>
<script>
    var app = angular.module('myApp', []);
    app.controller('currCtrl', function ($scope) {
        $scope.price = 50;
    });
</script>
<p>The currency filter formats a number to a currency format.</p>
</body>
</html>
```

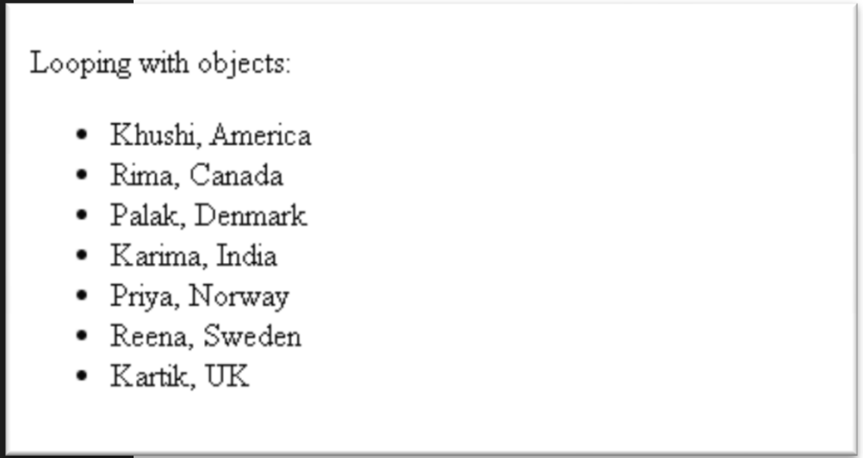
Price:Rs.50.00

The currency filter formats a number to a currency format.

AngularJS Filter (orderBy)



```
<!DOCTYPE html>
<html lang="en-US">
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>

<body>
<div ng-app="myApp" ng-controller="namesCtrl">
    <p>Looping with objects:</p>
    <ul>
        <li ng-repeat="x in names | orderBy:'country'">
            {{ x.name + ', ' + x.country }}
        </li>
    </ul>
</div>
<script>
    angular.module('myApp', []).controller('namesCtrl', function ($scope) {
        $scope.names = [
            { name: 'Karima', country: 'India' },
            { name: 'Reena', country: 'Sweden' },
            { name: 'Khushi', country: 'America' },
            { name: 'Priya', country: 'Norway' },
            { name: 'Palak', country: 'Denmark' },
            { name: 'Rima', country: 'Canada' },
            { name: 'Kartik', country: 'UK' }
        ];
    });
</script>
</body>
```



```
Looping with objects:

  • Khushi, America
  • Rima, Canada
  • Palak, Denmark
  • Karima, India
  • Priya, Norway
  • Reena, Sweden
  • Kartik, UK
```