

حتماً تا به حال در وب سایت‌های زیادی قسمت‌هایی را دیده‌اید که چیدمان عناصر آن به شکل زیر است:



این گونه چیدمان را حتماً در منوی Start ویندوز 8 بارها دیده‌اید! عناصر تشکیل دهنده‌ی این شکل از چیدمان، می‌توانند یک سری عکس باشند که تشکیل یک گالری عکس را داده‌اند و یا یک سری div که محتوای پست‌های یک وبلاگ را در خود جای داده‌اند. چیزی که این شکل از چیدمان عناصر را نسبت به چیدمان‌های معمول متمایز می‌کند این است که طول و عرض هر یک از این عناصر با یکدیگر متفاوت است و هدف از این گونه چیدمان آن است که این عناصر در فضایی که به آن‌ها اختصاص داده شده است، به صورت بهینه قرار گیرند تا کمترین فضا هدر رود.

برای اعمال این شکل از چیدمان در دنیای وب افزونه‌های زیادی بر فراز کتابخانه‌ی jQuery تدارک دیده شده است که از جمله مطرح‌ترین آن‌ها می‌توان به افزونه‌های [Masonry](#)، [Isotope](#) و [Gridster](#) اشاره کرد.

افزونه‌ی Isotope مزایایی را برای من در پی داشت و این افزونه را برای انجام کارهای خود، مناسب دیدم. نکته‌ی مهم اینجا است که هدف من بررسی Isotope نیست، چرا که اگر به وب سایت آن مراجعه کنید، با کوهی از مستندات مواجه می‌شوید که چگونه از آن در وب سایت‌های معمولی استفاده کنید.

در این مقاله قصد من این است که نشان دهم چگونه از افزونه‌ی Isotope در AngularJS استفاده کنیم؛ چگونه چیدمان آن را راست به چپ کنیم و چگونه آن را با محیط‌های واکنش گرا (Responsive) سازگار کنیم.

فرض کنید در یک وب سایت قصد داریم اطلاعات یک سری مطلب خبری را از سرور، به فرمت JSON دریافت کرده و نمایش دهیم. در AngularJS شیوه‌ی کار بدین صورت است که اطلاعاتی که به فرمت JSON هستند را با استفاده از directive ایی به نام

ng-repeat پیمایش کرده و آن‌ها را نمایش دهیم. حال اگر بخواهیم چیدمان مطالب را با استفاده از Isotope تغییر دهیم، می‌بینیم که هیچ چیزی نمایش داده نمی‌شود. دلیل آن بر می‌گردد به مراحل کامپایل کردن AngularJS و نامشخص بودن زمان اعمال چیدمان Isotope به عناصر است.

در AngularJS هنگامیکه با دستکاری DOM سر و کار پیدا می‌کنیم، معمولاً باید به سراغ Directive‌ها رفت و یک Directive سفارشی برای کار با Isotope تعریف کرد تا با مکانیزم‌های Angular سازگار باشد. خوشبختانه [Isotope Directive برای Angular](#) موجود می‌باشد. نکته‌ی مهم این است که این Directive برای نگارش 1 افزونه‌ی Isotope نوشته شده است. البته با نگارش 2 هم کار می‌کند که من برای انجام کار خود نسخه‌ی 1 را ترجیح دادم استفاده کنم.

نکته‌ی بعدی که باید رعایت شود این است که چیدمان عناصر باید از راست به چپ شوند. خوشبختانه این کار در نسخه‌ی 1 Isotope با [تغییر کوچکی در سورس Isotope](#) و تغییر یک تابع انجام می‌شود. گویا نسخه‌ی دوم [امکان پیش فرضی](#) را برای این کار دارد، اما نتوانستم آن را به خوبی پیاده سازی کنم و به همین دلیل ترجیح دادم از همان نسخه‌ی اول استفاده کنم.

برای اینکه در هنگام جابه جا شدن عناصر، انیمیشن‌ها نیز از راست به چپ انجام شوند، باید css‌های زیر را نیز اعمال نمود:

```
.isotope .isotope-item {
  -webkit-transition-property: right, top, -webkit-transform, opacity;
  -moz-transition-property: right, top, -moz-transform, opacity;
  -ms-transition-property: right, top, -ms-transform, opacity;
  -o-transition-property: right, top, -o-transform, opacity;
  transition-property: right, top, transform, opacity;
}
```

Responsive بودن این عناصر مسئله‌ی دیگری است که باید حل گردد. امروزه اکثر فریم ورک‌های مطرح css، واکنشگرا نیز هستند و برای پشتیبانی از سایزهای متفاوت صفحه نمایش، تدابیری در نظر گرفته‌اند. اساس کار واکنش گرا بودن این فریم ورک‌ها در تعیین ابعاد عناصر، بیان ابعاد به صورت درصدی است. مثلاً فلان عرض div برابر 50% باشد بدین معناست که همیشه عرض این div نصف عرض عنصر والد آن باشد.

متأسفانه Isotope میانه‌ی چندانی با این ابعاد درصدی ندارد و باید عرض عناصر به صورت دقیق و بر حسب پیکسل بیان شود. البته نسخه‌ی جدید آن و یا حتی پلاگین‌هایی برای کار با ابعاد درصدی نیز تدارک دیده شده است که به شخصه به نتیجه‌ی با کیفیتی نرسیدم.

برای حل این مشکل می‌توان از امکانات CSS به مانند دستورات زیر استفاده کرد:

```
@media (min-width: 768px) and (max-width: 980px) {
  .card {
    width: 320px;
  }
}

@media (min-width: 980px) and (max-width: 1200px) {
  .card {
    width: 260px;
  }
}

@media (min-width: 1200px) {
  .card {
    width: 340px;
  }
}
```

بدین صورت می‌توان در ابعاد مختلف نمایشگر تعیین کرد که عرض عناصر ما چقدر باشد.

اکنون یک گالری عکس را در نظر بگیرید که در زیر هر عکس توضیحی نیز نوشته شده است و ساختار HTML آن به این صورت است که داخل هر div عکسی نیز موجود است. اگر به شیوه‌ی ذکر شده عمل کنید با یک اشکال مواجه می‌شوید و عناصر روی هم قرار گرفته و اصطلاحاً overlapping می‌افتد. دلیل این امر این است که لود شدن عکس‌ها عملی زمان گیر است و

Isotope قبل از این که عکس لود شود، سایز آن عنصر را محاسبه کرده که در حقیقت این سایز بدون احتساب سایز عکس است و ابعاد واقعی عنصر ما نیست؛ در نتیجه وقتی عکس لود می‌شود آن div فضای بیشتری احتیاج دارد و به همین دلیل به زیر divهای دیگر می‌رود.

برای حل این مشکل باید به این صورت عمل کرد که وقتی عکس‌ها کامل لود شدند، Isotope وارد عمل شده و سایز عناصر را به دست آورده و آن‌ها را بچیند. برای این کار معمولا از افزونه‌ی [imagesLoaded](#) استفاده می‌کنند که با کمک این افزونه می‌توان مشخص کرد که وقتی تمام عکس‌های موجود در فلان div کامل لود شدند، Isotope وارد عمل شده و عناصر را چیدمان کند. البته بدون استفاده از افزونه‌ی imagesLoaded و به کمک امکانات AngularJS و تعریف یک Directive سفارشی می‌توان زمان لود شدن عکس‌ها را کنترل کرد.

```
app.directive('imageOnload', function () {
    return {
        restrict: 'A',
        link: function (scope, element, attrs) {
            element.bind('load', function () {
                scope.$emit('iso-method', { name: 'reLayout', params: null }); // call reLayout
            });
        }
    };
});
```

کار این directive این است که به ازای بارگذاری هر عکس، متد reLayout را از Isotope، فراخوانی می‌کند. از این جهت فراخوانی reLayout به ازای لود شدن هر عکس بهتر است که لود شدن تمامی عکس‌ها ممکن است مدت زمان زیادی طول بکشد و کاربر برای مدتی با یک ساختار بهم ریخته مواجه شود.

اگر در نمونه کدی که قرار داده‌ام، به انتهای کدهای کنترلر ListController دقت کنید، برای رویداد resize شی window، تابعی تعریف شده است تا به هنگام تغییر سایز صفحه فراخوانی شود. در این رویداد هر بار که سایز پنجره تغییر کرد، پس از یک ثانیه تابع reLayout افزونه‌ی Isotope را فراخوانی می‌کنیم تا مجدداً المنت‌های صفحه چیده شوند. البته ضرورتی وجود نداشته ولی در بعضی مواقع عناصر خوب چیده نمی‌شدند که با فراخوانی reLayout از چیدمان صحیح عناصر مطابق با سایز جدید صفحه اطمینان حاصل پیدا می‌کنیم. دلیل یک ثانیه تأخیر این است که اگر به ساز و کار تعاریف متدها در directive Isotope دقت کنید، از سرویس \$timeout به وفور استفاده شده است. ظاهراً اگر برای فراخوانی reLayout زودتر عمل کنیم با فراخوانی‌هایی این متد در ساختار خودش تداخل پیدا می‌کند.

```
$(window).resize(function () {
    $timeout(function myfunction() {
        $scope.$broadcast('iso-method', { name: 'reLayout', params: null }); // call
        reLayout isotope methode prevent overlaaping the items
    },1000);
});
```

در نهایت تمامی نکات گفته شده را به صورت یک نمونه کد آماده کردم: [دانلود نمونه کد](#)

0-گوگل قصد ندارد تولید نکسوس ها را متوقف کند



طنی ماههای اخیر بارها شاهد انتشار شایعاتی مبنی بر عدم تمایل گوگل نسبت به تولید و توسعه ابزارهای سری نکسوس بودیم. ابزارهایی که اندروید خالص بر روی آن‌ها نصب شده و به سفارش گوگل توسط شرکتی سخت‌افزاری این کمپانی تولید می‌شوند. حال گوگل این شایعات را بطور کامل رد کرده و گفته است که تولید نکسوس‌ها را ادامه خواهد داد.

1-چگونه اندروید L را روی نکسوس 5 یا نکسوس 7 نصب کنیم؟



طنی ماههای اخیر بارها شاهد انتشار شایعاتی مبنی بر عدم تمایل گوگل نسبت به تولید و توسعه ابزارهای سری نکسوس بودیم. ابزارهایی که اندروید خالص بر روی آن‌ها نصب شده و به سفارش گوگل توسط شرکتی سخت‌افزاری این کمپانی تولید می‌شوند. حال گوگل این شایعات را بطور کامل رد کرده و گفته است که تولید نکسوس‌ها را ادامه خواهد داد.

3-شاتر: تصاویری دیدنی از احساسات والدین در قلمرو حیوانات



طنی ماههای اخیر بارها شاهد انتشار شایعاتی مبنی بر عدم تمایل گوگل نسبت به تولید و توسعه ابزارهای سری نکسوس بودیم. ابزارهایی که اندروید خالص بر روی آن‌ها نصب شده و به سفارش گوگل توسط شرکتی سخت‌افزاری این کمپانی تولید می‌شوند. حال گوگل این شایعات را بطور کامل رد کرده و گفته است که تولید نکسوس‌ها را ادامه خواهد داد.

2-مقایسه‌ی تصویری اندروید L با اندروید کیتکت



طنی ماههای اخیر بارها شاهد انتشار شایعاتی مبنی بر عدم تمایل گوگل نسبت به تولید و توسعه ابزارهای سری نکسوس بودیم. ابزارهایی که اندروید خالص بر روی آن‌ها نصب شده و به سفارش گوگل توسط شرکتی سخت‌افزاری این کمپانی تولید می‌شوند. حال گوگل این شایعات را بطور کامل رد کرده و گفته است که تولید نکسوس‌ها را ادامه خواهد داد.

4-اپل قیمت آپل تاج را ضمن مجهز کردن نسخه‌ی 16 گیگابایتی به دوربین کاهش داد



5-تماشا کنید: بازگرداندن توانایی کنترل ماهیچه‌ها به افراد فلج به کمک فناوری نوروبریج