

What is scope in angular JS?

The Scope in Angular JS is the binding part between HTML (view) and JavaScript (controller) and it is a built-in object. It contains application data and objects. It is available for both the view and the controller. It is an object with available properties and methods.

Scope angular जेएस में एचटीएमएल (व्यू) और जावास्क्रिप्ट (controller) के बीच बाध्यकारी(binding) हिस्सा है और यह एक built-in object है। इसमें एप्लिकेशन डेटा और ऑब्जेक्ट शामिल हैं। यह व्यू और कंट्रोलर दोनों के लिए उपलब्ध है। यह उपलब्ध properties और Methods के साथ एक वस्तु है।

There are two types of scopes in Angular JS.

- \$Scope
- \$rootScope

1) **\$scope**: The \$scope in an AngularJS is a built-in object, which contains application data and methods. You can create properties to a \$scope object inside a controller function and assign a value or function to it.

The \$scope is glue between a controller and view (HTML). It transfers data from the controller to view and vice-versa.

angularJS में \$scope एक बिल्ट-इन ऑब्जेक्ट है, जिसमें एप्लिकेशन डेटा और Methods शामिल हैं।

आप controller फ़ंक्शन के अंदर \$scope ऑब्जेक्ट में गुण बना सकते हैं और इसे मान या फ़ंक्शन असाइन कर सकते हैं।

\$scope एक controller और view (HTML) के बीच glue है। यह controller से डेटा को देखने के

लिए transfer

करता है ।



```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
<script src="angular.js"></script>
</head>
<body ng-app="myNgApp">
  <div ng-controller="myController">
    Message: <br />
    {{ message }}<br />
  /div>
  <script>
    var ngApp = angular.module('myNgApp', []);
    ngApp.controller('myController', function ($scope) {
      $scope.message = "Hello World!";
    });
  </script>
</body>
</html>

```

2) **\$rootScope**: An AngularJS application has a single \$rootScope. All the other \$scope objects are child objects.

The properties and methods attached to \$rootScope will be available to all the controllers.

एक AngularJS एप्लिकेशन में एक एकल \$rootScope होता है। अन्य सभी \$scope ऑब्जेक्ट

चाइल्ड ऑब्जेक्ट हैं।

\$rootScope से जुड़े गुण और विधियाँ सभी नियंत्रकों के लिए उपलब्ध होंगी।

```
<!doctype html>

<html>

<head>

<script src="angular.js"></script>

</head>

<body ng-app="MyApp">

<div ng-controller="ParentController">

Controller Name: {{controllerName}}<br/>

Message: {{Message}}<br/>

<div ng-controller="childController" style="margin:10px 0 10px 20px;">

Controller Name: {{controllerName}} <br/>

Message: {{Message}} <br/>

</div>

</div>

<div ng-controller="outerController">

Controller Name: {{controllerName}} <br/>

Message: {{Message}}<br/>

</div>

<script>

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

//parent controller

app.controller('ParentController', function($scope, $rootScope){

$rootScope.Message="Hello i am controlled and available to all because i am defined

within rootScope";

$scope.controllerName="ParentController";
```

```

    });

    //child controller
    app.controller('childController', function($scope){
        $scope.controllerName="childController";
        $scope.Message="Hello world";
    });

    //outer controller
    app.controller('outerController', function($scope){
        $scope.controllerName="outer Controller";
    });
</script>
</body>
</html>

```

How to use the Scope?

When we make a controller in AngularJS, we will pass the \$scope object as an argument. In AngularJS, it creates and injects a different \$scope object into each controller in an application. Thus, the data and methods attached to \$scope inside one controller cannot be accessed on another controller.

- The scope is the model and it is a JavaScript object with properties and methods which are available for both the view and the controller.

जब हम AngularJS में कंट्रोलर बनाते हैं, तो हम \$scope ऑब्जेक्ट को एक तर्क के रूप में पास करेंगे। AngularJS में, यह एक एप्लिकेशन में प्रत्येक नियंत्रक में एक अलग \$scope ऑब्जेक्ट बनाता है और इंजेक्ट करता है। इस प्रकार, एक नियंत्रक के अंदर \$scope से जुड़े डेटा और विधियों को दूसरे नियंत्रक पर एक्सेस नहीं किया जा सकता है।

- दायरा मॉडल है और यह एक जावास्क्रिप्ट ऑब्जेक्ट है जिसमें गुण और विधियां हैं जो दृश्य और नियंत्रक दोनों के लिए उपलब्ध हैं।

