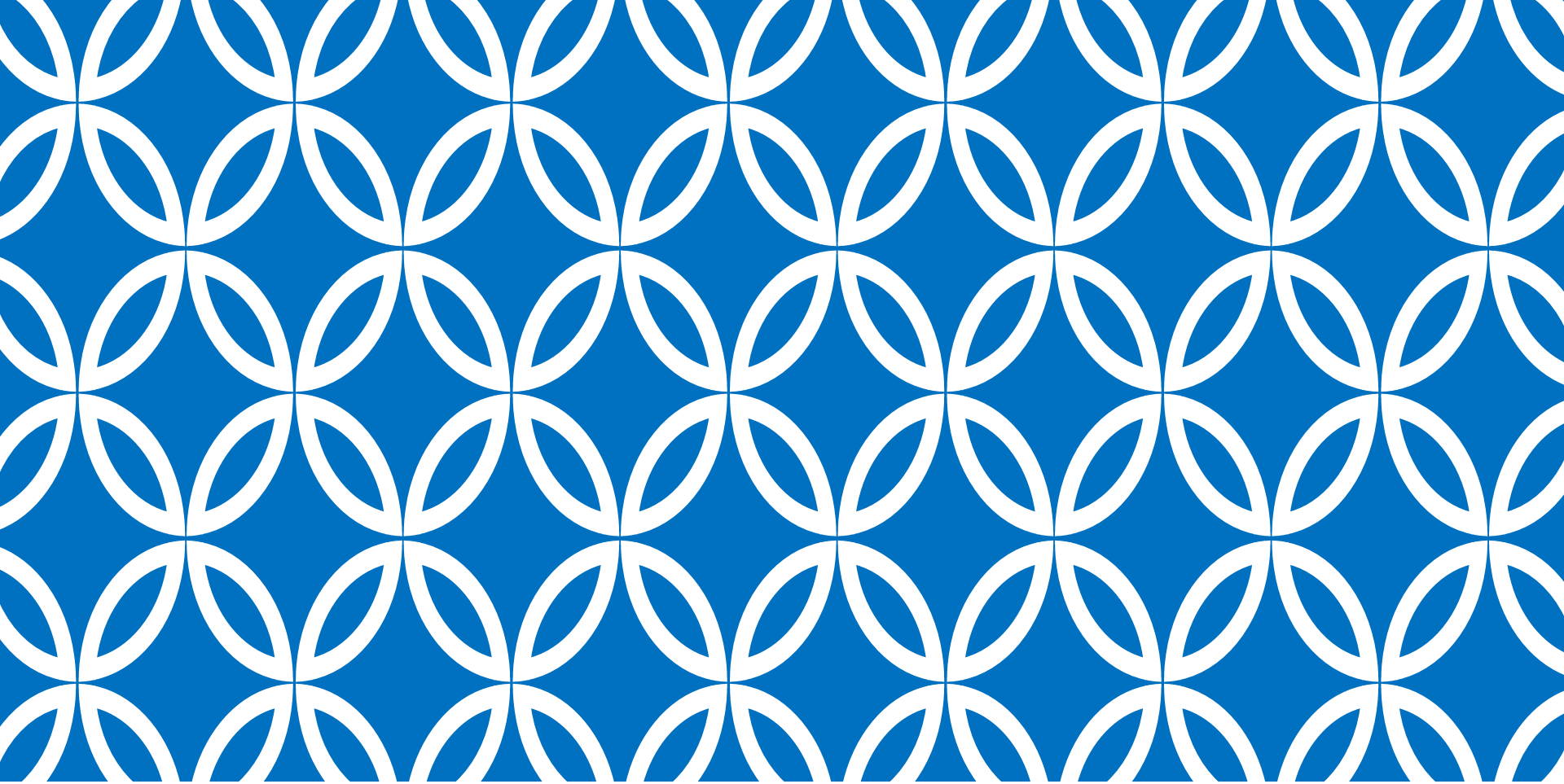
LEARN. DO. EARN

ACADGILD

FRONT END DEVELOPMENT (WITH ANGULARJS)

Website : http://www.acadgild.com
LinkedIn : https://www.linkedin.com/company/acadgild
Facebook : https://www.facebook.com/acadgild



Session 12 – AngularJS Introduction

Agenda – AngularJS Introduction

Agenda	Agenda Title
1	Introduction to AngularJS
2	Why Use it?
3	AngularJS Features
4	Getting Started with AngularJS
5	AngularJS Expression
6	Displaying Object Values Within Expression
7	AngularJS is MVC
8	MVC

Agenda	Agenda Title
9	Basic Concepts
10	1 way Data Binding
11	2 way Data Binding
12	View Controller Scope
13	Creating View and Controller
14	What is Angular Module?
15	AngularJS Directive

Introduction to AngularJS

- Open-source **JavaScript framework**.
- It is a library written in JavaScript.
- It extends HTML attributes with **Directives** and binds data to HTML with **Expressions**.
- It is used in Single Page Applications.
- Developed in 2009 by **MiskoHevery** and **Adam Abrons**.
- Maintained by Google.

Why Use It ?

- **The problem** – HTML is great for static pages, but has no tools for web applications.

- **The solution** – extend and adapt HTML vocabulary with some additional declarations that are useful for web applications.

AngularJS Features

- Declarative HTML approach
- Easy Data Binding : Two way Data Binding
- Reusable Components
- MVC(Model View Controller)/ MVVM(Model-View-ViewModel) Design Pattern
- Dependency Injection
- End to end Integration Testing / Unit Testing

Getting Started with AngularJS

- Download AngularJS from https://angularjs.org.
- **You can also use package management tools like bower** to set up AngularJS from command line.
- **Once downloaded, just include AngularJS within the script tag**.
- Then we need to decide the boundary of our application. That is done by simply adding an **attribute ng- app** to any tag.
- AngularJS will operate within the starting and ending of that tag that contains **ng-app**.
- Now to test it, just write {{1+5}} and AngularJS will evaluate the expression and show the output.

Angular JS Expression

- Expressions are JavaScript-like code snippets that are usually placed in bindings such as *{{ expression }}*
- AngularJS expressions doesn't support control flow statements(conditionals, loops, or exceptions). These supports filters to format data before displaying it.
- There are some valid AngularJS expressions:
  - {{ 1 + 2 }}
  - {{ x + y }}
  - {{ x == y }}
  - {{ x = 2 }}
  - {{ user.Id }}
  - {{ items[index] }}

Displaying Object Values Within Expression

- To display an object value within AngularJS Expression we need to initialize the value.
- Use custom AngularJS attribute called *ng-init*.

Syntax:

```
<div ng-init = "employee = {name:'smith',age:27}">
<p> The employee name and age are {{employee.name}}
and {{employee.age}}</p>
</div>
```

AngularJS is MVC

- MVC = Model-View-Controller
- Is a software design pattern for developing web applications
- It isolates the application logic from the user interface layer and supports separation of concerns
- Less dependencies
- Improves maintainability
- It is easier to read and understand code

MVC

Model

- Holds the data
- Notifies the View and the Controller for changes in the data
- Does not know about the View and the Controller

View

- Displays stuff (buttons, labels, …)
- This is what your users will see
- Knows about the Model
- Usually displays something related to the current state of the Model

Controller

- Controls everything
- Knows about both the Model and the View
- The “logic” resides in it. What to happen, when and how

Basic Concepts

- **Directives –** AngularJS directives are used to extend the HTML vocabulary i.e. they decorate html elements with new behaviors and help to manipulate html elements attributes in interesting

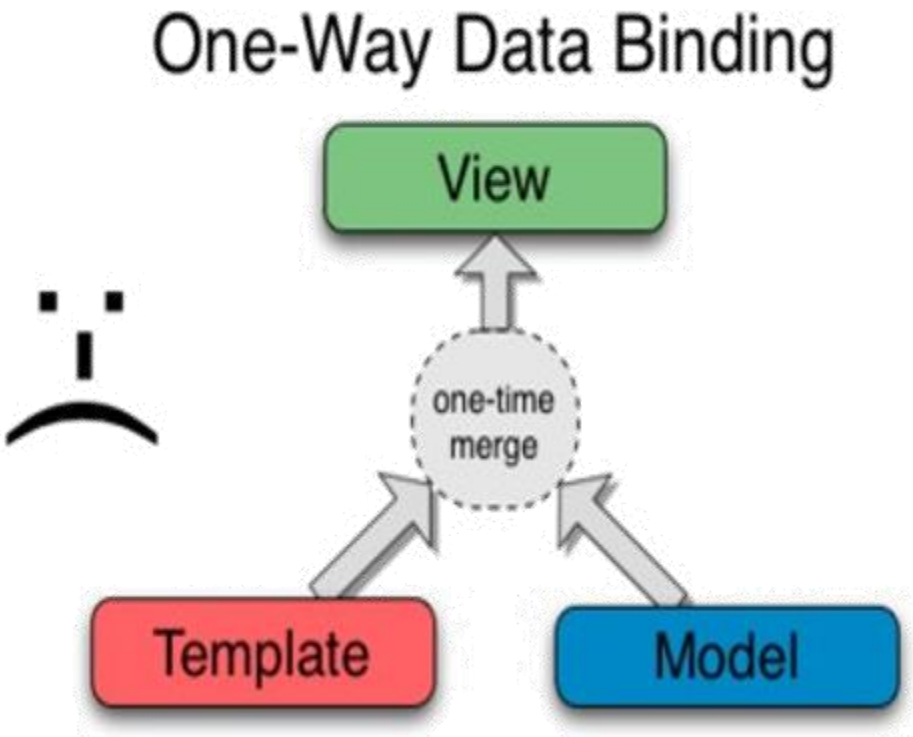
- **Data Binding –** Bind model to view using expressions **{{ }}**

- **Scope -** Scope is an object that refers to the application model. It acts as a context for evaluating expressions.

1 Way Data Binding

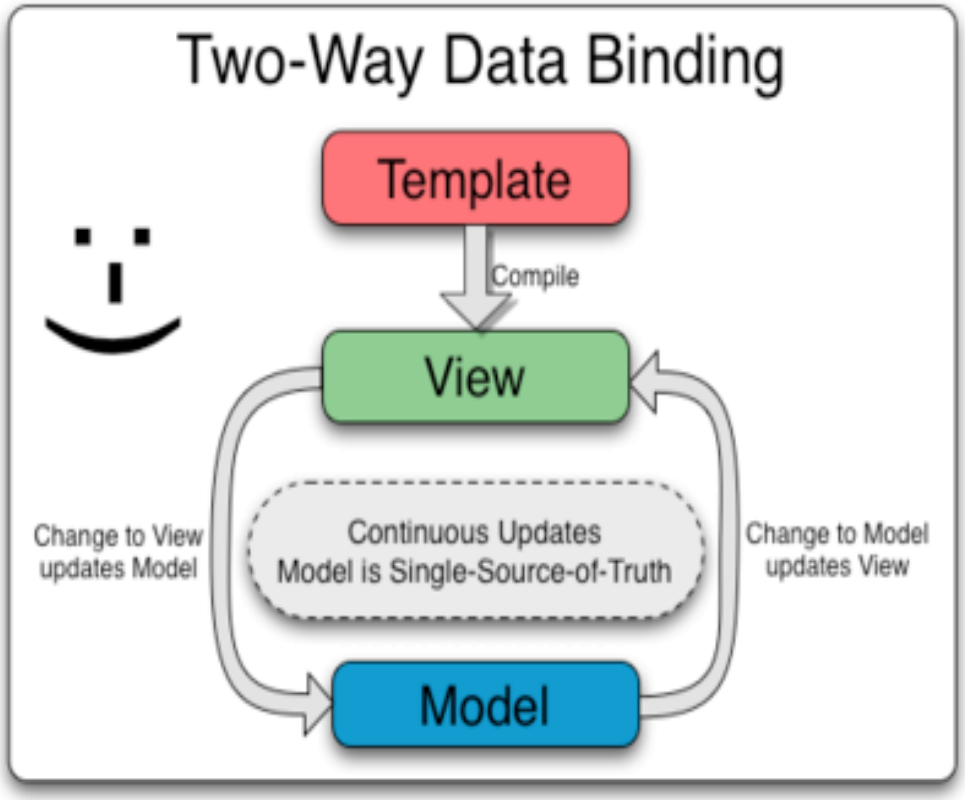
Data is bound to the view once during rendering.

Next time if there is any change in the data, we have to again refresh the view to reflect the data change.



2 Way Data Binding

- Automatic propagation of data changes
- Model is single source of truth
- Digest cycle



View Controller Scope

- **$scope** is the *"glue"* (ViewModel) between a controller and a view

Creating a View and Controller

```
<div class="container" data-ng-controller="SimpleController">
   <h3>Adding a Simple Controller</h3>
   <ul>
      <li data-ng-repeat="cust in customers">
        {{ cust.name }} - {{ cust.city }}
      </li>
   </ul>
</div>

<script>
   function SimpleController($scope) {
      $scope.customers = [
        { name: 'Dave Jones', city: 'Phoenix' },
        { name: 'Jamie Riley', city: 'Atlanta' },
        { name: 'Heedy Wahlin', city: 'Chandler' },
        { name: 'Thomas Winter', city: 'Seattle' }
      ];
   }
</script>
```

Define the controller to use

Access $scope

$scope injected dynamically

Basic controller

What is Angular Module?

- **Container for**
  - Controllers
  - Services
  - Directives
  - Factories
  - Filters
  - Configuration information

Note : Each Angular JS app contains at least one module

AngularJs Directive

AngularJS directives are extended HTML attributes with the prefix **ng-**.

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

Agenda – AngularJS Directives

1. Inbuilt Directives
2. AngularJS Template
3. Sample Template
4. Understanding Scopes
5. Controllers Scopes
6. Scopes Hierarchy
7. $parent Keyword
8. this Keyword

Inbuilt Directives

- Following are the commonly used directives:
- **ng-click:** To attach a click event.
- **ng-change:** Fires a function when user changes input
- **ng-init:** to initialize data
- **ng-show:** Shows or hides the given HTML element based on the expression provided to the ngHide attribute
- **ng-href:** To point a URL for a link

Inbuilt Directives

- **ng-include:** This directive fetches, compiles and includes an external HTML snippet.
- **ng-model:** to bind a form input .
- **ng-repeat:** This directive is used to repeat the html for a particular collection. Useful to show an array items in a list etc.
- **ng-selected:** If the expression value is true, then special attribute "selected" will be set on the specified element.
- **ng-show:** It shows or hides the given HTML element based on the expression provided to the ng-show attribute.

AngularJS Template

- AngularJS templates are just plain old HTML that contains Angular-specific elements and attributes.

- AngularJS used these templates to show information from the model and controller.

- Inside your AngularJS templates you can define following angular elements and attributes:
  - Directive
  - Angular Markup ({{ }})
  - Filters
  - Form controls

Sample Template

Template

```
<!DOCTYPE html>
<html>
  <head>
    <title>
      Title
    </title>
    <meta charset="UTF-8" />
    <style media="screen"></style>
    <script src="angular.min.js"></script>
  </head>
  <body>
    <!-- initialize the app -->
    <div ng-app>
      <!-- store the value of input field into a variable name -->
      <p>Name: <input type="text" ng-model="name"></p>
      <!-- display the variable name inside (innerHTML) of p -->
      <p ng-bind="name"></p>
    </div>
  </body>
</html>
```

Download this file from:
https://angularjs.org/

Directive

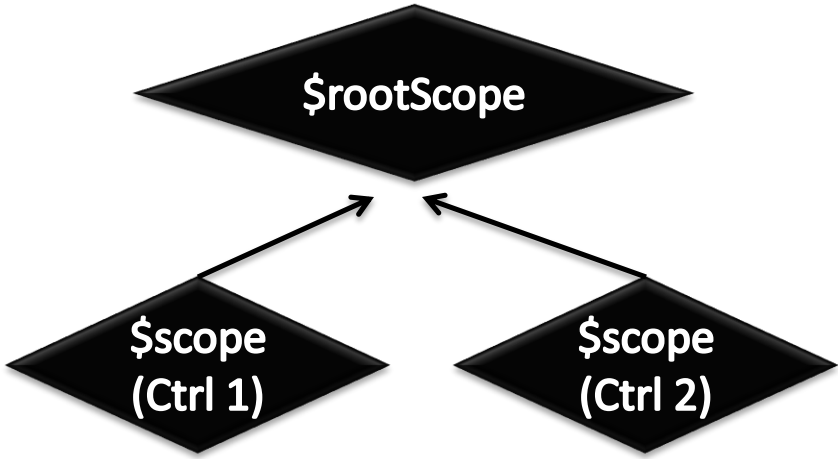
Directive

Understanding Scopes

- Scope is an object that refers to the application model.
- It acts as a context for evaluating expressions.
- Typically, it acts as a glue between controller and view.
- Scopes are hierarchical in nature and follow the DOM structure of your angular app.
- AngularJS has two scope objects: **$rootScope** and **$scope**.

Controllers Scopes

- A $scope is a JavaScript object which is used for communication between controller and view.
- Basically, $scope binds a view (DOM element) to the view model and functions defined in a controller.
- The **$rootScope** is the top-most scope.
- An app can have only **one $rootScope** which will be shared among all the components of an app. Hence it acts like a global variable.
- All other $scopes are children of the $rootScope.



Scopes Hierarchy

- **What happens if we keep one controller within another controller?**
  - The controller that has other controller inside becomes the parent controller. Now the child controller will inherit all the properties of the parent controller.
  - Any properties of the parent controller can be over written.

$parent Keyword

- Angular scopes include a variable called **$parent** that refers to the parent scope of a controller.

- If a controller is at the root of the application, then parent would be the root scopei.e. $rootScope

- There are instances when we have overwritten the property of the parent controller within child controller and we still want to access or modify it. In this case we can use the $parent keyword.

this Keyword

- We can use $scope injectible to give access to the data within the html. Though this works fine, we can also use **this** operator of the controller instance to achieve the same.

- The benefit of using **this** operator is that we don't need to use the $scope to attach data to the scope.

Lets Discuss Assignments