

خیلی خوشحالم که تا این مرحله، [این مقاله‌ها](#) را دنبال می‌کنید. [در مقالات قبل](#) مسائل ساده و مهمی در بحث Performance مطرح شد. در این مقاله می‌خواهم قدم سوم در بهبود Performance را توضیح دهم که رعایت کردن این مسائل می‌تواند کمک زیادی در بهبود عملکرد برنامه‌های مبتنی بر AngularJS داشته باشد.

scope؟

همه‌ی برنامه نویسان و توسعه دهندگان، یکی از اولین مفاهیمی را که در AngularJS یاد می‌گیرند، scope هست. اما scope چیست؟ به صورت خیلی ساده می‌توان گفت scope مشخص کننده‌ی حوزه متغیرها و توابعی هست که قرار است در View تاثیر داشته باشند. کد زیر را مشاهده کنید:

```
<div>{{name}} نام و نام خانوادگی</div>
<div>{{avg()}} معدل</div>
```

و کد سمت controller

```
scope.nums=[19,20,17,16,15,18,19];
scope.name='بهنام محمدی';
scope.avg= function(){
    return scope.nums.reduce(function(previousValue, currentValue) {
        return previousValue + currentValue;
    })/scope.nums.length;
}
```

و اما خروجی نهایی

نام و نام خانوادگی: بهنام محمدی
معدل: 17.71

خوب چون ما در قسمت controller به صورت scope.name و scope.avg عمل کرده‌ایم، می‌توانیم در View به صورت name و avg از این متغیرها استفاده کنیم. در نتیجه اگر ما در controller، به جای scope.name بنویسیم name و یا به جای scope.avg بنویسیم avg به مشکل بر می‌خوریم؛ چون قسمت View ما متغیرهای داخل scope را در View دخیل می‌کند و متغیرهای داخل scope توسط Watcherها رصد می‌شود.

خوب سؤال، همه چیز که عالی هست پس مشکل در کجاست؟

مشکل در متغیر scope.nums هست! به کد زیر توجه کنید:

```
var nums=[19,20,17,16,15,18,19];
scope.name='بهنام محمدی';
scope.avg= function(){
    return nums.reduce(function(previousValue, currentValue) {
        return previousValue + currentValue;
    })/nums.length;
}
```

فکر کنم متوجه تفاوت این کد با کد بالایی شده‌اید. اما کدام کد درست است؟ یا بهتر بگوییم کدام کد بر روی Performance تاثیر مناسبی دارد؟ کد پایینی Performance بالایی دارد. دلیل این موضوع این است وقتی ما از nums در View هیچ استفاده‌ای نمی‌کنیم، بهتر است به صورت var nums تعریف شود. در کد بالایی که این متغیر به صورت scope.nums تعریف شده بود، با اینکه در View استفاده نشده بود، ولی توسط AngularJS Watcher در هر لحظه رصد می‌شود و این کار باعث کندی و کاهش عملکرد AngularJS خواهد شد. بنابراین در کل متغیرهایی را که در View استفاده نمی‌کنید، به صورت var test استفاده نمایید تا AngularJS Watcher این متغیرها را رصد نکند.

امیدوارم از این مقاله لذت برده باشید. منتظر مقاله بعدی من باشید.