logo.png

CSS3

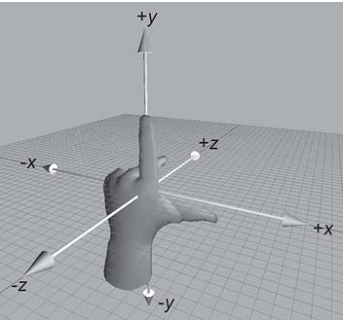
H5 web前端

[www.bufanui.com](http://www.bufanui.com)

# 一、3D转换

## 1.1、左手坐标系

伸出左手，让拇指和食指成“L”形，大拇指向右，食指向上，中指指向前方。这样我们就建立了一个左手坐标系，拇指、食指和中指分别代表X、Y、Z轴的正方向。如下图



## 1.2、CSS中的3D坐标系

CSS3中的3D坐标系与上述的3D坐标系是有一定区别的，相当于其绕着X轴旋转了180度，如下图



借助示例理解3D转换

1. 绕X轴旋转



1. 绕Y轴旋转



1. 绕Z轴旋转



d) 在X轴移动

d) 在Y轴移动

d) 在Z轴移动

## 1.3、透视（perspective）

电脑显示屏是一个2D平面，图像之所以具有立体感（3D效果），其实只是一种视觉呈现，通过透视可以实现此目的。

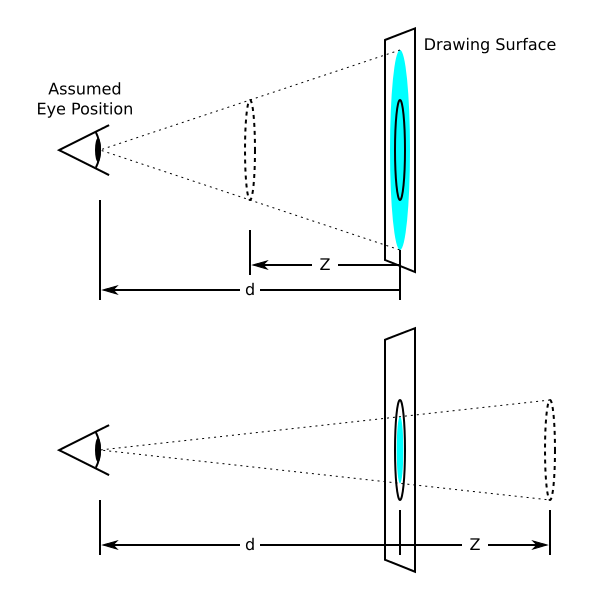
透视可以将一个2D平面，在转换的过程当中，呈现3D效果。

**perspective有两种写法**

a) 作为一个属性，设置给父元素，作用于所有3D转换的子元素

b) 作为transform属性的一个值，做用于元素自身

**5、理解透视距离**



透视会产生“近大远小”的效果

**6、3D呈现（transform-style）**

设置内嵌的元素在 3D 空间如何呈现，这些子元素必须为转换原素。

flat：所有子元素在 2D 平面呈现

preserve-3d：保留3D空间

transform-style.html

3D元素构建是指某个图形是由多个元素构成的，可以给这些元素的父元素设置transform-style: preserve-3d来使其变成一个真正的3D图形。

立方体.html

**7、backface-visibility：**visible/ hidden

设置元素背面是否可见

**[参考文档](http://isux.tencent.com/css3/index.html?transform)**

# 二、动画

动画是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果。

**1、必要元素：**

a、通过@keyframes指定动画序列；

b、通过百分比将动画序列分割成多个节点；

c、在各节点中分别定义各属性

d、通过animation将动画应用于相应元素；

## 2.1关键属性

1. Animation-name

动画名称

2. Animation-duration

动画持续时间

3. animation-timing-function

* linear 匀速。
* ease 缓冲。
* ease-in 由慢到快。
* ease-out 由快到慢。
* ease-in-out 由慢到快再到慢。
* cubic-bezier(number, number, number, number)： 特定的贝塞尔曲线类型，4个数值需在[0, 1]区间内

4. animation-delay

动画延迟（只是第一次）

5. animation-iteration-count

重复次数

infinite

6.animation-direction

动画是否重置后再开始播放

alternate动画直接从上一次停止的位置开始执行，倒着来

normal 动画第二次直接跳到0%的状态开始执行

7.animation-play-state

播放状态（ running 播放 和paused 暂停 ）

8.animation-fill-mode

动画执行完毕后状态

|  |  |
| --- | --- |
| 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）。 |
| backwards | 在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。 |
| both | 设置对象状态为动画结束或开始的状态，结束时状态优先 |

**参数值的顺序：**

关于几个值，除了名字，动画时间，延时有严格顺序要求其它随意

补充：skew

|  |  |
| --- | --- |
| skew(*x-angle*,*y-angle*) | 定义沿着 X 和 Y 轴的 2D 倾斜转换。 |
| skewX(*angle*) | 定义沿着 X 轴的 2D 倾斜转换。 |
| skewY(*angle*) | 定义沿着 Y 轴的 2D 倾斜转换。 |

## 2.2 animate.css

// 标准语法 监听x元素 动画结束的状态

Click mouseover ….   
x.addEventListener("animationend", myStartFunction);

补充 ：动画的事件：

* [animationstart](http://www.runoob.com/jsref/event-animationstart.html) - CSS 动画开始后触发
* [animationiteration](http://www.runoob.com/jsref/event-animationiteration.html) - CSS 动画重复播放时触发
* animationend - CSS 动画完成后触发

过渡有吗？transitionend