

Try it Option Online

You really do not need to set up your own environment to start learning AngularJS. Reason is very simple, we already have set up AngularJS environment online, so that you can execute all the available examples online at the same time when you are doing your theory work. This gives you confidence in what you are reading and to check the result with different options. Feel free to modify any example and execute it online.

*Try the following example using **Try it** option available at the top right corner of the below sample code box −*

```
<!doctype html>
<html ng-app>

   <head>
      <script src =
"https://ajax.googleapis.com/ajax/libs/angularjs/1.3.3/angular.min.js"></s
cript>
   </head>

   <body>
      <div>
         <label>Name:</label>
         <input type = "text" ng-model = "yourName" placeholder = "Enter a
name here">
         <hr />

         <h1>Hello {{yourName}}!</h1>
      </div>

   </body>
</html>
```

*For most of the examples given in this tutorial, you will find **Try it** option, so just make use of it and enjoy your learning.*

In this chapter we will discuss about how to set up AngularJS library to be used in web application development. We will also briefly study 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** − Click on this button to go to GitHub and get all of the latest scripts.
- **Download** − Or click on this button, a screen as below would be seen −

- 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 itself are self descriptive. **legacy** has version less than 1.2.x and **latest** has 1.4.x version.
        - We can also go with the minified, uncompressed or zipped version.
    - **CDN access** − You also have access to a CDN. The CDN will give you access around the world to regional data centers that in this case, Google host. This means using CDN moves the responsibility of hosting files from your own servers to a series of external ones. This also offers an advantage that if the visitor to your webpage has already downloaded a copy of AngularJS from the same CDN, it won't have to be re-downloaded.

> *We are using the CDN versions of the library throughout this tutorial.*

Example

Now let us write a simple example using AngularJS library. Let us create an HTML file *myfirstexample.html* as below −

```
<!doctype html>
<html>

   <head>
      <script src = "https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-
beta.17/angular.min.js"></script>
   </head>

   <body ng-app = "myapp">

      <div ng-controller = "HelloController" >
         <h2>Welcome {{helloTo.title}} to the world of Tutorialspoint!</h2>
      </div>

      <script>
         angular.module("myapp", [])

         .controller("HelloController", function($scope) {
```

```
            $scope.helloTo = {};
            $scope.helloTo.title = "AngularJS";
         });
      </script>

   </body>
</html>
```

Following sections describe the above code in detail −

Include AngularJS

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

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

Check the latest version of AngularJS on their official website.

Point to AngularJS app

Next we tell what part of the HTML contains the AngularJS app. This done by adding the *ng-app* attribute to the root HTML element of the AngularJS app. You can either add it to *html* element or *body* element as shown below −

```
<body ng-app = "myapp">
</body>
```

View

The view is this part −

```
<div ng-controller = "HelloController" >
   <h2>Welcome {{helloTo.title}} to the world of Tutorialspoint!</h2>
</div>
```

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

The controller part is −

```
<script>
   angular.module("myapp", [])

   .controller("HelloController", function($scope) {
      $scope.helloTo = {};
      $scope.helloTo.title = "AngularJS";
   });
</script>
```

This code registers a controller function named *HelloController* in the angular module named *myapp*. We will study more about modules and controllers in their respective chapters. The controller function is registered in angular via the angular.module. . . .controller. . . function call.

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.

Execution

Save the above code as *myfirstexample.html* and open it in any browser. You will see an output as below −

```
Welcome AngularJS to the world of Tutorialspoint!
```

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.