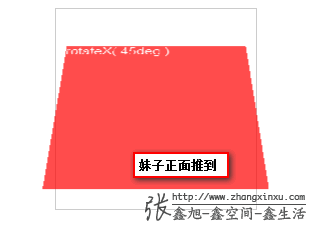
# Css3 3D概念



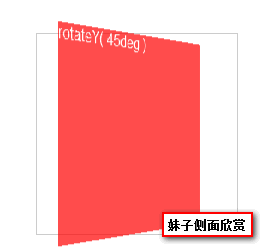
上图是一个正常3D分解图

### 一、3D旋转的3个属性：rotateX, rotateY, rotateZ

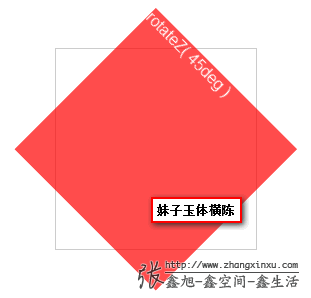
分别对应三个不同的轴的旋转。



rotateX就是把下面妹子推倒。



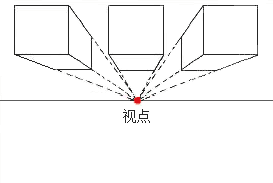
rotateY 就是把妹子转过身来（找好体位）。



rotateZ就是从床上看妹子，如果是电动圆床的话，你懂的。（这里跟2D旋转很像了）

### 属性

Perspective就是透视属性。

 美术上面的透视图。

在CSS上面的透视，指的是拥有该属性的元素与人眼之间的距离。

比方说，一个1680像素宽的显示器中有张美女图片，应用了3D transform，同时，该元素或该元素父辈元素设置的perspective大小为2000像素。则这张美女多呈现的3D效果就跟你本人在1.2个显示器宽度的地方(1680\*1.2≈2000)看到的真实效果一致！！  


### translateZ

translateZ可以理解为透视点事物的距离。

比方说，我们设置元素perspective为201像素，则其子元素，设置的translateZ值越小，则子元素大小越小（因为元素远去，我们眼睛看到的就会变小）；translateZ值越大，该元素也会越来越大，当translateZ值非常接近201像素，但是不超过201像素的时候（如200像素），该元素的大小就会撑满整个屏幕（如果父辈元素没有类似overflow:hidden的限制的话）。因为这个时候，子元素正好移到了你的眼睛前面，所谓“一叶蔽目，不见泰山”，就是这么回事。当translateZ值再变大，超过201像素的时候，该元素看不见了——这很好理解：我们是看不见眼睛后面的东西的！

### Perspective + translateZ 写法

perspective属性有两种书写形式，一种用在舞台元素上（动画元素们的共同父辈元素）；第二种就是用在当前动画元素上，与transform的其他属性写在一起。

例子1：

HTML:

<section id="darkred" class="container">

<div class="box"></div>

</section>

CSS:

#darkred .box {

background-color: darkred;

transform: perspective(600px) rotateY(45deg);

}

透视点与透视事物放到同一样元素上面。

例子2：

HTML：

<section id="darkblue" class="container">

<div class="box"></div>

</section>

CSS：

#darkblue {

perspective: 600px;

}

#darkblue .box {

background-color: darkblue;

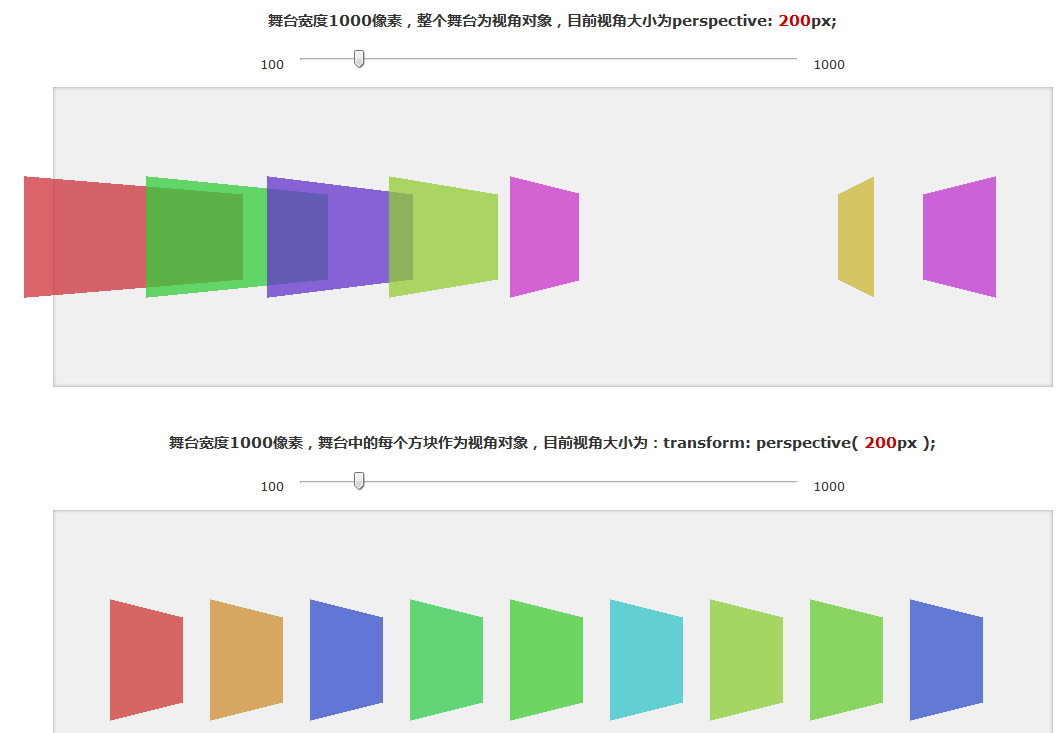
transform: rotateY(45deg);

}

透视点在父级元素上面。

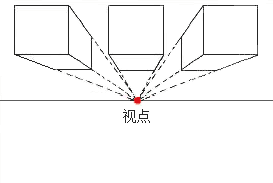
上面的例子虽然同样效果，但是因为只是一个事物所以体现在不出来问题，大家可以看网址http://www.zhangxinxu.com/study/201209/transform-perspective-same-rotate.html

两种写法的区别。



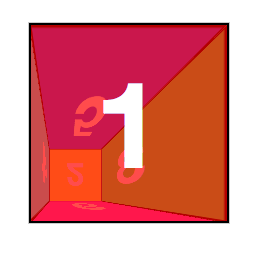
### perspective-origin 透视视点

所有透视都有一个视点，也就是这个透视最终点。

比如这图，想像当视点不是在中心的话而是偏向一边呢？如果我们把左边的立方体看成正面的话，中间跟边的立方体还是现在所看到的样子吗？？

下面为立方体的实际应用透视效果图：

perspective-origin: 25% 75%;



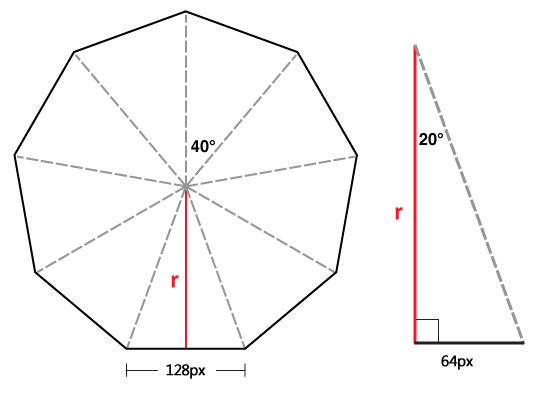
### 不可少的transform-style: preserve-3d

基本上，我们想要根据现实经验实现一些3D效果的时候，transform-style: preserve-3d是少不了的。一般而言，该声明应用在3D变换的兄弟元素们的父元素上，也就是舞台元素。

### backface-visibility

默认下某元素后页面的元素是可以直接透视看到的。这个效果有时候我们需要但如果需要更加贴近实现的话，肯定不能看到给挡住的事物（想想妹子那身衣服）。所以有时候我们就要设置这个值为backface-visibility:hidden;

1. demo计算公式

拿本demo距离，每张美女图片的宽度是128像素，因此，有如下理想方位效果图：  


上图中红色标注的r就是的demo页面中图片要translateZ的理想值（该值可以让所有图片无缝围成一个圆）！

r的计算很简单，有初中数学水平的人应该都会：

r = 64 / Math.tan(20 / 180 \* Math.PI) ≈ 175.8

demo页面为了好看，图片之间留了点间距，使用的translateZ的值为175.8 + 20 = 195.8.  
