**rotate旋转的意思，rotateX旋转X轴，rotateY旋转Y轴，rotateZ旋转Z轴**







**Perspective**

该属性可以定義3D視覺的角度，讓底下子元素使用3D特效時能夠完整展示。

没有透视，不成3D.

CSS3 3D transform的**透视点是在浏览器的前方**！

或者这么理解吧：显示器中3D效果元素的透视点在显示器的上方（不是后面），近似就是我们眼睛所在方位！

translateZ则可以帮你理解透视位置。

我们都知道近大远小的道理，对于没有rotateX以及rotateY的元素，translateZ的功能就是让元素在自己的眼前或近或远。比方说，我们设置元素perspective为201像素，如下：

perspective: 201px;

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使用方法**



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。他们的效果都是一样的，如下代码示例：

.stage {

perspective: 600px;

}

以及：

#stage .box {

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

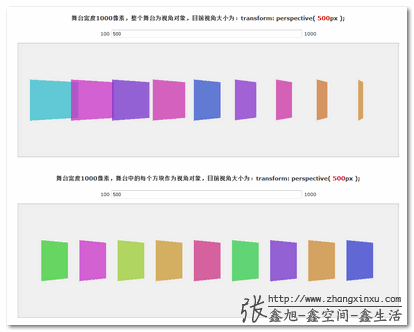
}

不过在chrome中发现一个问题，那就是perspective一定要在rotateY（或rotateX）的前面。如果代码写成下面的情况，perspective的设置会无效。

<div id="animateTest"

style="-webkit-transform: rotateY(40deg) perspective(400px);">

在FireFox中也是这种情况



好吧，图中的效果其实不难理解。上面舞台整个作为透视元素，因此，显然，我们看到的每个子元素的形体都是不一样的；而下面，每个元素都有一个自己的视点，因此，显然，因为rotateY的角度是一样的，因此，看上去的效果也就一模一样了！

另外還有個屬性叫**perspective-origin**  
功能是用來定義X和Y軸為基礎的3D位置(定義初始位置)

表示你那双色迷迷的眼睛看的位置。默认就是所看舞台或元素的中心



**注意：perspective和perspective-origin受影響的是子元素，而非元素本身**

**当视点在距离屏幕为屏幕宽度的1.2倍时，看到的3d物体就跟实物一样**

参考:

<http://www.zhangxinxu.com/wordpress/2012/09/css3-3d-transform-perspective-animate-transition/>

<http://blog.johnsonlu.org/csstransform-3d-perspective/>