Unit I AngularJS Core Concepts:

AngularJS is an open-source web application framework. It was originally developed in 2009 by Misko Hevery and Adam Abrons. It is now maintained by Google. Its latest version is 1.2.21.

Definition of AngularJS

AngularJS is a structural framework for dynamic web applications. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application components clearly . Its data binding and dependency injection eliminate much of the code you currently have to write. And it all happens within the browser, making it an ideal partner with any server technology.

General Features

The general features of AngularJS are as follows −

- AngularJS is a efficient framework that can create Rich Internet Applications (RIA).
- AngularJS provides developers an options to write client side applications using JavaScript in a clean Model View Controller (MVC) way.
- Applications written in AngularJS are cross-browser compliant. AngularJS automatically handles JavaScript code suitable for each browser.
- AngularJS is open source, completely free, and used by thousands of developers around the world. It is licensed under the Apache license version 2.0.

Overall, AngularJS is a framework to build large scale, high-performance, and easyto-maintain web applications.

Core Features

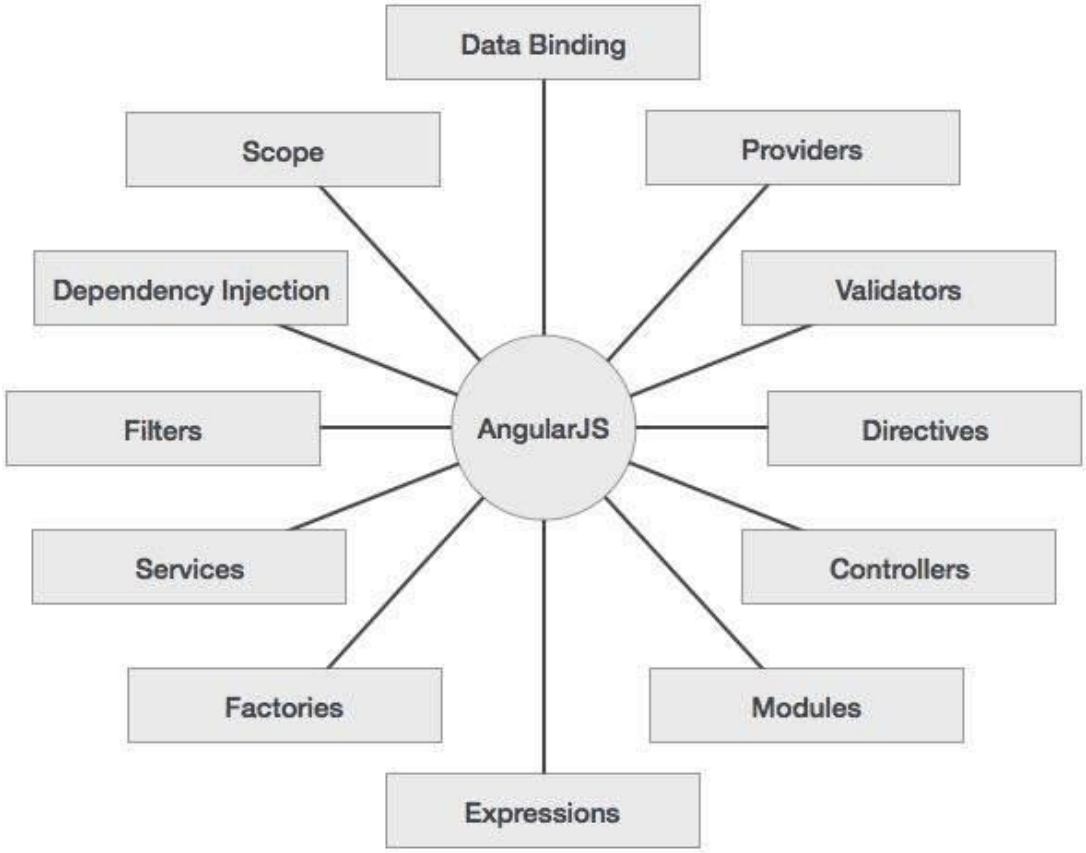
The core features of AngularJS are as follows −

1. **Data-binding** − It is the automatic synchronization of data between model and view components.
2. **Scope** − These are objects that refer to the model. They act as a glue between controller and view.
3. **Controller** − These are JavaScript functions bound to a particular scope.

4. **Services** − AngularJS comes with several built-in services such as $http to make a XMLHttpRequests. These are singleton objects which are instantiated only once in app.
5. **Filters** − These select a subset of items from an array and returns a new array.
6. **Directives** − Directives are markers on DOM elements such as elements, attributes, css, and more. These can be used to create custom HTML tags that serve as new, custom widgets. AngularJS has built-in directives such as ngBind, ngModel, etc.
7. **Templates** − These are the rendered view with information from the controller and model. These can be a single file (such as index.html) or multiple views in one page using *partials*.
8. **Routing** − It is concept of switching views.
9. **Model View Whatever** − MVW is a design pattern for dividing an application into different parts called Model, View, and Controller, each with distinct responsibilities. AngularJS does not implement MVC in the traditional sense, but rather something closer to MVVM (Model-View-ViewModel). The Angular JS team refers it humorously as Model View Whatever.
10. **Deep Linking** − Deep linking allows to encode the state of application in the URL so that it can be bookmarked. The application can then be restored from the URL to the same state.
11. **Dependency Injection** − AngularJS has a built-in dependency injection subsystem that helps the developer to create, understand, and test the applications easily.

Concepts

The following diagram depicts some important parts of AngularJS which we will discuss in detail in the subsequent chapters.



Concept	Description
Template	HTML with additional markup
Directives	Extends the HTML with custom attributes and elements
Model	The data shown to the user in the view and with which the user interacts
Scope	A context where the model is stored so that controllers, directives and expressions can access it
Expressions	Executes JavaScript code inside brackets {{ }}.
Compiler	Parses the template and instantiates directives and expressions
Filter	Formats the value of an expression for display to the user
View	what the user sees (the DOM)
Data Binding	Sync data between the model and the view
Controller	Maintains the application data and business logic
Module	a container for different parts of an app including controllers, services, filters, directives which configure the Injector
Service	Reusable business logic, independent of views
Dependency Injection	Creates and wires objects and functions

Concept	Description
Injector	Dependency injection container

Advantages of AngularJS

The advantages of AngularJS are −

- It provides the capability to create Single Page Application in a very clean and maintainable way.
- It provides data binding capability to HTML. Thus, it gives user a rich and responsive experience.
- AngularJS code is unit testable.
- AngularJS uses dependency injection and make use of separation of concerns.
- AngularJS provides reusable components.
- With AngularJS, the developers can achieve more functionality with short code.
- In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.

On the top of everything, AngularJS applications can run on all major browsers and smart phones, including Android and iOS based phones/tablets.

Key differences between JavaScript and AngularJS

Both are popular choices in the market; let us discuss some of the major difference:

- JavaScript is a client-side scripting language for creating web applications.

  AngularJS is a framework written in JS to build single page applications.

- JavaScript is a language used to manipulate the document object

  model. AngularJS extend its attributes with other technologies like HTML.

- JavaScript has a feature to validate the user input at browser level before

  submitting the page. AngularJS is a good technology used as server side.

- Javascript makes websites more interactive. AngularJS makes web applications simple and easy to create from start.
- JavaScript is versatile and complex language. AngularJS is simple and efficient.
- JavaScript code is generally fast. Angular JS application becomes generally slow.
- JavaScript does not support Dependency Injection. AngularJS supports data binding and dependency injection.
- JavaScript has a rich user interface design like sliders etc. AngularJS is a full data-driven approach to develop web applications.
- JavaScript is less time consuming while fixing bugs and defects at a large scale. AngularJS is more time consuming doing the same as compared to JavaScript.
- JavaScript gives us immediate feedback while running the application on the browser. AngularJS does not have this feature.
- javascript is the most powerful web development techniques used for developing web applications. AngularJS has been preferred as mostly as JS framework for creating web applications.

- JavaScript uses the same technique from the years. AngularJS has been enhanced to typescript, which makes the applications more light and interactive.

JavaScript vs AngularJS Comparison Table

The primary Comparison between are discussed below:

The basis of comparison	**JavaScript**	**AngularJS**
Definition	JavaScript is an object-oriented scripting language.	AngularJS is an open source framework for creating dynamic web applications.
Programmed	JavaScript most interpreters written in C and C++.	AngularJS is written in Java Script.
Applications	JavaScript used for creating dynamic web applications.	AngularJS used for creating large single page applications.

Developed	It was mainly developed by Netscape Communications.	It was mainly developed by Google.
Concept	JavaScript based on a concept of dynamic typing as an interpreted language	It is based on a concept of model view controller to build apps.
Syntax	JavaScript syntax is complex as compared to AngularJS.	AngularJS syntax is simple.
Learnability	JavaScript is complex to learn.	AngularJS can learn easily if one knows JavaScript.
Filters	JavaScript does not support filters.	AngularJS does support filters.

Both are the two different web technologies used for developing web applications. JavaScript is a programming language whereas AngularJS is an open source framework and based on MVC architecture.

JavaScript is mainly written for websites to run in the client's browser. JavaScript also provides the interactivity, when a user hovers the mouse on any component.

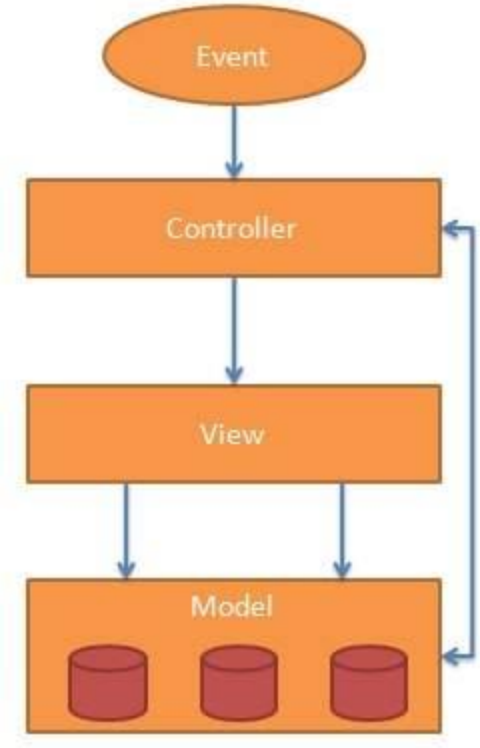
JavaScript can be used with any of the latest user interface technology for creating web apps. It is mainly used with Java applications and JQuery.

AngularJS is mainly used for large-scale single page web applications. AngularJS is most popular among the developers for creating web apps and mainly to make a user interface more interactive. AngularJS is easy to learn if a developer knows JavaScript.

Model **V**iew **C**ontroller 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.

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. The MVC abstraction can be graphically represented as follows.



The Model

The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself.

The View

A 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.

The Controller

The 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 is a MVC based framework.

Introduction to SPA

In traditional web applications, the client (browser) initiates the communication with the server by requesting a page. The server then processes the request and sends the HTML of the page to the client. In subsequent interactions with the page, for example the user navigates to a link or submits a form with data, a new request is sent to the server and the flow starts again. The server processes the request and sends a new page to the browser in response to the new action requested by the client.

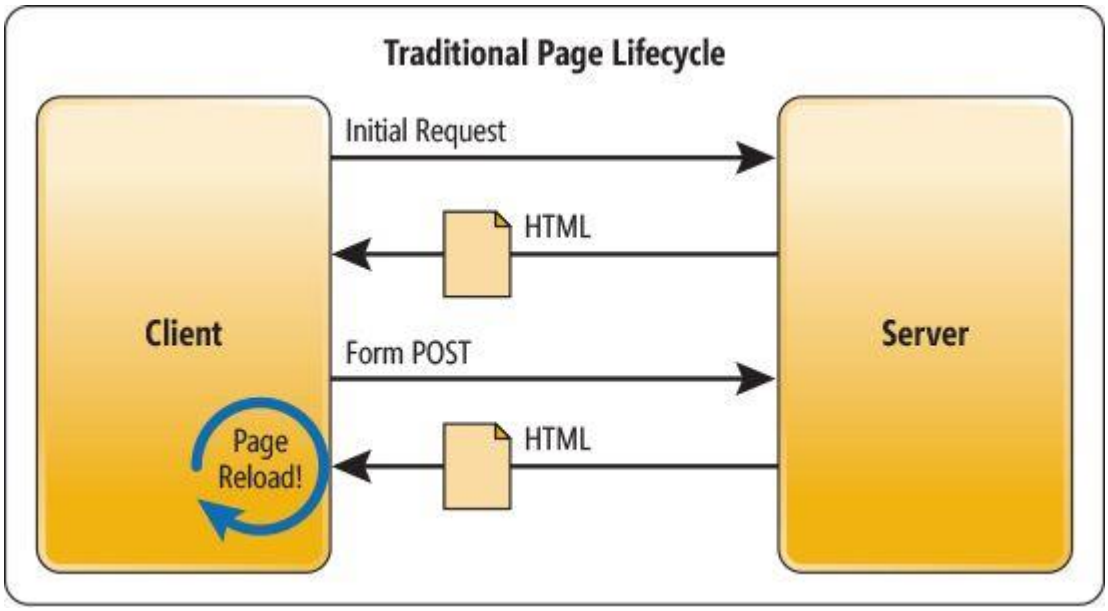
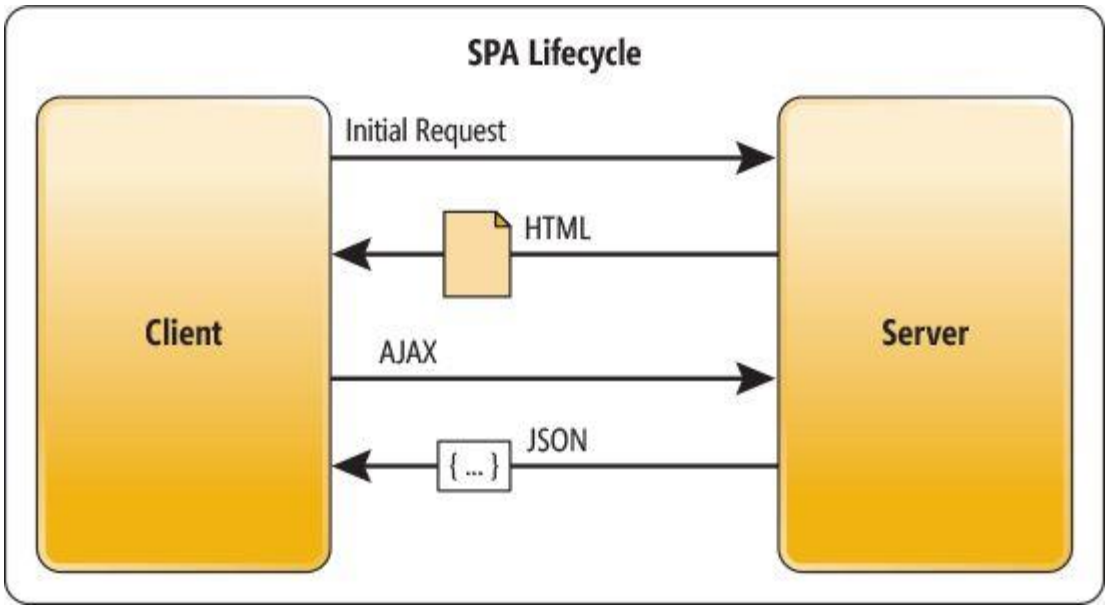


Image 1.

In Single-Page Applications (SPAs) the entire page is loaded into the browser after the initial request, but subsequent interactions take place using Ajax requests. This means that the browser must update only the portion of the page that has changed; there is no need to reload the entire page. The SPA approach reduces the time taken by the application to respond to user actions, resulting in a more fluid experience.

Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app.



SPA means **Single Page Applications**. Such web applications load the HTML, CSS or JS of the page once, and don't require reloading within page-to-page navigation.

AngularJS is one of the first JS frameworks for single page applications. It is a set of tools that significantly simplifies SPA development. AngularJS smoothly integrates with other libraries, thus, it's easy to build a unique application that perfectly fits all your needs.

Another advantage of this framework is that it duplicates the old JavaScript modules, so a developer can start coding without additional learning.

Angular JS Environment Setup

This chapter describes how to set up AngularJS library to be used in web application development. It also briefly describes the directory structure and its contents.

When you open the link https://angularjs.org/, you will see there are two options to download AngularJS library −



- **View on GitHub** − By clicking on this button, you are diverted to GitHub and get all the latest scripts.
- **Download AngularJS 1** − By clicking on this button, a screen you get to see a dialog box shown as −



This screen gives various options of using Angular JS as follows −

- **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 1.3.x.
    - We can also go with the minimized, uncompressed, or zipped version.
- **CDN access** − You also have access to a CDN. The CDN gives you access to regional data centers. In this case, the Google host. The CDN transfers the responsibility of hosting files from your own servers to a series of external ones. It also offers an advantage that if the visitor of your web page has already downloaded a copy of AngularJS from the same CDN, there is no need to re-download it.

First AngularJS Application

Let's create a simple AngularJS web application step by step and understand the basic building blocks of AngularJS.

1. First, create an HTML document with <head> and <body> elements, as show below.

Example: HTML Template

```
<!DOCTYPEhtml>

<html>
<head>

</head>
<body>

</body>
</html>
```

2. Include angular.js file in the head section (you have learned how to download angular library in the previous section). You can take a reference from the CDN also. (all the examples in this tutorials will use CDN reference.)

Example: Include AngularJS Library

```
<!DOCTYPEhtml>
<html>
<head>
<title>First AngularJS Application</title>
<scriptsrc= "~/Scripts/angular.js"></script>
</head>
<body>

</body>
</html>
```

3. Here, we will be creating a simple multiplier application which will multiply two numbers and display the result. User will enter two numbers in two separate textboxes and the result will be displayed immediately, as shown below.



First AngularJS Application

The following is the HTML code with AngularJS for the above multiplier example.

Example: First AngularJS Application

```
<!DOCTYPEhtml>

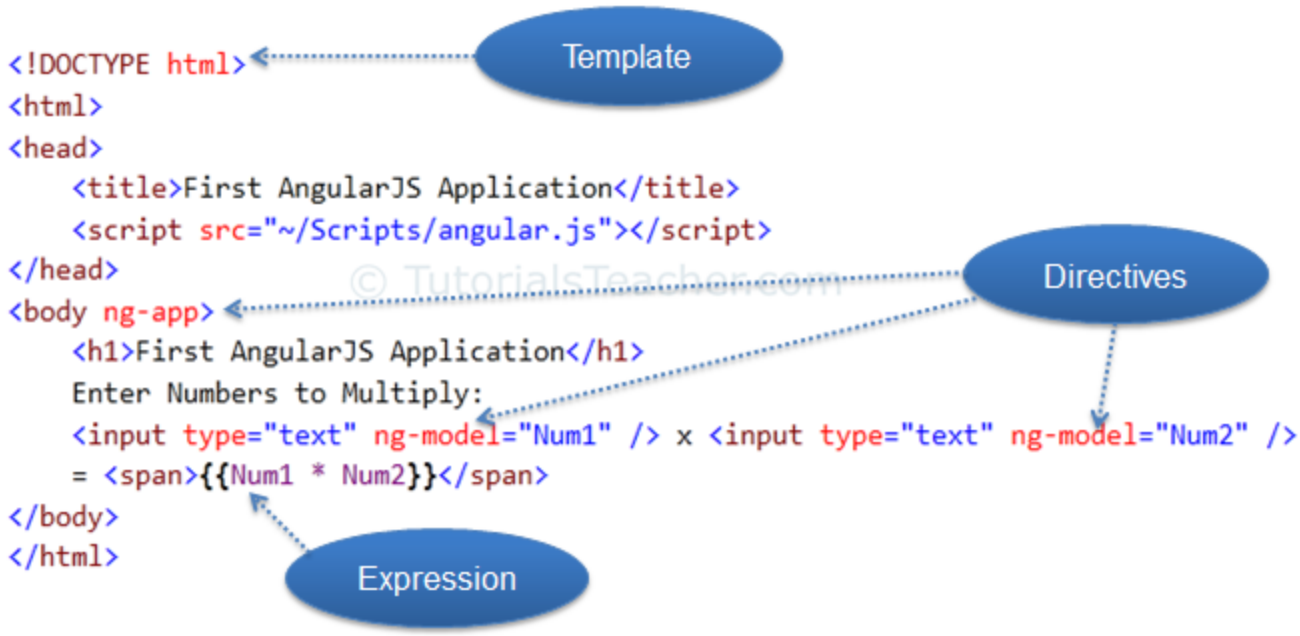
<html>
<head>
<title>First AngularJS Application</title>
<scriptsrc= "~/Scripts/angular.js"></script>
</head>
<bodyng-app>
<h1>First AngularJS Application</h1>

   Enter Numbers to Multiply:
<inputtype="text"ng-model="Num1"/>          x          <inputtype="text"ng-model="Num2"/>
   = <span>{{Num1 * Num2}}</span>
</body>
</html>
```

Try it

The above example is looks like HTML code with some strange attributes and braces such as ng-app, ng-model, and {{}}. These built-in attributes in AngularJS are called directives.

The following figure illustrates the AngularJS building blocks in the above example.



First AngularJS Application

Template

In AngularJS, a template is HTML with additional markups. AngularJS compiles templates and renders the resultant HTML.

Directive

Directives are markers (attributes) on a DOM element that tell AngularJS to attach a specific behavior to that DOM element or even transform the DOM element and its children.

Most of the directives in AngularJS are starting with **ng**. It stands for Angular. We have applied ng-app and ng-model directive in the above example.

ng-app: The ng-app directive is a starting point. If AngularJS framework finds ng-app directive anywhere in the HTML document then it bootstraps (initializes) itself and compiles the HTML template.

ng-model: The ng-model directive binds HTML element to a property on the $scope object. You will learn about this model later but for now let us consider this as a model property.

In the above example, we have included ng-model directive to both the textboxes with different names Num1 and Num2. AngularJS framework will create two

properties called Num1 and Num2 in the scope and will assign a value that we type into textboxes.

Expression

An expression is like JavaScript code which is usually wrapped inside double curly braces such as {{ expression }}. AngularJS framework evaluates the expression and produces a result. In the above example, {{ Num1 * Num2}} will simply display the product of Num1 and Num2.

First angular js Example

AngularJS First Example

AngularJS applications are a mix of HTML and JavaScript. The first thing you need is an HTML page.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. .
5. .
6. </head>
7. <body>
8. .
9. .
10. </body>
11. </html>
```

Second, you need to include the AngularJS JavaScript file in the HTML page so we can use AngularJS:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular.min.js"
   ></script>
5. </head>
6. <body>
7. .
8. .
```

```
9.  </body>
10. </html>
```

Note: You should always use the latest version of AngularJS, so it is not necessary to use the same version as the above example.

AngularJS First Example

Following is a simple "Hello Word" example made with AngularJS. It specifies the Model, View, Controller part of an AngularJS app.

```
<!DOCTYPE html>
<html >
<head>
   <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular.min.js">
</script>
</head>
<body ng-app="myapp">
<div ng-controller="HelloController" >
<h2>Hello {{helloTo.title}} !</h2>
</div>

<script>
angular.module("myapp", [])
   .controller("HelloController", function($scope) {
      $scope.helloTo = {};
      $scope.helloTo.title = "World, AngularJS";
   } );
</script>
</body>
</html>
```

AngularJS extends HTML with **ng-directives**.

The **ng-app** directive defines an AngularJS application.A module is created by using the AngularJS function angular.module

The "myapp" parameter refers to an HTML element in which the application will run.

Add a controller to your application, and refer to the controller with the ng-controller directive as “HelloController”

View Part

```
<div ng-controller="HelloController" >
<h2>Hello {{helloTo.title}} !</h2>
</div>
```

Controller Part

```
<script>
angular.module("myapp", [])
    .controller("HelloController", function($scope) {
        $scope.helloTo = {};
        $scope.helloTo.title = "World, AngularJS";
    });
</script>
```