

AngularJS Scope

- The scope is the binding part between the HTML (view) and the JavaScript (controller).
- The scope is an object with the available properties and methods.
- The scope is available for both the view and the controller.

How to Use the Scope?

When you make a controller in AngularJS, you pass the `$scope` object as an argument:

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

<div ng-app="myApp" ng-controller="myCtrl">

<h1>{{carname}}</h1>

</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.carname = "Volvo";
});
</script>

<p>The property "carname" was made in the controller, and can be referred to in
the view by using the {{ }} brackets.</p>

</body>
</html>
```

Output:

Volvo

The property "carname" was made in the controller, and can be referred to in the view by using the `{{ }}` brackets.

When adding properties to the `$scope` object in the controller, the view (HTML) gets access to these properties.

In the view, you do not use the prefix `$scope`, you just refer to a property name, like `{{carname}}`.

Understanding the Scope

If we consider an AngularJS application to consist of:

- View, which is the HTML.
- Model, which is the data available for the current view.
- Controller, which is the JavaScript function that makes/changes/removes/controls the data.

Then the scope is the Model.

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

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

<div ng-app="myApp" ng-controller="myCtrl">

<input ng-model="name">

<h1>My name is {{name}}</h1>

</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.name = "John Doe";
});
</script>

<p>When you change the name in the input field, the changes will affect the
model, and it will also affect the name property in the controller.</p>

</body>
</html>
```

Output:

My name is John Doe

When you change the name in the input field, the changes will affect the model, and it will also affect the name property in the controller.



My name is Awad Bin-Jawed

When you change the name in the input field, the changes will affect the model, and it will also affect the name property in the controller.

