Q1) write a note on following directive with proper syntax:
Ans:
AngularJS uses directives to augment HTML with extra functionality. Essentially, directives are a convenient way to declaratively call JavaScript functions. Ng-show, ng-controller, ng-init, etc.

A)ng-show
Ans:
- ng-show is a part of AngularJS directives. ng-show Directive is used to show or hide the specified HTML element.
- If the given expression in ng-show attribute is true then the HTML element will display otherwise it will hide the HTML element.

Syntax:

```
<element ng-show="expression"> </element>
```

Snippet:

```
<body ng-app = "">
<p ng-show = “true”>This paragraph can be seen </p>
</body>
```

B)ng-controller
Ans:
- The ng-controller Directive in AngularJS is used to add controllers to the application.
- It can be used to add methods, functions and variables that can be called on some event like click, etc to perform certain actions.
- Controllers in AngularJs take the data from the View, process the data, and then it sends that data across to the view which is displayed to the end user.
- 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.

Syntax:

```
<element ng-controller="expression"> Contents... </element>
```

Snippet:

```
<body ng-app="myapp" ng-controller="myCtrl”>
    {{message}}
</body>
<script>
    myapp = angular.module('myapp', [])
    myapp.controller('myCtrl', function ($scope) {
        $scope.message = "Hello World";
    })
</script>
```

C)ng-init
Ans:

- The ng-init directive is used to initialize AngularJS Application data.
- It defines the initial value for an AngularJS application and assigns values to the variables.
- The ng-init directive defines initial values and variables for an AngularJS application.

Syntax:

```
<element ng-init = "expression">
    ...
</element>
```

Snippet:

```
<body ng-app="" ng-init="quantity=1;price=5">
    People: <input type="number" ng-model="quantity">
    Registration Price: <input type="number" ng-model="price">
    Total: {{quantity * price}}
</body>
```

Q2) Explain a predefined filter used in angularJS.
Ans:

A filter formats the value of an expression to display to the user. AngularJS Filters allow us to format the data to display on UI without changing the original format. Filters can be used with an expression or directives using pipe | sign. There are built-in filters such as 'lowercase', 'uppercase' which can retrieve the output in lowercase and uppercase accordingly. Similarly, for numbers, you can use other filters.

AngularJS provides filters to transform data:

- **currency :** Format a number to a currency format.
- **date :** Format a date to a specified format.
- **filter :** Select a subset of items from an array.
- **json:** Format an object to a JSON string.
- **limitTo :** Limits an array/string, into a specified number of elements/characters.
- **lowercase :** Format a string to lowercase.
- **number :** Format a number to a string.
- **orderBy :** Orders an array by an expression.
- **uppercase :** Format a string to uppercase.

Q3) What is service? What are the different methods used for creating services in angularJS? Explain with examples.
Ans:

AngularJS services are JavaScript functions, which are responsible for performing only specific tasks. This makes them individual entities which are maintainable and testable. The controllers and filters can call them on a requirement basis. Services are normally injected using the dependency injection mechanism of AngularJS.

AngularJS provides many inbuilt services. For example, $http, $route, $window, $location, etc. The inbuilt services are always prefixed with $ symbol.
There are two ways to create a service :

- **Factory :**

In this method, we first define a factory and then assign method to it.
```
var mainApp = angular.module("mainApp", []);
mainApp.factory('MathService', function() {
var factory = {};
factory.multiply = function(a, b)
{
return a * b
}
return factory;
});
```

- **Service :**

In this method, we define a service and then assign a method to it. We also inject an already available service to it.
```
mainApp.service('CalcService', function(MathService) {
this.square = function(a) {
return MathService.multiply(a,a);
}
});
```

Q4) What is routing? Explain with a proper example.
Ans:

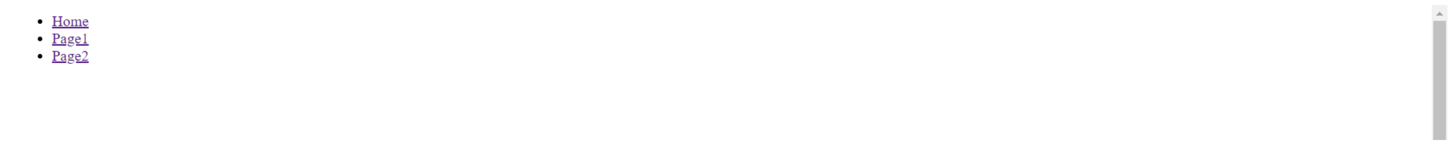
We can build Single Page Application (SPA) with AngularJS. It is a web app that loads a single HTML page and dynamically updates that page as the user interacts with the web app. AngularJS supports SPA using the routing module ngRoute. This routing module acts based on the URL.

When a user requests a specific URL, the routing engine captures that URL and renders the view based on the defined routing rules. AngularJS routes enable the user to create different URLs for different content in an application. The ngRoute module helps in accessing different pages of an application without reloading the entire application.

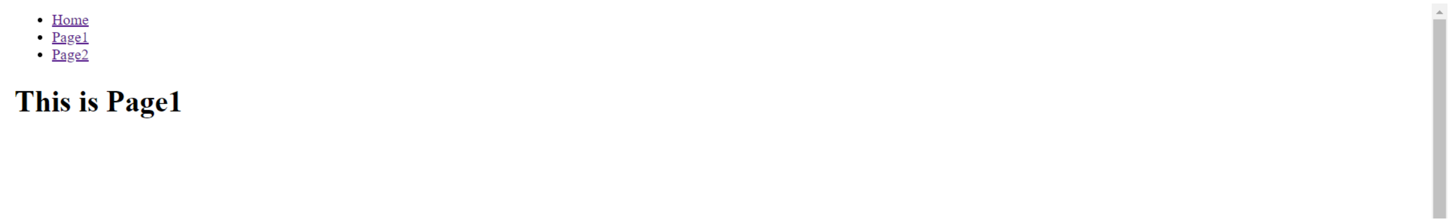
example

```
<html>
<head>
    <title>Route Example</title>
    <script src="../angular.min.js"></script>
    <script src="../angular-route.min.js"></script>
</head>

<body>
    <div ng-app="myapp" ng-controller='myCtrl'>
        <ul>
            <li><a href="#!/">Home</a></li>
            <li><a href="#!/page1">Page1</a></li>
            <li><a href="#!/page2">Page2</a></li>

        </ul>

        <div ng-view></div>
    </div>
    <script>
        var myapp = angular.module('myapp', ['ngRoute'])
        myapp.controller('myCtrl', function ($scope) {
            $scope.title = "Route Example" //uncomment only if error
        })
        myapp.config(['$routeProvider', function ($routeProvider) {
            $routeProvider
                .when('/page1', {
                    template: '<h1>This is Page1</h1>'
                })
                .when('/page2', {
                    template: '<h1>This is Page2</h1>'
                })
                .otherwise({
                    redirectTo: '/'
                })
            }])
        </script>
    </body>
</html>
```

output

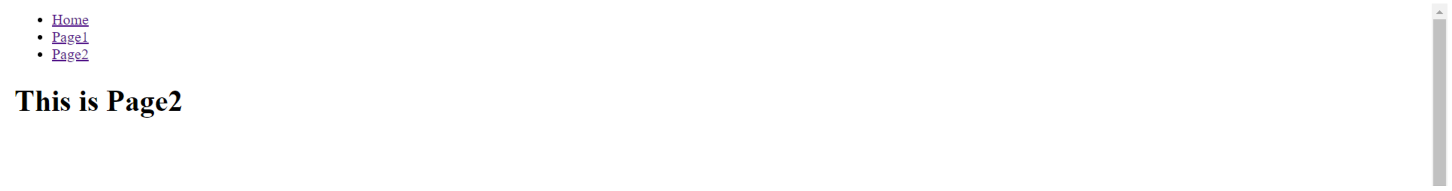


- Home
- Page1
- Page2



- Home
- Page1
- Page2

This is Page1



- Home
- Page1
- Page2

This is Page2

Q5) Create an AngularJS ToDo app which will add or delete a list of items. (Execution is imp ..not front-end)
Ans:

html file

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="a1_todo_main.css">
    <script src="../angular.min.js"></script>
    <script src="../angular-route.min.js"></script>
    <script src="a1_todo_app.js" defer></script>
    <title>ToDo App</title>
</head>
<body>
    <div class="root" ng-app="myapp">
        <div class="navigation"></div>

        <header class="hero">
                <h1 class="title">Welcome To ToDo List</h1>
        </header>

        <div class="content" ng-controller="content_ctrl">
            <section class="tools">
                <input class="input text" type="text" ng-model="t_name">
                <button class="btn" ng-click="addTask()">Add Task</button>
            </section>

            <section>
                <ul class="list">
                    <li class="task" ng-repeat="i in task_list track by $index">
                        <label for="{{i}}"><p class="name text">{{i}}</p></label>
                        <button class="check" id="{{i}}"
ng-click="completeTask(task_list, i)">Complete</button>
                    </li>
                </ul>
            </section>
        </div>
    </div>

</body>
</html>
```

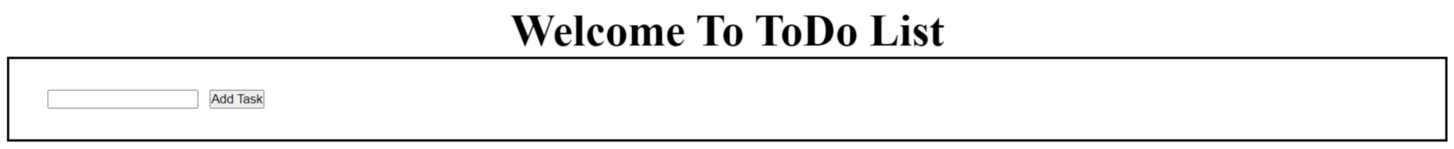
css file

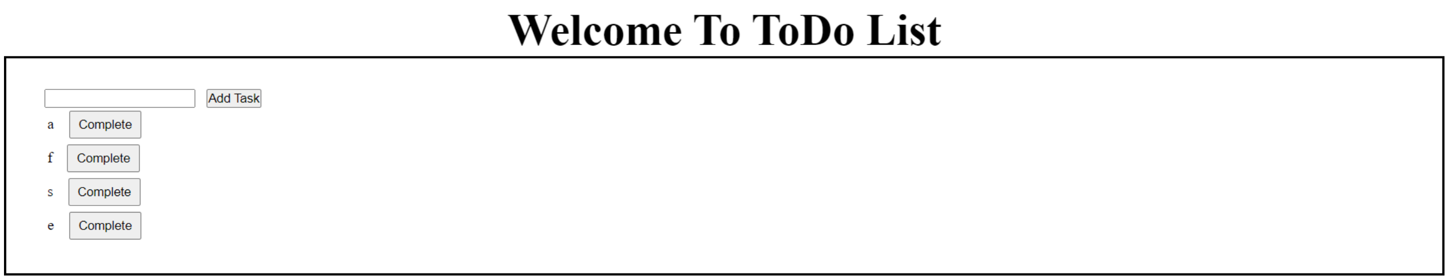
```
* {
    margin: 0px;
    padding: 0px;
}

.content {
    padding: 2rem;
    border: 3px solid black;
}

.list {
    display: flex;
    flex-direction: column;
    list-style-type: none;
}

.task {
    display: flex;
    flex-direction: row;
    justify-content: flex-start;
    align-items: baseline;
    margin: 0.2rem;
}

.text {
    margin: 0 0.5rem;
}

.title {
    text-align: center;
    font-size: 3rem;
    margin: 2rem 0 0 0;
}

.check {
    margin: 0 0.5rem;
    padding: 0.3rem 0.5rem;
}
```

js file

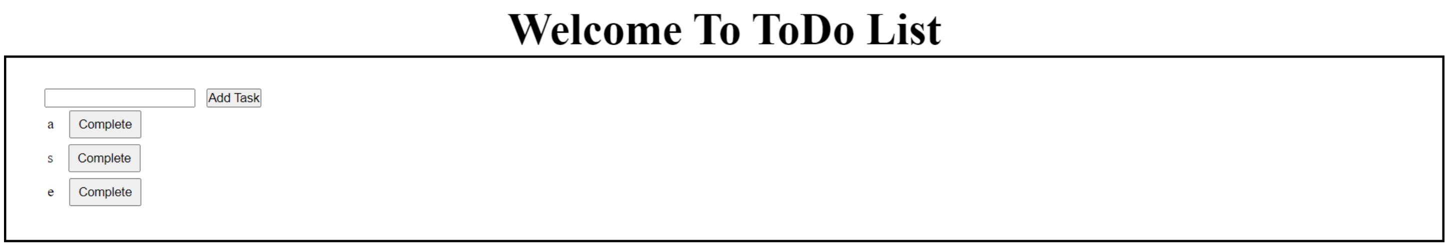
```
var myapp= angular.module("myapp", ["ngRoute"]);

myapp.service("ToDoService", function() {
    this.addTsk= function(arr, t) {
        arr.push(t);
        return(arr);
    }
    this.deleteTsk= function(arr, val) {
        for(let i=0; i< arr.length; i++) {
            if(arr[i]=== val) {
                arr.splice(i, 1);
                break;
            }
        }
        return(arr);
    }
});

myapp.controller("content_ctrl", ($scope, ToDoService) => {
    $scope.task_list= new Array();
    $scope.t_name= "";

    $scope.addTask= function() {
        if($scope.t_name) {
            ToDoService.addTsk($scope.task_list, $scope.t_name);
        }
        $scope.t_name= "";
    }

    $scope.completeTask= function(arr, val) {
        console.log("Removing ", val);
        ToDoService.deleteTsk($scope.task_list, val);
    }
});
```

<u>output</u>


Welcome To ToDo List



Welcome To ToDo List



Welcome To ToDo List