Child Scope IN ANGULARJS

- Scopes In AngularJS support **Hierarchies/inheritance.**
- Each Angular application may have several **child scopes**, but it has exactly one **root scope.**
- At the point when new scopes are created, they are including as children of their parent scopes.
- It is makes a tree structure which parallel the DOM where they are connected.
- AngularJS support **prototypical** inheritance. That means, if we try to access a property defined on the parent scope from the child scope, JavaScript will first search the child scope, if no such property is found, it searches the parent scope and so on until the root scope is reached.
- The **child scopes** prototypically inherit from their patents scope.

 © Copyright 2016. All Rights Reserved.

Sample coding for Scope Hierarchies in AngularJS:

```
<!DOCTYPE html>
<html>
	<head>
		<title>Wikitechy AngularJS Tutorials</title>
	</head>
		<script src="https://ajax.googleapis.com/ajax/libs/
		angularjs/1.5.6/angular.min.js"></script>
	<body ng-app="myApp" >
		<h2> Wikitechy Scope Hierarchies in AngularJS  </h2>
		<div ng-controller="controller1">
			<h3>Hello {{ city }}!</h3>
		</div>
		<div ng-controller="controller2">
			<ol>
				<li ng-repeat="name in names">
				  {{ name }} from {{ country }}, city: {{ city }}
				</li>
			</ol>
		</div>
		<script>
			angular.module('myApp', [ ]);
			.controller('controller1',[ '$scope', '$rootScope' ,
			 function($scope, $rootScope) {
			    $scope.names=['Jasmine', 'Mark'];
			    $rootScope.country = 'UK';
			    $scope.city='London';
			}])
			.controller('controller2', ['$scope', function($scope) {
			    $scope.names = ['Adam', 'Angel', 'Leo'];
			}]);
		</script>
	</body>
</html>
```

 © Copyright 2016. All Rights Reserved.

Data:

- The data been defined for our AngularJS Application.

```
names = ['Adam', 'Angel', 'Leo'];
city='London';
names=['Jasmine', 'Mark'];
country = 'UK';
```

Logic:

- Controller logic for the AngularJS application

```
angular.module('myApp', [ ]);
.controller('controller1',[ '$scope', '$rootScope' ,
  function($scope, $rootScope) {
    $scope.names=['Jasmine', 'Mark'];
    $rootScope.country = 'UK';
    $scope.city='London';
}])
.controller('controller2', ['$scope', function($scope) {
   $scope.names = ['Adam', 'Angel', 'Leo'];
}]);
```

HTML:

- Viewable HTML contents in AngularJS Application.

```
<body ng-app="myApp" >
        <h2> Wikitechy Scope Hierarchies in AngularJS  </h2>
        <div ng-controller="controller1">
           <h3>Hello {{ city }}!</h3>
        </div>
        <div ng-controller="controller2">
           <ol>
              <li ng-repeat="name in names">
                        {{ name }} from {{ country }}, city: {{ city }} </li>
           </ol>
        </div>
```

 © Copyright 2016. All Rights Reserved.

Code Explanation for Scope Hierarchies in AngularJS:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Wikitechy AngularJS Tutorials</title>
    <script src="https://ajax.googleapis.com/ajax/libs/
      1       angularjs/1.5.6/angular.min.js">
    </script>
  </head>
  <body ng-app="myApp">
    <h2>Wikitechy Scope Hierarchies in AngularJS</h2>
    <div  ng-controller="controller1">
4     <h3>Hello {{city}}!</h3>
    </div>
    <div  ng-controller="controller2">
      <ol>
6       <li ng-repeat="name in names">
7         {{name}} from {{country}}, city: {{city}}
        </li>
      </ol>
    </div>
    <script>
      angular.module('myApp', [])
9     .controller('controller1',['$scope', '$rootScope',
      function($scope, $rootScope) {
        $scope.names=['Jasmine','Mark'];
        $rootScope.country = 'UK';
        $scope.city='London';
      }])
11    .controller('controller2', ['$scope', function($scope) {
12      $scope.names = ['Adam', 'Angel', 'Leo'];
      }]);
    </script>
  </body>
</html>
```

1. AngularJS is distributed as a JavaScript file, and can be added to a HTML page with a <script> tag.

 © Copyright 2016. All Rights Reserved.

2. The AngularJS application is defined by ng-app="**myApp".** The application runs inside the <body> tag. It's also used to define a <body> tag as a root element.
3. The ng-controller=**" controller1"** is an AngularJS directive. It is used to define a controller name as **"controller1".**
4. The **{{ city }}** is used to bind the city name, which is defined in the controller1 in JavaScript.
5. The ng-controller=**" controller2"** is an AngularJS directive. It is used to define a controller name as **"controller2".**
6. The ng-repeat directive is used to repeat a set of list element from an **names** array which is declared in controller2.
7. Here the **{{name}}** and **{{ country}}** values are bind and the **{{city}}** value is not bind because the name is declared in **controller2** and the country is declared in **controller1** as a **rootScope**. But the **city** value is declared in controller1 as a **scope** element.
8. angular.module function is used to create a module. Here we have passed an empty array to it.
9. Here we have declared a controller module using **.controller()** function. The value of the controller modules is stored in scope object. In AngularJS, **$scope and $rootScope** are passed as first argument to **.controller()** during its constructor definition.
10. Here we have set the value for scopes and root scopes.

 © Copyright 2016. All Rights Reserved.

Scope declarations

$scope.names as ['Jasmine', 'Mark'];

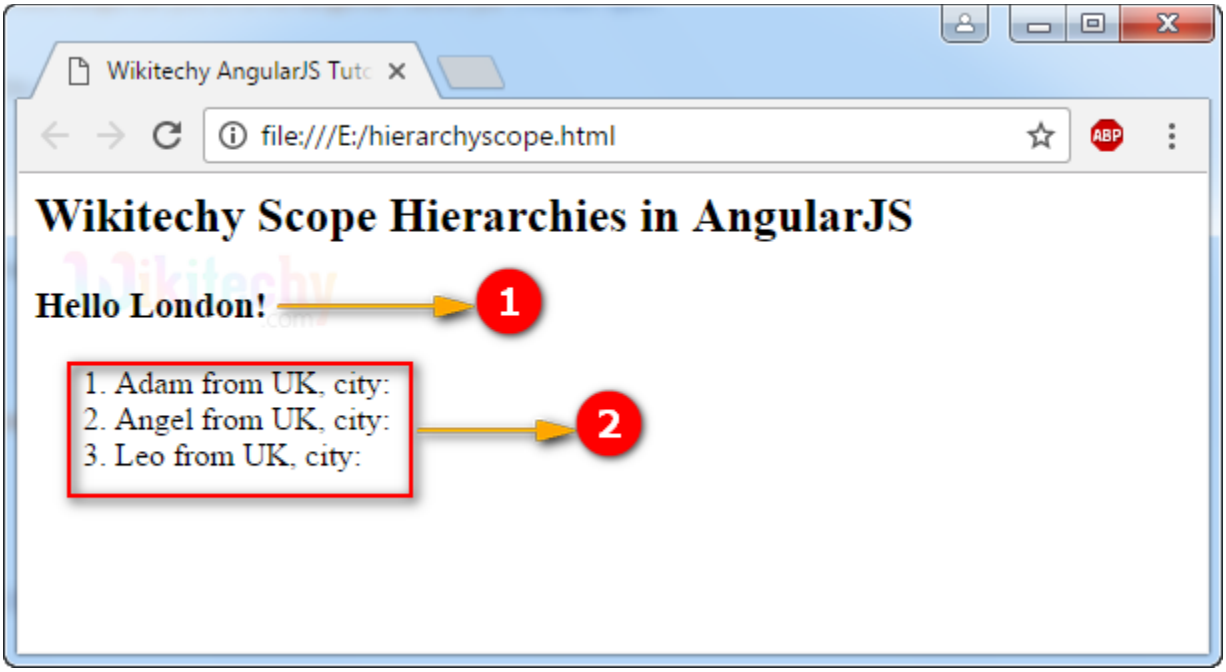
$scope.city as 'London';

Root scope declaration : we can use globally

$rootscope.country as UK;

11. Here we have declared a controller module using **.controller()** function. The value of the controller modules is stored in scope object. In AngularJS, **$scope** is passed as first argument to **.controller()** during its constructor definition.
12. Here we have set the value of **$scope.names** as **['Adam', 'Angel', 'Leo'];**

Sample Output for Scope Hierarchies in AngularJS:



 © Copyright 2016. All Rights Reserved.

1. The output displays a city value as **"London"** which is defined in **controller1** module as well as the value is accessed to HTML element which have the **controller1** as their controller.
2. The output displays a **name** and **country** value which are accessed from **controller2** and **controller1** to HTML element which have a **controller2** as their controller. Here the **country** value is inherited from controller 1 because it has been defined as a root scope, when it come for the city we can't inherit the values because it is declared under the scope and not in the root scope.

 © Copyright 2016. All Rights Reserved.