

یک نکته‌ای که در توسعه سیستم‌ها و نرم افزارها تاکید فراوانی به آن می‌شود استفاده مجدد از کدهای نوشته شده قبلی است. یعنی تا جای ممکن باید ساختار پروژه به گونه‌ای نوشته شود که از تکرار کدها در جای جای پروژه جلوگیری شود. این مورد به خوبی در زبان‌های شیء‌گرا نظیر C# رعایت می‌شود اما در پروژه‌هایی که مبتنی بر Javascript هستند نظیر angular، باید با استفاده از خاصیت prototype جاوا اسکریپت این مورد را رعایت نمود. در [مقاله](#) Dr. Axel Rauschmayer، قدم به قدم و به خوبی روش‌های وراثت در Javascript توضیح داده شده است.

در [این پست](#) با روش‌های وراثت در کنترلرهای انگولار آشنا شدید. این وراثت محدود به ارث بری scope می‌شود. اما یکی از بخش‌های بسیار مهم پروژه‌های انگولار نوشتن سرویس‌هایی با قابلیت توسعه مجدد در سایر بخش‌های پروژه می‌باشد. معادل آن، مفهوم Overriding در OOP است. با ذکر مثالی این مورد را با هم بررسی خواهیم کرد. ابتدا یک سرویس به نام BaseService ایجاد کنید:

```
angular.module('myApp').service('BaseService', function() {
    var BaseService = function(title) {
        this.title = title;
    };

    BaseService.prototype.getMessage = function() {
        var self = this;
        return 'Hello ' + self.title;
    };

    return BaseService;
});
```

سرویس بالا دارای سازنده‌ای است که مقدار title باید در اختیار آن قرار گیرد. با استفاده از خاصیت prototype تابعی تعریف می‌کنیم که این تابع خروجی مورد نظر را برای ما تامین خواهد نمود. حال اگر ماژول و کنترلری جهت نمایش خروجی به صورت زیر ایجاد کنیم:

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

app.controller('myCtrl', function ($scope,BaseService) {
    var instance = new BaseService('Masoud');
    $scope.title = instance.getMessage();
});
```

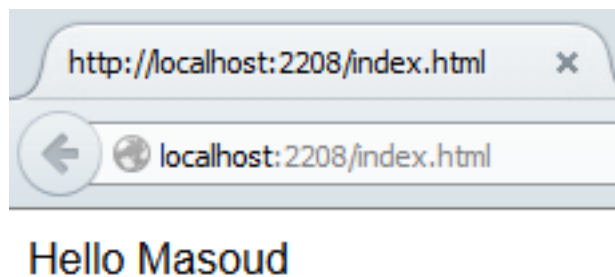
با کدهای Html زیر:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" ng-app="myApp">
<head>
    <title></title>
</head>
<body ng-controller="myCtrl">
    <div>
        {{title}}
    </div>

</body>
<script src="Scripts/jquery-2.1.1.min.js"></script>
<script src="Scripts/angular.js"></script>
<script src="App/app.js"></script>
```

</html>

در نهایت خروجی به صورت زیر قابل مشاهده است:



تا اینجا کار روال معمول تعاریف سرویس در انگولار بوده است. اما قصد داریم سرویس جدیدی را ایجاد نمایم تا خروجی سرویس قبلی را اندکی تغییر دهد. به جای اینکه سرویس قبلی را تغییر دهیم یا بدتر از آن سرویس جدیدی بسازیم و کدهای قبلی را در آن کپی کنیم کافیهست به صورت زیر عمل نماییم:

```
app.service('ExtService', function(BaseService) {
  var ExtService = function() {
    BaseService.apply(this, arguments);
  };

  ExtService.prototype = new BaseService();

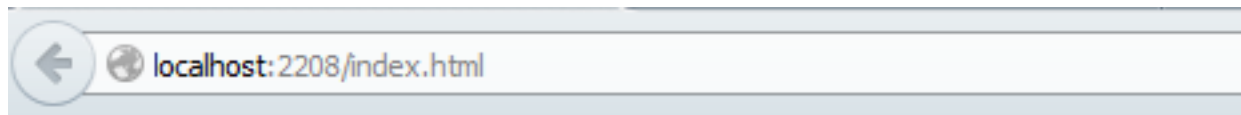
  ExtService.prototype.getMessage = function() {
    var self = this;
    return BaseService.prototype.getMessage.apply(this, arguments) + ' From Ext Service';
  };

  return ExtService;
});
```

حال می‌توان کنترلر را به صورت زیر بازنویسی کرد.

```
app.controller('myCtrl', function ($scope, BaseService , ExtService) {
  var baseInstance = new BaseService('Masoud');
  var extInstance = new ExtService('Dotnettips');
  $scope.title = baseInstance.getMessage() + ' and ' + extInstance.getMessage();
});
```

در کنترلر بالا هر دو سرویس تزریق شده‌اند. خروجی سرویس دوم متن From Ext Service را نیز به همراه خواهد داشت. پس از اجرای برنامه خروجی زیر قابل مشاهده است:



Hello Masoud and Hello Dotnettips From Ext Service