

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

لیست انتخاب (Dropdown) در AngularJS

مدرس : مهندس افشین رفوآ

[دوره آموزشی AngularJS](#)

لیست انتخاب (Dropdown) در AngularJS

AngularJS به شما این امکان را می دهد تا بر اساس لیستی از عناصر درون یک آرایه و یا یک شی (object)، یک dropdown بسازید.

ساختن یک dropdown با استفاده از ng-options

اگر می خواهید با استفاده از یک object یا یک آرایه در AngularJS یک dropdown بسازید باید از دایرکتیو ng-option استفاده کنید. به عنوان مثال:

```
<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <select ng-model="selectedName" ng-options="x for x in names"></select>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.names = ["Emil", "Tobias", "Linus"];
    });
  </script>
  <p>This example shows how to fill a dropdown list using the ng-options directive.</p>
</body>
</html>
```

شما همچنین می توانید از دایرکتیو **ng-repeat** برای ساخت یک **dropdown** مشابه استفاده کنید:

```
<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <select>
      <option ng-repeat="x in names">{{x}}</option>
    </select>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.names = ["Emil", "Tobias", "Linus"];
    });
  </script>
  <p>This example shows how to fill a dropdown list using the ng-repeat directive.</p>
</body>
</html>
```

دایرکتیو **ng-repeat** یک بلاک از کد **HTML** را برای هر عنصر داخل آرایه تکرار می کند و به همین دلیل می توان از آن برای ساخت گزینه های **یک dropdown** استفاده کرد، اما دایرکتیو **ng-options** به صورت اختصاصی برای ساختن گزینه های **یک dropdown** ساخته شده است و همچنین یک مزیت بزرگ نسبت به **ng-repeat** دارد. **Dropdown** هایی که با **ng-options** ساخته می شوند، گزینه های انتخاب شده را یک **object** و همچنین **dropdown** هایی که با **ng-repeat** ساخته می شوند گزینه های انتخاب شده را یک رشته در نظر می گیرند.

برای چه باید استفاده کنیم؟

فرض کنید مانند شکل زیر یک آرایه ای از **object** ها به صورت زیر در اختیار داریم:

```
$scope.cars = [
  { model: "Ford Mustang", color: "red" },
  { model: "Fiat 500", color: "white" },
  { model: "Volvo XC90", color: "black" }
];
```

دایرکتیو **ng-repeat** محدودیت هایی دارد و هر گزینه انتخاب شده را یک رشته در نظر می گیرد:

```

<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <p>Select a car:</p>
    <select ng-model="selectedCar">
      <option ng-repeat="x in cars" value="{{x.model}}">{{x.model}}</option>
    </select>
    <h1>You selected: {{selectedCar}}</h1>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.cars = [
        { model: "Ford Mustang", color: "red" },
        { model: "Fiat 500", color: "white" },
        { model: "Volvo XC90", color: "black" }
      ];
    });
  </script>
  <p>When you use the ng-repeat directive to create dropdown lists, the selected value must be a string.</p>
  <p>In this example you will have to choose between the color or the model to be your selected value.</p>
</body>
</html>

```

زمانی که از دایرکتیو **ng-options** استفاده می کنید، هر گزینه انتخاب شده را **یک object** در نظر می گیرید:

```

<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <p>Select a car:</p>
    <select ng-model="selectedCar" ng-options="x.model for x in cars"></select>
    <h1>You selected: {{selectedCar.model}}</h1>
    <p>It's color is: {{selectedCar.color}}</p>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.cars = [
        { model: "Ford Mustang", color: "red" },
        { model: "Fiat 500", color: "white" },
        { model: "Volvo XC90", color: "black" }
      ];
    });
  </script>
  <p>When you use the ng-options directive to create dropdown lists, the selected value can be an object.</p>

```

```
<p>In this example you can display both the model and the color of the selected element.</p>
</body>
</html>
```

گزینه های انتخاب شده به صورت **object**، اطلاعات بیشتری درباره ی آن گزینه در خود نگه داری می کند و برنامه ی شما را انعطاف پذیرتر خواهد کرد. در این مقاله ما از دایرکتیو **ng-options** استفاده می کنیم.

منبع داده (Data Source) به صورت یک **object**

در مثال هایی که در بالا گفته شد، **data source** ما یک آرایه بود اما ما می توانیم از **object** هم به عنوان **data source** استفاده کنیم.

فرض کنید یک **object** جفت مقدار ارزشی (**key-value pairs**) در اختیار داریم:

```
$scope.cars = {
  car01: "Ford",
  car02: "Fiat",
  car03: "Volvo"
};
```

بیان اتریبوت **ng-options** برای **object** ها مقداری متفاوت است. در مثال زیر، از یک **object** به عنوان **data source** استفاده شده است و **x** بیانگر کلید (**key**) و **y** بیانگر مقدار می باشد:

```
<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <p>Select a car:</p>
    <select ng-model="selectedCar" ng-options="x for (x, y) in cars"></select>
    <h1>You selected: {{selectedCar}}</h1>
  </div>
  <p>This example demonstrates the use of an object as the data source when creating a dropdown list.</p>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.cars = {
        car01: "Ford",
        car02: "Fiat",
        car03: "Volvo"
      }
    });
  </script>
```

```
</script>
</body>
</html>
```

در **key-value pair** ، گزینه انتخاب شده همیشه **value** می باشد. در **key-value pair** می تواند یک **object** باشد. در مثال زیر خواهید دید که مقدار انتخاب شده در **key-value pair** ، **value** و فقط یک مرتبه به صورت **object** خواهد بود.

```
<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <p>Select a car:</p>
    <select ng-model="selectedCar" ng-options="x for (x, y) in cars"></select>
    <h1>You selected: {{selectedCar.brand}}</h1>
    <h2>Model: {{selectedCar.model}}</h2>
    <h3>Color: {{selectedCar.color}}</h3>
    <p>Note that the selected value represents an object.</p>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.cars = {
        car01: { brand: "Ford", model: "Mustang", color: "red" },
        car02: { brand: "Fiat", model: "500", color: "white" },
        car03: { brand: "Volvo", model: "XC90", color: "black" }
      }
    });
  </script>
</body>
</html>
```

گزینه های داخل **dropdown** در **key-value pair** نیستند ولی می توانند **value** یا ویژگی **value** یک **object** باشند:

```
<!DOCTYPE html>
<html>
<head>
  <title>داده تحلیل آموزشگاه</title>
  <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>
<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    <p>Select a car:</p>
    <select ng-model="selectedCar" ng-options="y.brand for (x, y) in cars"></select>
    <h1>You selected: {{selectedCar.brand}}</h1>
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.cars = {
        car01: { brand: "Ford", model: "Mustang", color: "red" },
        car02: { brand: "Fiat", model: "500", color: "white" },
        car03: { brand: "Volvo", model: "XC90", color: "black" }
      }
    });
  </script>
</body>
</html>
```

```
<h2>Model: {{selectedCar.model}}</h2>
<h3>Color: {{selectedCar.color}}</h3>
<p>The visible text inside the dropdown list can also be a property of the value
object.</p>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function ($scope) {
  $scope.cars = {
    car01: { brand: "Ford", model: "Mustang", color: "red" },
    car02: { brand: "Fiat", model: "500", color: "white" },
    car03: { brand: "Volvo", model: "XC90", color: "black" }
  }
});
</script>
</body>
</html>
```

