Array of Objects

```
<!DOCTYPE html> <html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/an
gular.min.js"></script><body>
<div ng-app="" ng-init="names=[
{name:'Marees',country:'India'},
{name:'Hema',country:'America'},
{name:'Kumar',country:'Denmark'}]">
<p>Looping with objects:</p>
<ul>
 <li ng-repeat="x in names">
 {{ x.name + ', ' + x.country }}</li>
</ul></div></body></html>
```

Looping with objects:

- Marees, India
- Hema, America
- Kumar, Denmark

AngularJS Modules

- An AngularJS module defines an application.
- The module is a container for the different parts of an application.
- The module is a container for the application controllers.
- Controllers always belong to a module.
- **Creating a Module**
  - A module is created by using the AngularJS function ***angular.module***
  - …

    <script>

    var app = **angular.module("myApp", []);**

    </script>
  - The "myApp" parameter refers to an HTML element in which the application will run.
  - Now you can add controllers, directives, filters, and more, to your AngularJS application.

AngularJS 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**.

Step1 : Creating a Module

- …
  <script>var app = angular.module("myApp", []); </script>
- The "myApp" parameter refers to an HTML element in which the application will run.
- Now you can add controllers, directives, filters, and more, to your AngularJS application.

Step2 : Adding a Controller

- Add a controller to your application, and refer to the controller with the **ng-controller** directive:

```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/ang
ular.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>
        {{ firstName + " " + lastName }}
</div><script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
   $scope.firstName = “Vijayan";
   $scope.lastName = “R";
});</script></body></html>
```

Application explained:

- The AngularJS application is defined by **ng-app="myApp"**. The application runs inside the .
- The **ng-controller="myCtrl"** attribute is an AngularJS directive. It defines a controller.
- 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).

```
<!doctype html><html><head>
<script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js
"></script></head>
<body ng-app = "myapp">
<div ng-controller = "HelloController" >
<h2>Welcome {{helloTo.title}} to the world of Web
Technology!</h2>
</div><script>
angular.module("myapp", []).controller("HelloController",
function($scope) {
        $scope.helloTo = {};
        $scope.helloTo.title = "AngularJS";
      });</script></body></html>
```

Include AngularJS

Point to AngularJS app

View

Controller

- **Include AngularJS**

included the AngularJS JavaScript file in the HTML page so we can use AngularJS

- **Point to AngularJS app**

tell what part of the HTML contains the AngularJS app either add it to *html* element or *body* element

- **View**

ng-controller tells AngularJS what controller to use with this view. *helloTo.title* tells AngularJS to write the "model" value named helloTo.title to the HTML at this location.

- **Controller**

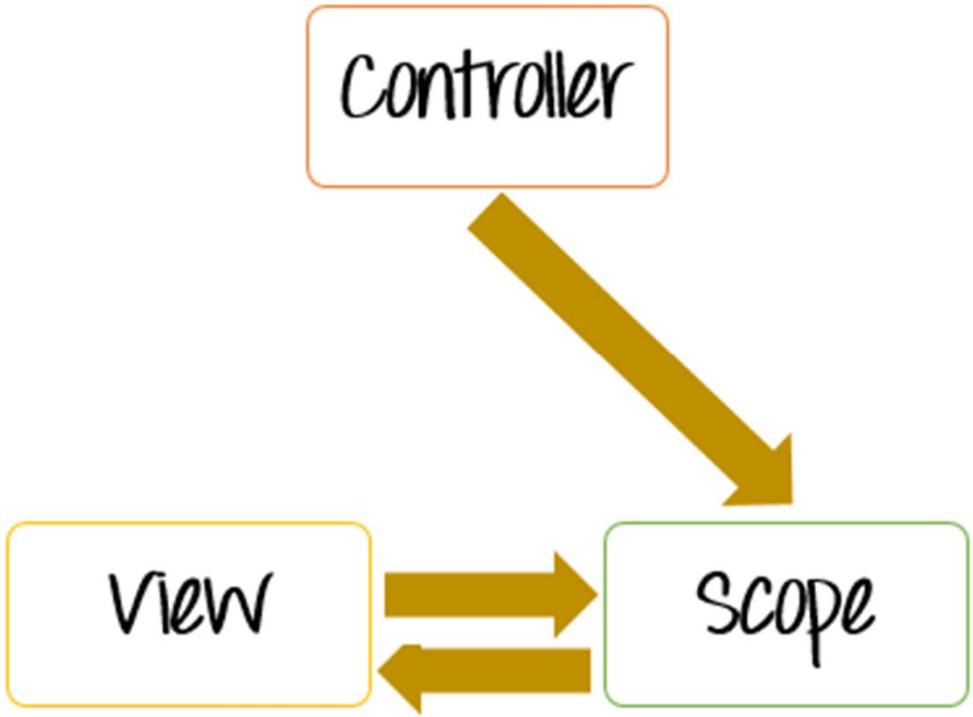
code registers a controller function named *HelloController* in the angular module named *myapp*.

The $scope parameter passed to the controller function is the *model*. The controller function adds a *helloTo* JavaScript object, and in that object it adds a *title* field.

When the page is loaded in the browser, following things happen:

- HTML document is loaded into the browser, and evaluated by the browser. AngularJS JavaScript file is loaded, the angular *global* object is created. Next, JavaScript which registers controller functions is executed.
- Next AngularJS scans through the HTML to look for AngularJS apps and views. Once view is located, it connects that view to the corresponding controller function.
- Next, AngularJS executes the controller functions. It then renders the views with data from the model populated by the controller. The page is now ready.

- The controller's primary responsibility is to control the data which gets passed to the view. The scope and the view have two-way communication.
- The properties of the view can call "functions" on the scope. Moreover events on the view can call "methods" on the scope.



AngularJS Routing

- The ngRoute module helps your application to become a Single Page Application.
- If you want to navigate to different pages in your application, but you also want the application to be a SPA (Single Page Application), with no page reloading, you can use the ngRoute module.
- The ngRoute module *routes* your application to different pages without reloading the entire application.
- Angularrouting.html & angularrouting2.html

Ng-view

- Your application needs a container to put the content provided by the routing.
- This container is the ng-view directive.
- There are three different ways to include the ng-view directive in your application:

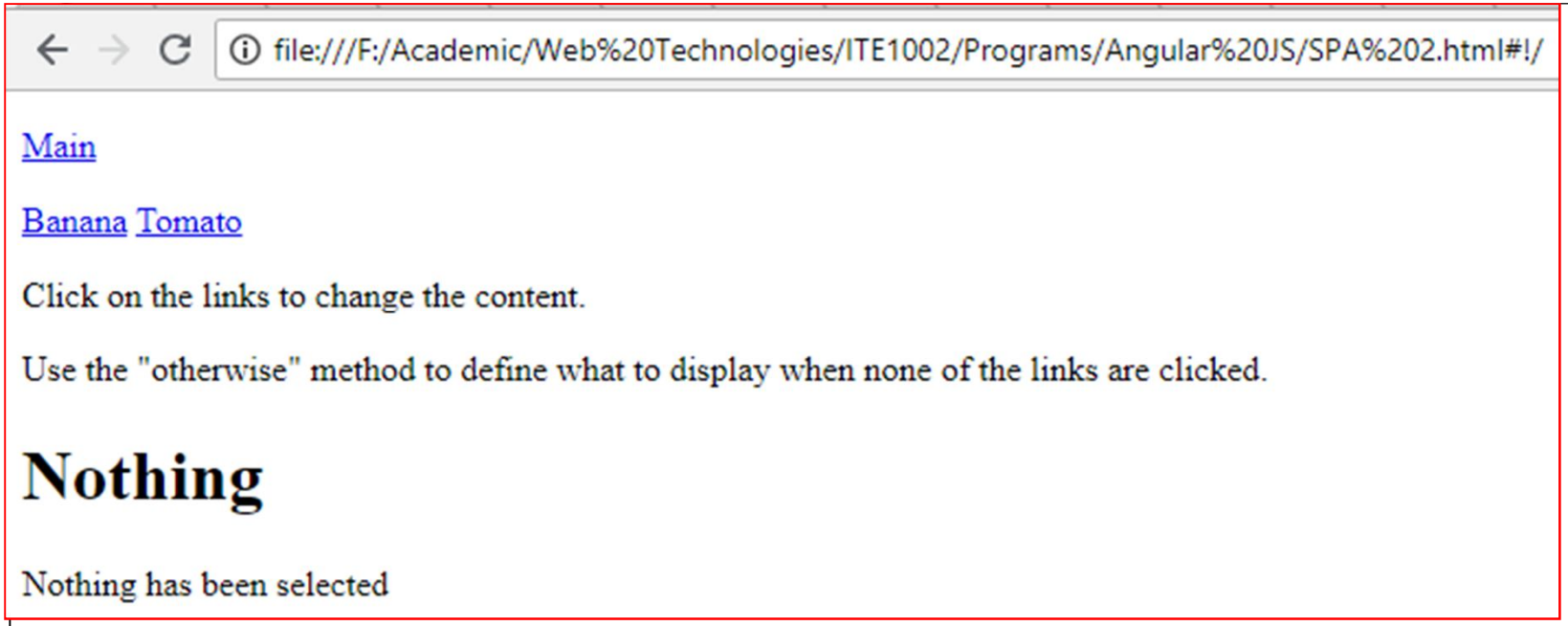
- 1. <div ng-view>
- 2. <ng-view></ng-view>
- 3.

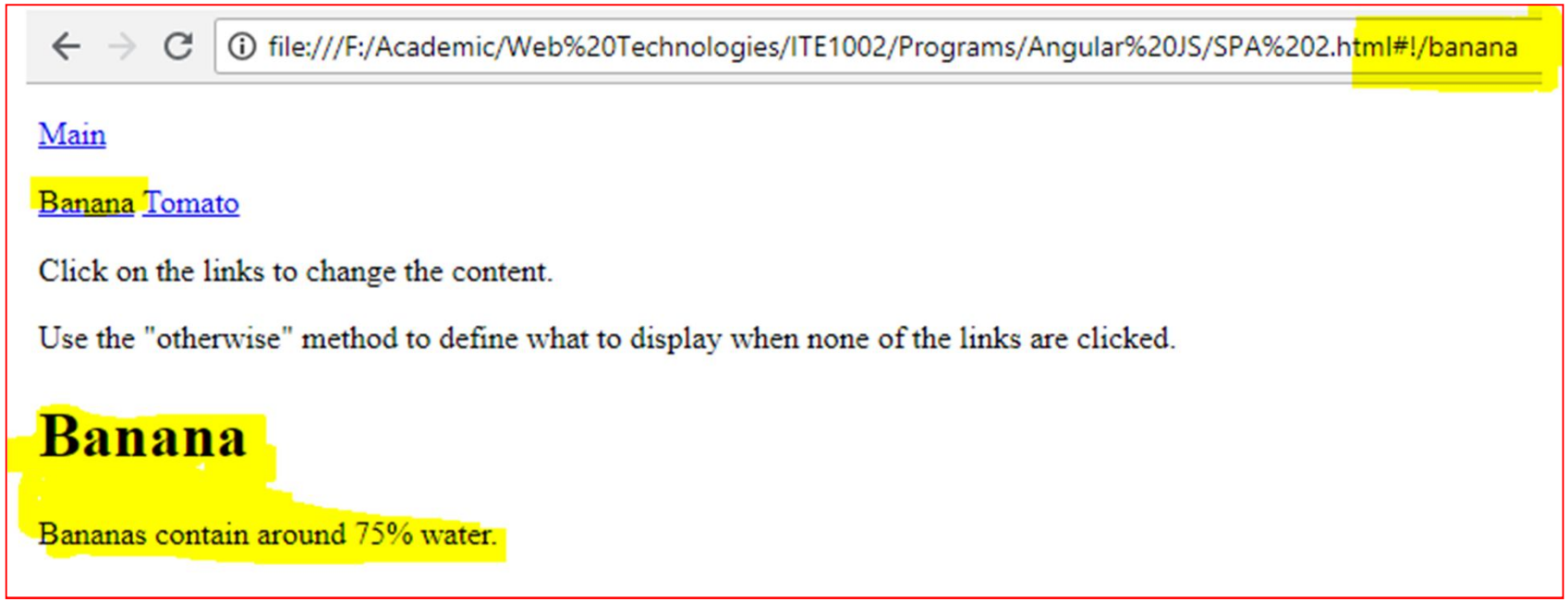
- Applications can only have one ng-view directive, and this will be the placeholder for all views provided by the route.

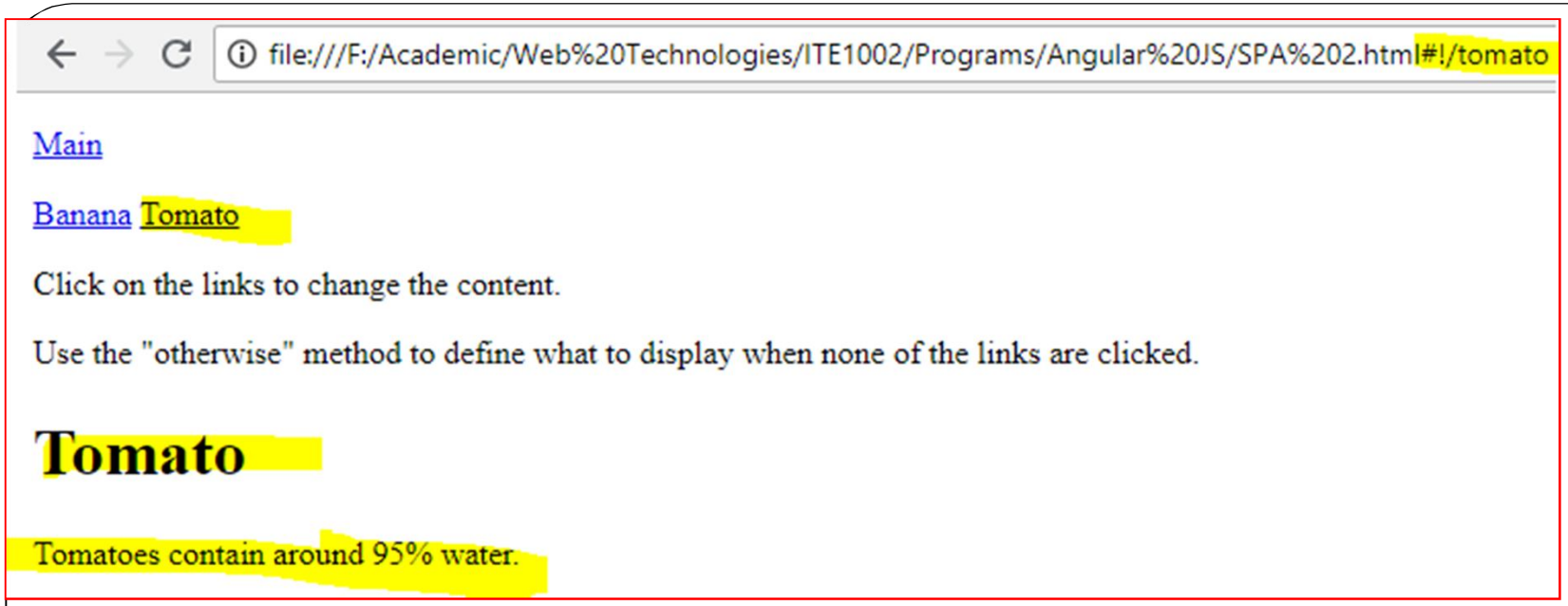
Single Page Application

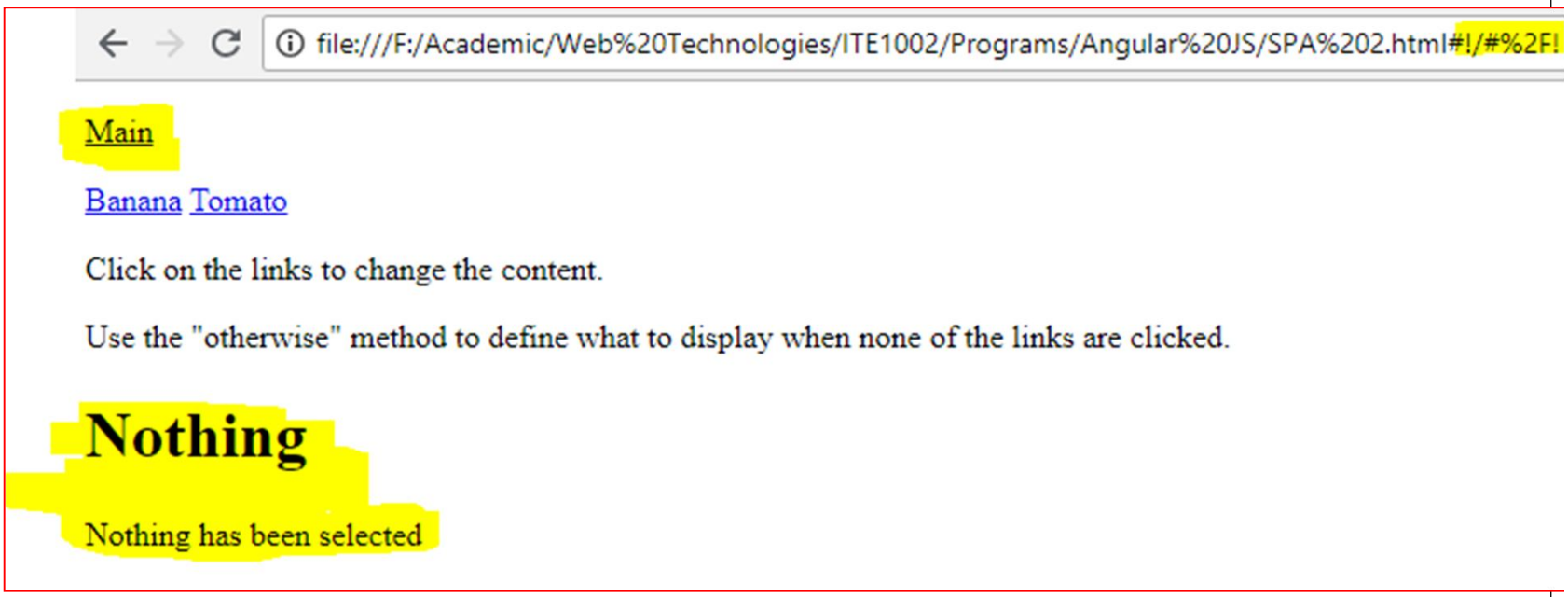
```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"
> </script>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-
route.js"></script>
<body ng-app="myApp">
<p><a href="#/!">Main</a></p>
<a href="#!banana">Banana</a>
<a href="#!tomato">Tomato</a>
<p>Click on the links to change the content.</p>
<p>Use the "otherwise" method to define what to display when none of the
links are clicked.</p>
```

```
<div ng-view></div><script>
var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {
    $routeProvider
    .when("/banana", {
        template: "<h1>Banana</h1><p>Bananas contain around
75% water.</p>"    })
    .when("/tomato", {
        template : "<h1>Tomato</h1><p>Tomatoes contain around
95% water.</p>"    })
    .otherwise({
        template : "<h1>Nothing</h1><p>Nothing has been
selected</p>"
    });});</script></body></html>
```


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/
Main
Banana Tomato
Click on the links to change the content.
Use the "otherwise" method to define what to display when none of the links are clicked.
Nothing
Nothing has been selected


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/banana
Main
Banana Tomato
Click on the links to change the content.
Use the "otherwise" method to define what to display when none of the links are clicked.
Banana
Bananas contain around 75% water.


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/tomato
Main
Banana Tomato
Click on the links to change the content.
Use the "otherwise" method to define what to display when none of the links are clicked.
Tomato
Tomatoes contain around 95% water.


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/SPA%202.html#!/#%2F!
Main
Banana Tomato
Click on the links to change the content.
Use the "otherwise" method to define what to display when none of the links are clicked.
Nothing
Nothing has been selected

```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/ang
ular.min.js"></script>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/ang
ular-route.js"></script>
<body ng-app="myApp">
<p><a href="#/!">Main</a></p>
<a href="#!Home">Home</a>
<a href="#!Blog">Blog</a>
<a href="#!About">About</a>
<div ng-view></div>
```

```
<script>var app = angular.module("myApp", ["ngRoute"]);
app.config(function($routeProvider) {    $routeProvider
    .when("/", {
        templateUrl : "SPA1.html"    })
    .when("/Home", {
        templateUrl : "SPA1_Home.html"    })
    .when("/Blog", {
        templateUrl : "SPA1_Blog.html"    })
    .when("/About", {
        templateUrl : "SPA1_About.html"
    });});</script><p>Click on the links to navigate to "Home
page", "Blog page", "About", or back to "main
page"</p></body></html>
```

AngularJS Events

- You can add AngularJS event listeners to your HTML elements by using one or more of these directives:
- ng-blur, ng-change, ng-focus
- ng-click, ng-dblclick
- ng-copy, ng-cut, ng-paste
- ng-keydown, ng-keypress, ng-keyup
- **Mouse Events :** Mouse events occur when the cursor moves over an element, in this order:

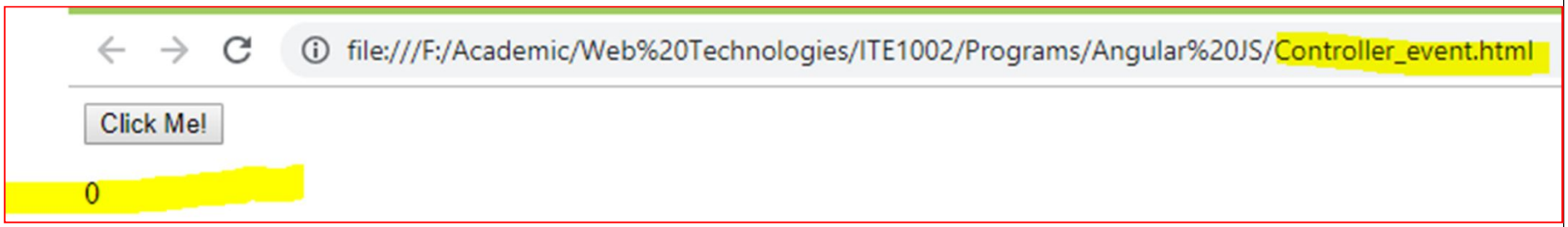
1.ng-mouseover 2. ng-mouseenter 3. ng-mousemove 4. ng-mouseleave

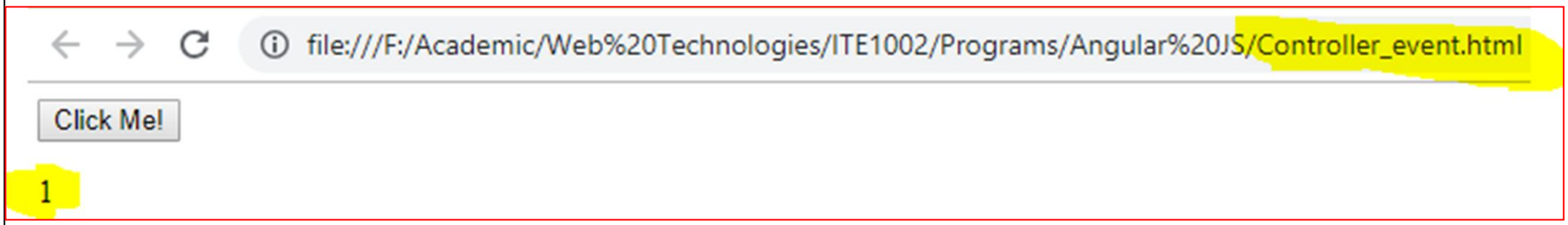
- Or when a mouse button is clicked on an element, in this order:

1. ng-mousedown 2. ng-mouseup 3. ng-click

- You can add mouse events on any HTML element.

```
<!DOCTYPE html><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">
<button ng-click="count = count + 1">Click Me!</button>
<p>{{ count }}</p>
</div><script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
   $scope.count = 0;
});</script></body></html>
```

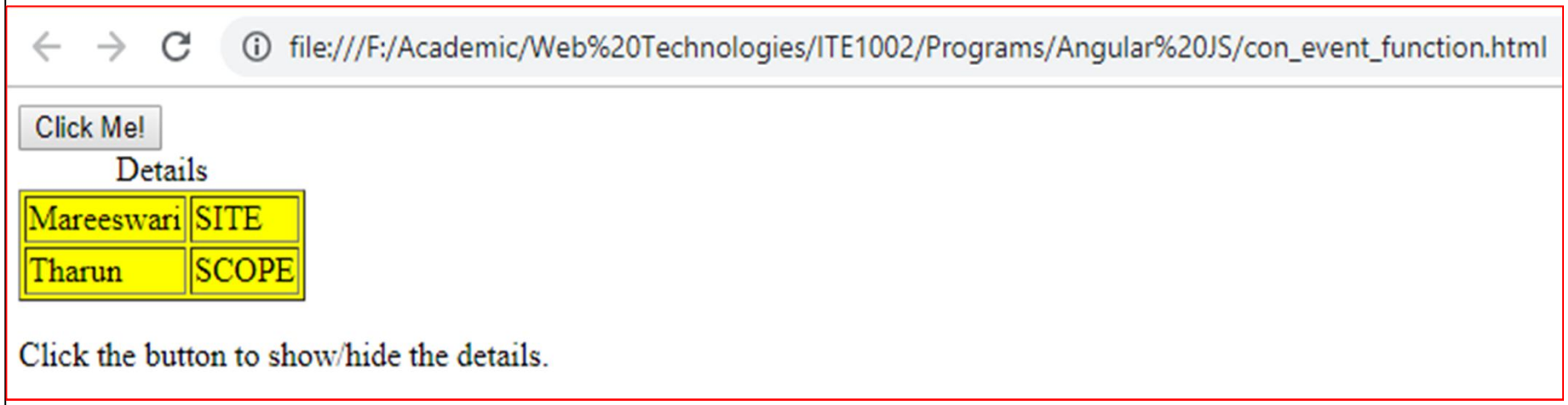

file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/Controller_event.html
Click Me!
0


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/Controller_event.html
Click Me!
1

```
<!DOCTYPE html><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">
<button ng-click="myFunc()">Click Me!</button>
<div ng-show="showMe">
<table border=1 bgcolor="yellow"><caption>Details</caption>
<tr><td>Mareeswari</td><td>SITE</td></tr>
<tr><td>Tharun</td><td>SCOPE</td></tr></table>
</div> </div> <script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
   $scope.showMe = false;
   $scope.myFunc = function() {
      $scope.showMe = !$scope.showMe;
   } });</script> <p>Click the button to show/hide the details.</p>
</body></html>
```

Call function at click for show / hide


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/con_event_function.html
Click Me!
Click the button to show/hide the details.


file:///F:/Academic/Web%20Technologies/ITE1002/Programs/Angular%20JS/con_event_function.html
Click Me!
Details
Mareeswari
SITE
Tharun
SCOPE
Click the button to show/hide the details.

Angular JS Forms

- Forms in AngularJS provides data-binding and validation of input controls.

Input Controls

- Input controls are the HTML input elements:
- input elements(formss.html)
- select elements(ngswitchselect.html)
- button elements
- Radiobutton(ngswitch1)
- textarea elements

- Input controls provides data-binding by using the ng-model directive.

Form Validation

- AngularJS offers client-side form validation.
- AngularJS monitors the state of the form and input fields (input, textarea, select), and lets you notify the user about the current state.
- AngularJS also holds information about whether they have been touched, or modified, or not.
- You can use standard HTML5 attributes to validate input, or you can make your own validation functions.

Form State and Input State

- AngularJS is constantly updating the state of both the form and the input fields.
- Input fields have the following states:
  - $untouched The field has not been touched yet
  - $touched The field has been touched
  - $pristine The field has not been modified yet
  - $dirty The field has been modified
  - $invalid The field content is not valid
  - $valid The field content is valid
- They are all properties of the input field, and are either true or false.

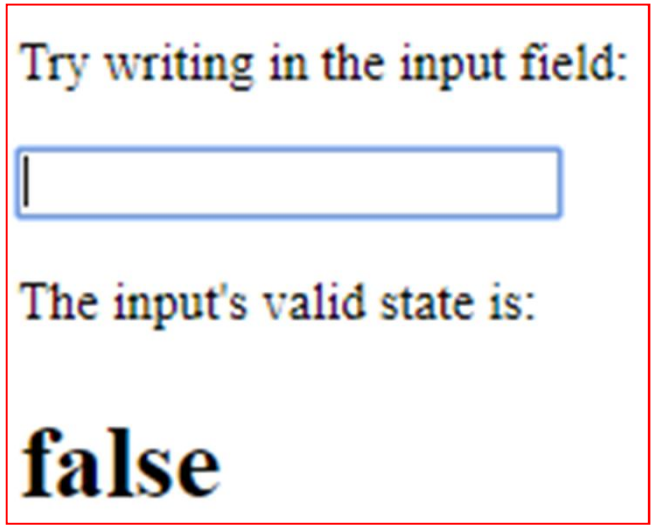
- Forms have the following states:
  - $pristine No fields have been modified yet
  - $dirty One or more have been modified
  - $invalid The form content is not valid
  - $valid The form content is valid
  - $submitted The form is submitted
- They are all properties of the form, and are either true or false.

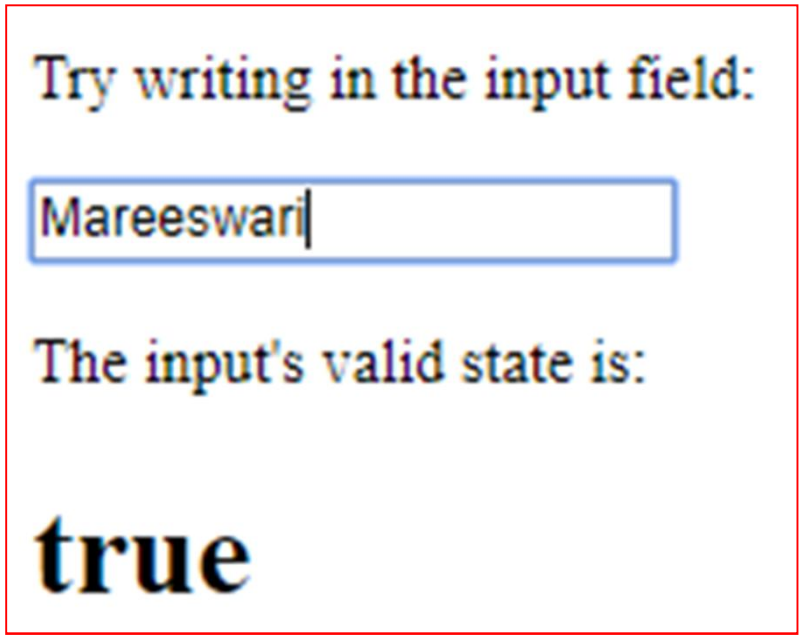
CSS Classes

- AngularJS adds CSS classes to forms and input fields depending on their states.
- The following classes are added to, or removed from, input fields:
  - ng-untouched The field has not been touched yet
  - ng-touched The field has been touched
  - ng-pristine The field has not been modified yet
  - ng-dirty The field has been modified
  - ng-valid The field content is valid
  - ng-invalid The field content is not valid
  - Cssclasses.html

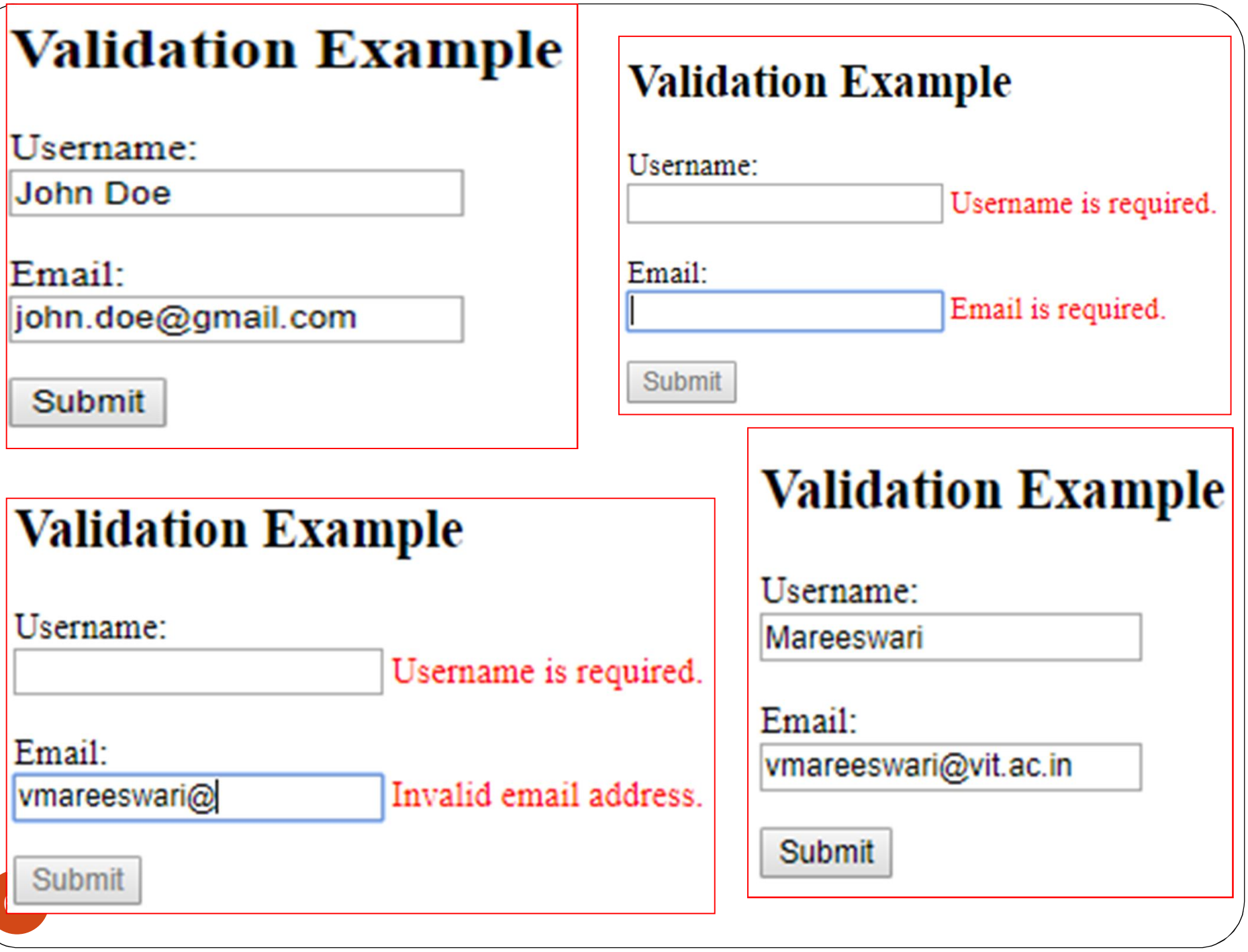
Form Validation

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min
.js"></script>
<body ng-app="">
<p>Try writing in the input field:</p>
<form name="myForm">
<input name="myInput" ng-model="myInput" required>
</form>
<p>The input's valid state is:</p>
<h1>{{myForm.myInput.$valid}}</h1>
</body></html>
```


Try writing in the input field:
The input's valid state is:
false


Try writing in the input field:
Mareeswari
The input's valid state is:
true

```
<!DOCTYPE html><html><script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body><h2>Validation Example</h2>
<form ng-app="myApp" ng-controller="validateCtrl" 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.$invalid"></p></form>
<script>var app = angular.module('myApp', []);
app.controller('validateCtrl', function($scope) {
   $scope.user = 'John Doe';
   $scope.email = 'john.doe@gmail.com';
});</script></body></html>
```

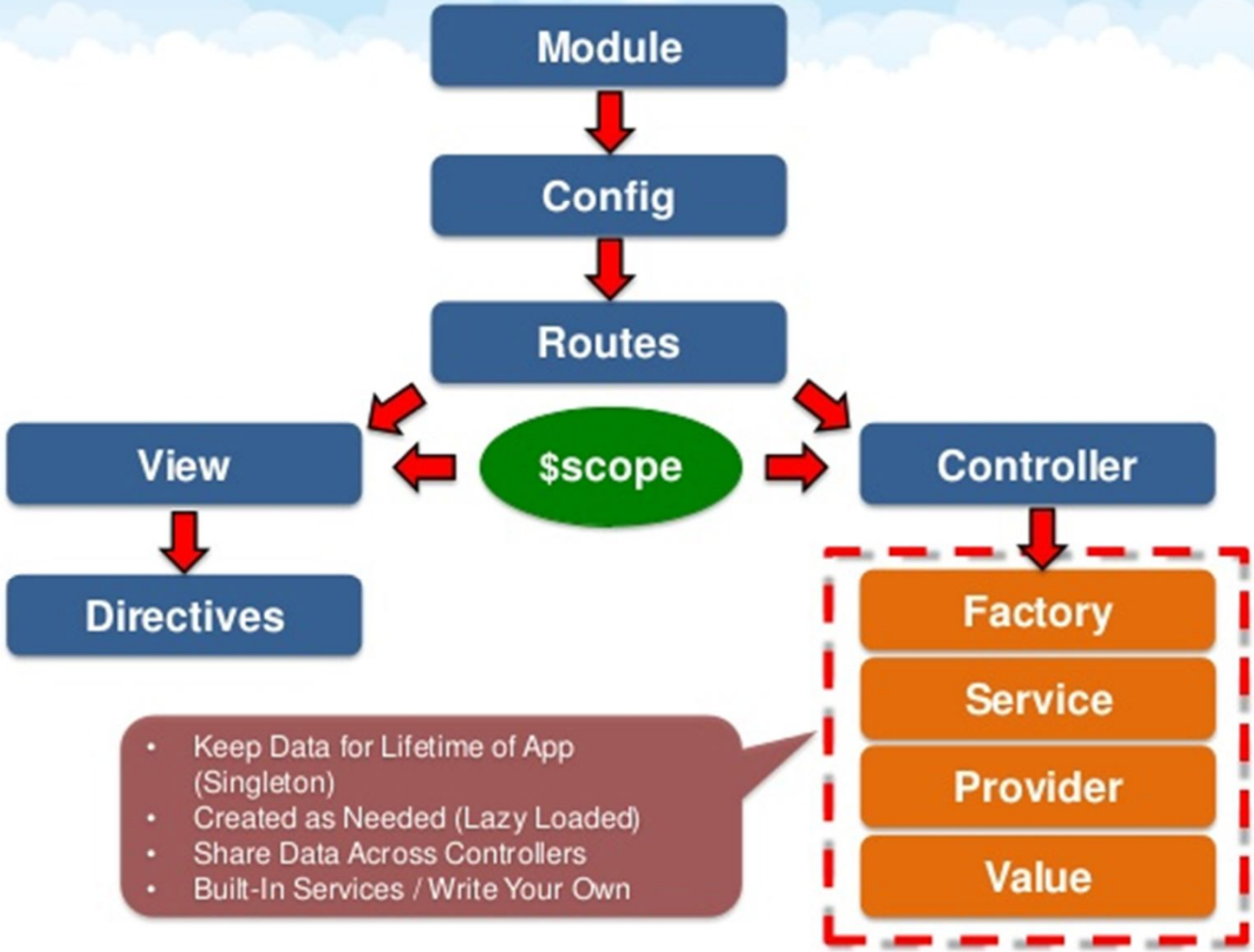

Validation Example
Username:
John Doe
Email:
john.doe@gmail.com
Submit
Validation Example
Username:
Username is required.
Email:
Email is required.
Submit
Validation Example
Username:
Username is required.
Email:
vmareeswari@
Invalid email address.
Submit
Validation Example
Username:
Mareeswari
Email:
vmareeswari@vit.ac.in
Submit

AngularJS Includes

- With AngularJS, you can include HTML from an external file using the **ng-include** directive:

-

Anatomy of an Angular App



Responsive Web Design

- Having a responsive website is very important. One must be able to access it from all devices making it available on all platforms. Thus, you must give the developers enough time to work on the website and not rush into creating a website that won't work across multiple devices.

Responsive breakpoints

- // Small devices (landscape phones, 576px and up)
- **@media (min-width: 576px) { ... }**

- // Medium devices (tablets, 768px and up)
- **@media (min-width: 768px) { ... }**

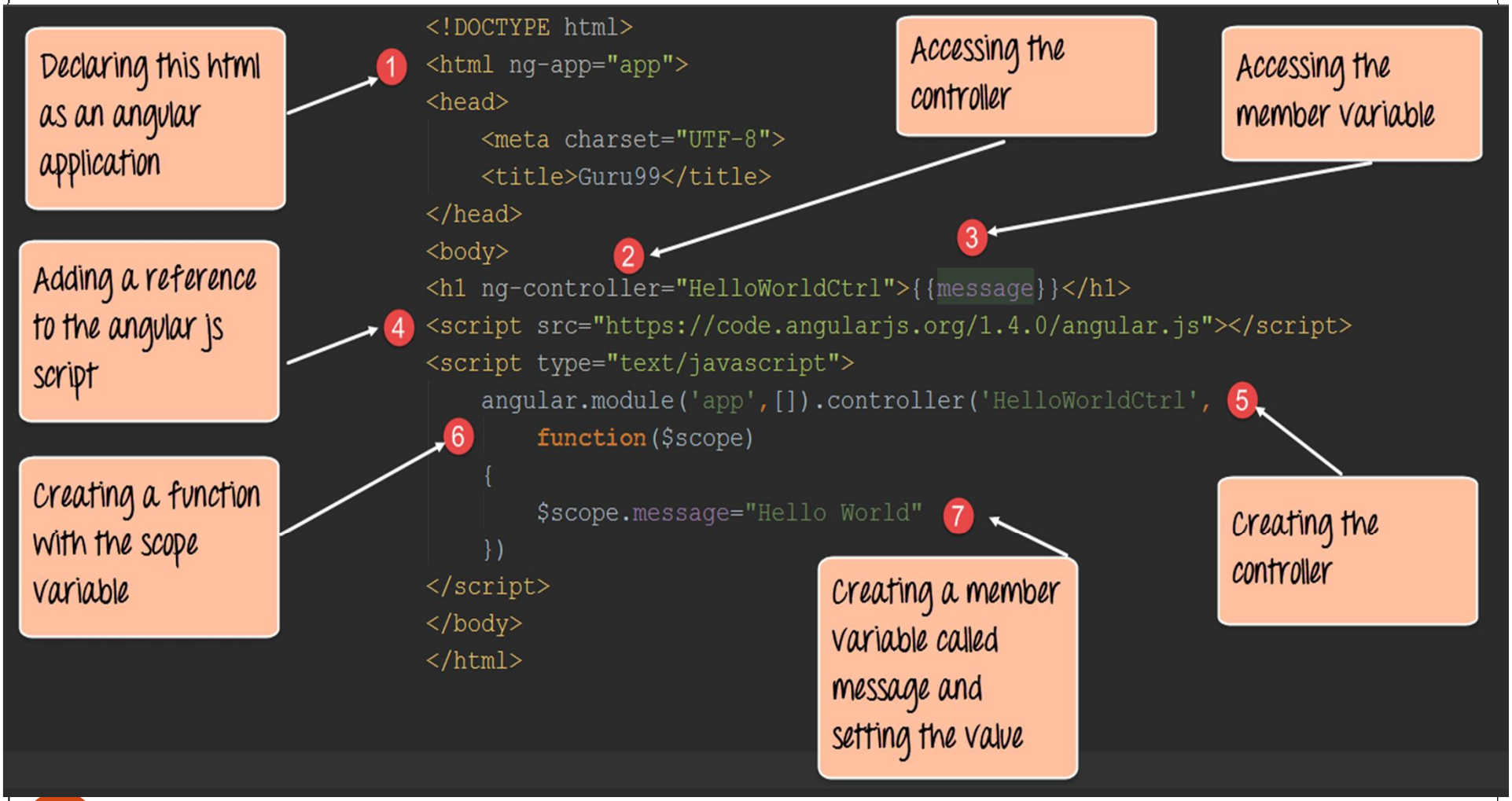
- // Large devices (desktops, 992px and up)
- **@media (min-width: 992px) { ... }**

- // Extra large devices (large desktops, 1200px and up)
- **@media (min-width: 1200px) { ... }**

Grid System

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column	Yes				

Summary


Declaring this html as an angular application
Adding a reference to the angular js script
Creating a function with the scope variable
Accessing the controller
Accessing the member variable
Creating the controller
Creating a member variable called message and setting the value
1
2
3
4
5
6
7
<!DOCTYPE html>
<html ng-app="app">
<head>
<meta charset="UTF-8">
<title>Guru99</title>
</head>
<body>
<h1 ng-controller="HelloWorldCtrl">{{message}}</h1>
<script src="https://code.angularjs.org/1.4.0/angular.js"></script>
<script type="text/javascript">
angular.module('app',[]).controller('HelloWorldCtrl',
function($scope)
{
$scope.message="Hello World"
})
</script>
</body>
</html>

References

- https://youtu.be/0kmdjqgO9IY - good for begineers