

AngularJS :-

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 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>
```

An AngularJS application consists of following three important parts –

- **ng-app** – This directive defines and links an AngularJS application to HTML.
- **ng-model** – This directive binds the values of AngularJS application data to HTML input controls.
- **ng-bind** – This directive binds the AngularJS Application data to HTML tags.

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.

Creating AngularJS Application

Step 1: Load framework

Being a pure JavaScript framework, it can be added using <Script> tag.

```
<script  
  src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js">  
</script>
```

Step 2: Define AngularJS application using ng-app directive

```
<div ng-app = "">  
  ...  
</div>
```

Step 3: Define a model name using ng-model directive

```
<p> Name: <input type = "text" ng-model = "name"></p>
```

Step 4: Bind the value of above model defined using ng-bind directive

```
<p ng-bind="name"></p>
```

Executing AngularJS Application

Use the above-mentioned three steps in an HTML page.

AngularJS Extends HTML example :-

Example of angular js (save file ok.html) :-

```
<!DOCTYPE html>
```

```
<html>
```

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

```
<body>

<div ng-app="">

<p>Input something in the input box:</p>

<p>Name: <input type="text" ng-model="name"></p>

<p ng-bind="name"></p>

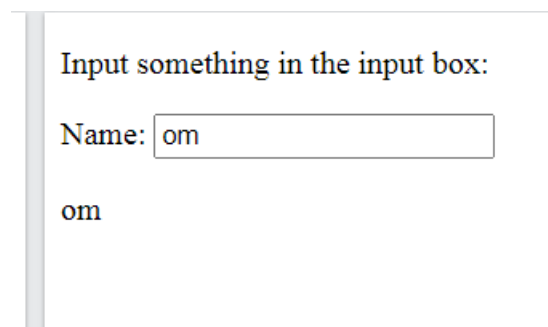
</div>

</body>

</html>
```

Open the file **ok.html** in a web browser. Enter your name and see the result.

Output:-



Example explained:-

AngularJS starts automatically when the web page has loaded.

The **ng-app** directive tells AngularJS that the <div> element is the "owner" of an AngularJS **application**.

The **ng-model** directive binds the value of the input field to the application variable **name**.

The **ng-bind** directive binds the content of the <p> element to the application variable **name**.

AngularJS Directives

As you have already seen, AngularJS directives are HTML attributes with an **ng** prefix.

The **ng-init** directive initializes AngularJS application variables.

Example :-(example.html)

```
<!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='John'">

<p>The name is <span ng-bind="firstName"></span></p>

</div>

</body>

</html>
```

Output:-

John

Alternatively with valid HTML:

AngularJS Example:-

```
<!DOCTYPE html>

<html>

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

<body>

<div data-ng-app="" data-ng-init="firstName='John'">

<p>The name is <span data-ng-bind="firstName"></span></p>

</div>

</body>

</html>
```

Output:-

The name is John

AngularJS Expressions:-

AngularJS expressions are written inside double braces: **{{ expression }}**.

AngularJS will "output" data exactly where the expression is written:

AngularJS Example:-

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

<div ng-app="">
  <p>My first expression: {{ 5 + 5 }}</p>
</div>

</body>
</html>
```

Output:-

My first expression: 10

Note:- AngularJS expressions bind AngularJS data to HTML the same way as the **ng-bind** directive.

AngularJS Example (example.html)

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

<div ng-app="">

<p>Input something in the input box:</p>
```

```
<p>Name: <input type="text" ng-model="name"></p>
```

```
<p>{{name}}</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

Output:-

Input something in the input box:

Name:

om

AngularJS Applications

AngularJS **modules** define AngularJS applications.

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`

```
<div ng-app="myApp">...</div>

<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** control AngularJS applications.

Adding a Controller:-

Add a controller to your application, and refer to the controller with the

`ng-controller` directive:

Example

```
<div ng-app="myApp" ng-controller="myCtrl">
{{ firstName + " " + lastName }}
</div>

<script>

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

app.controller("myCtrl", function($scope) {
```



```
    $scope.firstName = "John";  
    $scope.lastName = "Doe";  
});  
  
</script>
```

The **ng-app** directive defines the application, the **ng-controller** directive defines the controller.

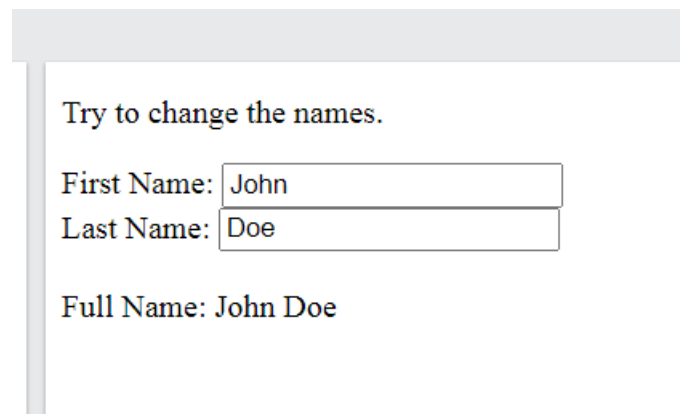
AngularJS Example

```
<!DOCTYPE html>  
  
<html>  
  
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>  
  
<body>  
  
<p>Try to change the names.</p>  
  
<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= "John";
    $scope.lastName= "Doe";
});
</script>

</body>
</html>
```

Output:-



Try to change the names.

First Name:

Last Name:

Full Name: John Doe

Note:- AngularJS modules define applications:

AngularJS Module

```
var app = angular.module('myApp', []);
```

AngularJS controllers control applications:

AngularJS Controller

```
app.controller('myCtrl', function($scope) {  
  $scope.firstName= "John";  
  $scope.lastName= "Doe";  
});
```

AngularJS Routing:-

What is Routing in AngularJS?

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.

Example:

Navigate to "main.html ,red.html", "green.html", and "blue.html":-

What do you Need?

To make your applications ready for routing, you must include the AngularJS Route module:

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

Then you must add the `ngRoute` as a dependency in the application module:

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

Now your application has access to the route module, which provides the `$routeProvider`.

Use the `$routeProvider` to configure different routes in your application:

```
app.config(function($routeProvider) {
  $routeProvider
    .when("/", {
      templateUrl : "main.html"
    })
    .when("/red", {
      templateUrl : "red.html"
    })
    .when("/green", {
      templateUrl : "green.html"
    })
    .when("/blue", {
      templateUrl : "blue.html"
    });
});
```

Where Does it Go?

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:

Example:

```
<div ng-view></div>
```

Example:

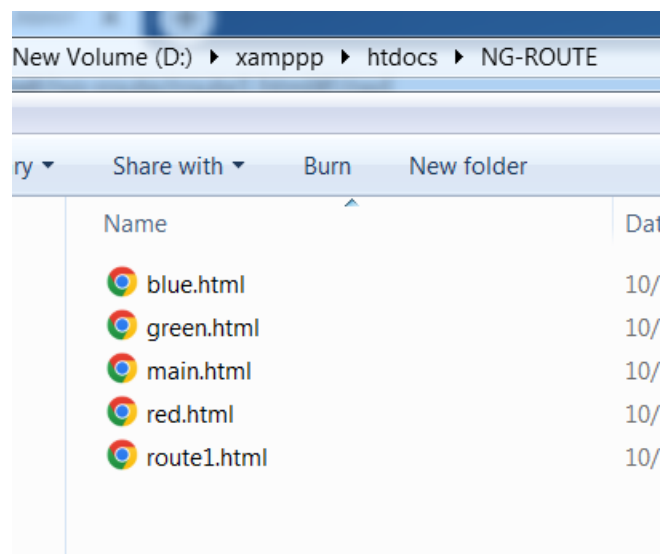
```
<ng-view></ng-view>
```

Example:

```
<div class="ng-view"></div>
```

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

First of all go to htdocs folder inside your xampp folder create a folder “**NG-ROUTE**” **AS SHOWN BELOW:-**



Here create html files

1) blue.html :-

```
<html>
<head> blue page</head>
<body>
<h2 style="background-color:blue;"> it is red color page</h2>
    </body>
</html>
```

2)red.html file :-

```
<html>
<head> it is red page</head>
<body>
<h2 style="background-color:red;"> it is red color page</h2>
    </body>
</html>
```

3)green.html file :-

```
<html>
<head> Javascript Events</head>
<body>
<h2 style="background-color:green;"> it is red color page</h2>
</body>
</html>
```

4)main.html file :-

```
<html>
```

```
<head> Javascript Events</head>
```

```
<body>
```

```
<h2> it is main page</h2>
```

```
</body>
```

```
</html>
```

5)write code for route1.html file:-

```
<!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="#!red">Red</a>
```

```
<a href="#!green">Green</a>
```

```
<a href="#!blue">Blue</a>
```

```
<ng-view></ng-view>
```

```
<script>
```

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

```
app.config(function($routeProvider) {
```

```
    $routeProvider
```

```
    .when("/", {
```

```
        templateUrl : "main.html"
```

```
    })
```

```
    .when("/red", {
```

```
        templateUrl : "red.html"
```

```
    })
```

```
    .when("/green", {
```

```
        templateUrl : "green.html"
```

```
    })
```

```
    .when("/blue", {
```

```
        templateUrl : "blue.html"
```

```
    });
```

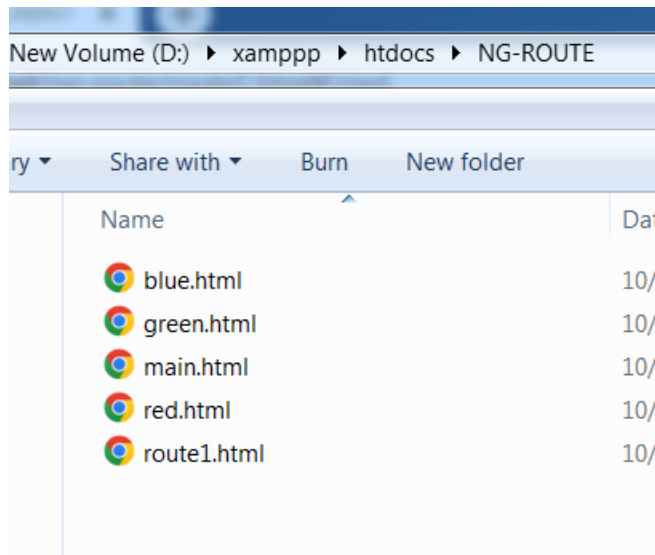
```
});
```

```
</script>
```

```
<p>Click on the links to navigate</p>
```

```
</body>
```

```
</html>
```

Now run your xampp as shown below :-

