1. **动画**
   * @keyframes 规则 用于创建动画。在 @keyframes 中规定某项 CSS 样式，能创建由当前样式逐渐改为新样式的动画效果。
   * animation-name规定 @keyframes 动画的名称。
   * animation-duration规定动画完成一个周期所花费的秒或毫秒。默认是 0
   * animation-timing-function速度曲线
     + ease默认先慢，再快，再慢
     + linear匀速
     + ease-in：由慢到快。
     + ease-out：由快到慢。
     + ease-in-out：由慢到快再到慢。
     + step-start跳到鼠标开始那帧
     + step-end跳到鼠标结束那帧
   * animation-delay规定动画何时开始。默认是 0，允许负值；
   * animation-iteration-count规定动画被播放的次数。默认是 1，
     + infinite无限循环
   * animation-direction规定动画是否在下一周期逆向地播放。默认是 "normal"
     + alternate反向播放
   * animation:name duration timing-function delay iteration-count direction;
   * animation-play-state规定动画是否正在运行或暂停。默认是 "running"。
     + paused暂停
   * animation-fill-mode规定对象动画时间之外的状态。
     + none不改变默认行为
     + forwards当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）。

**注释：**在 @keyframes 中创建动画时，把它捆绑到某个选择器，否则不会产生动画效果。通过规定至少以下两项 CSS3 动画属性，即可将动画绑 定到选择器：

1.规定动画的名称

2.规定动画的时长

Internet Explorer 10、Firefox 以及 Opera 支持 @keyframes 规则和 animation 属性。

Chrome 和 Safari 需要前缀 -webkit-。

Internet Explorer 9，以及更早的版本，不支持 @keyframe 规则或 animation 属性。

1. **3D介绍**



转换是使元素改变形状、尺寸和位置的一种效果。

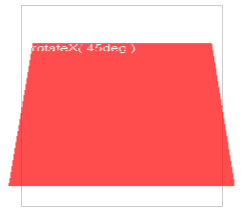
您可以使用 2D 或 3D 转换来转换您的元素。

在咱们的转换概念当中：是没有display这么一说的，

通过改变元素的透明度去实现从无到有

1. **rotateX()方法**（围绕其在一个给定度数X轴旋转的元素）

注意：添加过渡效果比较明显



div{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

transition: all 5s;

}

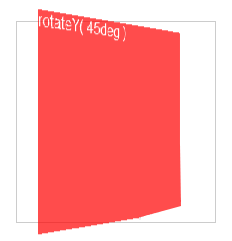
div:hover{

transform:rotateX(-180deg);

-webkit-transform:rotateX(-180deg); /\* Safari and Chrome \*/

}

1. **rotateY()方法**（围绕其在一个给定度数Y轴旋转的元素）



div{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

transition: all 5s;

}

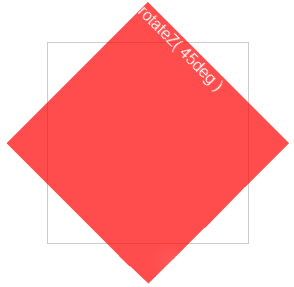
div:hover{

transform:rotateY(-180deg);

-webkit-transform:rotateY(-180deg); /\* Safari and Chrome \*/

}

1. **rotateZ()方法**（围绕其在一个给定度数Z轴旋转的元素 默认值）



div{

width:100px;

height:75px;

background-color:red;

border:1px solid black;

transition: all 5s;

}

div:hover{

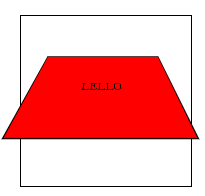
transform:rotateZ(-180deg);

-webkit-transform:rotateZ(-180deg); /\* Safari and Chrome \*/

}

1. perspective景深(value) 离屏幕多远的距离去观察元素，值越大幅度越小

http://www.zhangxinxu.com/study/201209/transform-perspective-same-rotate.html



#div1{

position: relative;

height: 150px;

width: 150px;

margin: 50px;

padding:10px;

border: 1px solid black;

perspective:150px;

-webkit-perspective:150px; /\*Safari and Chrome \*/

}

#div2{

padding:50px;

position: absolute;

border: 1px solid black;

background-color: red;

transform: rotateX(45deg);

-webkit-transform: rotateX(45deg); /\* Safari and Chrome \*/

}

<div id="div1">

<div id="div2">LELLO</div>

</div>

1. translate3d()函数的效果transform:translate3d(30px,30px,200px);

值代表X轴、Y轴、Z轴（正值在前负值在后）

1. transform-style 属性规定如何在 3D 空间中呈现被嵌套的元素。preserve-3d表示3D透视
2. 立方体的绘制

ul{

width: 200px;

height: 200px;

position: absolute;

top: 50%;

left: 50%;

margin-top:-100px;

margin-left:-100px;

list-style: none;

transform-style: preserve-3d;

transition: all 10s;

}

ul li{

width:200px;

height: 200px;

position: absolute;

background: #000000;

opacity: 0.5;

border:1px solid #0000FF;

color: #fff;

}

ul li:nth-child(1){

transform: translateZ(100px);

}

ul li:nth-child(2){

transform: translateZ(-100px);

}

ul li:nth-child(3){

transform: rotateX(90deg) translateZ(100px);

}

ul li:nth-child(4){

transform: rotateX(-90deg) translateZ(100px);

}

ul li:nth-child(5){

transform: rotateY(90deg) translateZ(100px);

}

ul li:nth-child(6){

transform: rotateY(-90deg) translateZ(100px);

}

ul:hover{

transform: rotateX(360deg) rotateY(360deg);

}