



黑客派

CSS3 3D transform 变换实现简易 3D 动画

作者: [zjhch123](#)

原文链接: <https://hacpai.com/article/1472888131285>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<h2 id="CSS3-3D-transform变换制作简易动画">CSS3 3D transform 变换制作简易动画</h2>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
</script>
(adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h2 id="0--感谢">0. 感谢</h2>
<p>感谢<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.zhangxinxu.com%2F" target="_blank" rel="nofollow ugc">张鑫旭</a>老师的博文<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.zhangxinxu.com%2Fwordpress%2F2012%2F09%2Fss3-3d-transform-perspective-animate-transition%2F" target="_blank" rel="nofollow ugc">吧, CSS3 3D transform 变换, 不过如此! </a>对我的指导。看完这篇文章之后, 我对 CSS 3D transform 有了蛮多的理解, 并且自己制作了一个小 demo。在这里和大家分享一下我的理解以及制作过程。</p>
<h2 id="1--效果展示">1. 效果展示</h2>
<p><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2F139.129.132.196%2Fshow%2F3d-slide-view%2F" target="_blank" rel="nofollow ugc">demo 效果</a></p>
<p></p>
<h2 id="2--先看3D-transform">2. 先看 3D transform</h2>
<h3 id="1--方法">1. 方法</h3>
<p>3D transform 中主要包含这几个方法</p>
<ul>
<li><code>rotateX(angle)</code></li>
<li><code>rotateY(angle)</code></li>
<li><code>rotateZ(angle)</code></li>
<li><code>translateX(x)</code></li>
<li><code>translateY(y)</code></li>
<li><code>trasnlateZ(z)</code></li>
</ul>
<p>其分别表示</p>
<ul>
<li>绕 X 轴旋转</li>
<li>绕 Y 轴旋转</li>
<li>绕 Z 轴旋转</li>
<li>沿 X 轴平移</li>
<li>沿 Y 轴平移</li>
<li>沿 Z 轴平移</li>
</ul>
<p>其写法例如: </p>
<p><code>transform: rotateX(20deg) translateZ(300px)</code></p>
<h3 id="2--轴-">2. 轴? </h3>
<p></p>
<p>说实话我第一次看见这个图的时候也是很难理解, 也是多亏了<a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.zhangxinxu.com%2F" target="_blank" rel="nofollow ugc">张鑫旭</a>老师的简单易懂的例子, 加上之后自己的一些实践才慢慢掌握。</p>
<h4 id="1--x轴-y轴">1. x 轴, y 轴</h4>
<p>假设我们此时正面向电脑, 目光直视电脑屏幕的正中。</p>
<p></p>  
<p>红线即代表 X 轴，黄线就代表 Y 轴。</p>  
<p>沿 X 轴平移就表示一个物体在屏幕上进行左右方向上的移动。同理可以想象沿 Y 轴平移</p>  
<p>沿 X 轴旋转，则可以想象我的电脑屏幕，以红线为转轴进行转动。同理可以想象沿 Y 轴旋转</p>  
<h4 id="2--z轴">2. z 轴</h4>  
<p>z 轴概念有些抽象。我们可以想象有一根坐标轴从电脑屏幕的正中，垂直于电脑屏幕向外射出，就是 Z 轴。</p>  
<p>沿 Z 轴平移，直观的说就是一个物体在你眼前忽近忽远。(将电脑拿近一点，其实就是电脑在沿着 Z 轴移动)</p>  
<p>沿 Z 轴旋转，直观的说就是一个物体绕着你的视线旋转。(站在一个闹钟的前方，我们可以看见针分针秒针正绕着 Z 轴不断旋转)</p>  
<h4 id="3--从CSS上来看简单的3D效果">3. 从 CSS 上来看简单的 3D 效果</h4>  
<p><a href="https://link.hacpai.com/forward?goto=http%3A%2F%2F139.129.132.196%2Fshow%2Fsimple-3d%2F3d.html" target="\_blank" rel="nofollow ugc">demo 源代码</a></p>  
<h5 id="1--rotate旋转">1. rotate 旋转</h5>  
<p></p>  
<h5 id="2--translate平移">2. translate 平移</h5>  
<p></p>  
<h3 id="3--perspective属性">3. perspective 属性</h3>  
<h4 id="1--解读">1. 解读</h4>  
<p>perspective 的中文意思为透视。</p>  
<p>perspective 属性的存在与否决定了你所看见的效果是 2D 的还是 3D 的。</p>  
<p>在上面"从 CSS 上来看简单的 3D 效果"的例子中，如果我们把 perspective 属性全部删除，则到的效果会变成这样</p>  
<p></p>  
<p>这样就没有 3D 效果了。</p>  
<p>perspective 属性的值表示视距。简单的说，在上面的例子中，我将 perspective 属性的值设置了 300px，则这些物体的 3d 效果就相当于人眼在距离他们 300px 的位置上观察到的效果。</p>  
<h4 id="2--再看translateZ">2. 再看 translateZ</h4>  
<p>大家都知道，对于一个物体来说，人的直观感受就是“近大远小”。</p>  
<p>在页面上，我们已经对一个元素实现了“透视”，那么怎么样才能让这个元素离我们更“近”呢</p>  
<p>答案很简单，就是通过 translateZ 属性。</p>  
<p>在上文中讲过，translateZ 表示将一个物体沿着 Z 轴平移，Z 轴是垂直于电脑平面的轴，沿着 Z 轴平移的意思也就是将这个物体“拿近”或者“拿远”。</p>  
<p>我们可以使用开发者工具对 <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2F139.129.132.196%2Fshow%2Fsimple-3d%2F3d.html" target="\_blank" rel="nofollow ugc">demo 源代码</a>里的 translateZ 元素进行修改，修改其 translateZ 的值，我们可以发现，这个元素大小“变化”了。</p>  
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>  
<!-- 黑客派PC帖子内嵌-展示 -->  
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>  
</script>  
(adsbygoogle = window.adsbygoogle || []).push({);  
</script>  
<p></p>

<p>当然，这种大小变化并不是说这个元素真实的 width 和 height 发生了变化，而是反应出一种 “近” 和 “远离” 的变化。</p>  
<p>当我们将 translateZ 的大小，改成 299 的时候，会发现这个元素铺满了整个屏幕。</p>  
<p></p>  
<p>这是因为，我们 perspective 的值设置的是 300，你就想象，我离着 300px 的距离观察一个物是这么大，当我离着 1px 的距离去观察这个物体，那这个物体应该会铺满我的整个视线。</p>  
<p>当 translateZ 的值改为 300 或 300 以上，这个元素 “消失” 了。其实并不是消失，而是因为他 Z 轴平移到了我们的视距之外。</p>  
<p></p>  
<h4 id="3--书写方案">3. 书写方案</h4>  
<p>对于 perspective 的值，有两种书写方案。</p>  
<p>第一种是写在动画元素的父元素上，表示对展示动画元素的舞台，拥有一个视距。</p>  
<p>例如：</p>  
<pre><code class="highlight-chroma">.stage {  
 perspective: 300px;  
}  
</code></pre>  
<p>第二种是写在动画元素上，表示对于每一个动画元素，都拥有一个视距。</p>  
<p>例如：</p>  
<pre><code class="highlight-chroma">.block {  
 transform: perspective(300px) rotateY(30deg);  
}  
</code></pre>  
<p>其两种写法的效果，可以参考我实现的 <a href="https://link.hacpai.com/forward?goto=htt%3A%2F%2F139.129.132.196%2Fshow%2Fperspective%2Fperspective.html" target="\_blank" el="nofollow ugc">perspective 书写 demo</a></p>  
<p>可以看出，这两种写法的效果是不同的。因为如果对于整个舞台我们有一个透视，那么其每个子素的效果就会不一样。如果对于每个子元素都有一个透视，那么其展示效果就会相同(因为在 demo 中 rotateY 的值相同)。</p>  
<h3 id="4--perspective-origin">4. perspective-origin</h3>  
<p>perspective-origin 属性指的是默认我们所观测的中心点。默认是正中，我们也可以进行改变，如：</p>  
<p><code>perspective-origin:30px 30%</code></p>  
<h3 id="5--transform-style">5. transform-style</h3>  
<p>transform-style 属性很重要。在 3d 变换中，我们将其设置为</p>  
<p><code>transform-style: preserve-3d</code></p>  
<p>一般给做 3D 变换的父元素设置这个属性。如果不设置的话，视觉效果会展示 2D 的效果。</p>  
<h3 id="6--backface-visibility">6. backface-visibility</h3>  
<p>该属性用于设置元素的背后是否可见，一般可以设置为</p>  
<p><code>backface-visibility:hidden;</code></p>  
<p>不可见。</p>  
<h2 id="3--实现小动画">3. 实现小动画</h2>  
<p>3D 效果横竖就是前文说的那些属性，只要利用好，可以做出很多酷炫的效果。</p>  
<h3 id="1--静态">1. 静态</h3>  
<p>我这里学习张鑫旭老师的结构，即</p>  
<pre><code class="highlight-chroma">舞台  
 容器  
 block  
 block  
 block  
 ...  
</code></pre>

<p>就像上文说的那样，对于舞台我们先加一个视距</p>  
<p><code>perspective: 800px</code></p>  
<p>对于做 3D 变换的 block 的父元素容器，我们加上</p>  
<p><code>transform-style: preserve-3d</code></p>  
<p>使其展示效果为 3D。</p>  
<p>此时代码应该长这样</p>  
<p></p>  
<p>我们再加上 9 个小方块，并且给舞台，容器，方块设置一些必要的属性。</p>  
<p>首先给小方块设置好长宽高，背景颜色，然后使用 <code>position: absolute</code> 让小方重叠在一起，共用一个中心点。再给舞台和容器设置好一些长宽，位置的属性。因为观测的中心点(perspective-origin)默认是舞台(设置过 perspective 属性的元素)的正中，所以我们要让容器位于舞台正，或者让舞台大小和容器大小一致。这里我选择大小一致的写法</p>  
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>  
<!-- 黑客派PC帖子内嵌-展示 -->  
<ins class="adsbygoogle" style="display: block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>  
<script>  
 (adsbygoogle = window.adsbygoogle || []).push({});  
</script>  
<p></p>  
<p><strong>最重要的要来啦！</strong></p>  
<p>接下来让小方框有 3d 效果！因为有 9 块方块， $360 / 9 = 40$ ，所以每个小方块的角度间隔为 40</p>  
<p></p>  
<p>然后看一下效果</p>  
<p></p>  
<p>好像有什么地方不对...</p>  
<p>这时候应该想到，小方块都黏在了一起，实际效果是小方块应该距离中心有一个距离才对。</p>  
<p>我们使用 translateZ 属性。</p>  
<p>那么 translateZ 属性的值应该设置为多少呢？</p>  
<p>我在这里先设置为 180px。其实是有一个计算的公式的，只要设置的值大于等于我们计算出来的就可以。大家可以自己想想如何计算。</p>  
<p></p>  
<p>看看效果</p>  
<p></p>  
<p>大功告成！静态 3D 效果完成啦！</p>  
<h3 id="2--动态">2. 动态</h3>  
<p><strong>让它动起来！</strong></p>  
<p>首先我们先实现一个绕 Y 轴旋转的动画。</p>  
<p></p>  
<p>然后给容器加上这个动画，让容器“动”起来！</p>  
<p></p>  
<p>之后，如果想实现像我那样一开始飞出的效果，就需要用 js 或者多定义一些动画来实现啦！</p>

<p>大家可以自己尝试实现一下~</p>  
<h3 id="3--Q-A">3. Q&amp;A</h3>  
<p>正在编辑</p>  
<h2 id="4--请尽情使用吧-">4. 请尽情使用吧! </h2>