

渐变

线性渐变

为了创建一个线性渐变，你必须至少定义两种颜色结点。颜色结点即你想要呈现平稳过渡的颜色。同时，你也可以设置一个起点和一个方向（或一个角度）。

从上到下（默认情况下）预定义方向 to bottom、to top、to right、to left、to bottom right
`background: linear-gradient(direction, color-stop1, color-stop2, ...); background: linear-gradient(to right, red , blue);`

如果你想要在渐变的方向上做更多的控制，你可以定义一个角度， `background: linear-gradient(180deg, red, blue);`

使用透明度（Transparency）

CSS3 渐变也支持透明度（transparency），可用于创建减弱变淡的效果。为了添加透明度，我们使用 `rgba()` 函数来定义颜色结点。`rgba()` 函数中的最后一个参数可以是 0 到 1 的值，它定义了颜色的透明度：0 表示完全透明，1 表示完全不透明。 `background: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1))`

重复的线性渐变 `repeating-linear-gradient()` 函数用于重复线性渐变： `background: repeating-linear-gradient(red, yellow 10%, green 20%);`

CSS3 径向渐变

径向渐变由它的中心定义。径向渐变 - 颜色结点均匀分布（默认情况下） `background: radial-gradient(red, green, blue);`

径向渐变 - 颜色结点不均匀分布

`background: radial-gradient(red 5%, green 15%, blue 60%);`

`shape` 参数定义了形状。它可以是值 `circle` 或 `ellipse`。其中，`circle` 表示圆形，`ellipse` 表示椭圆形。默认值是 `ellipse`。

`background: radial-gradient(circle, red, yellow, green);`

`repeating-radial-gradient()` 函数用于重复径向渐变： `background: repeating-radial-gradient(red, yellow 10%, green 15%);`

CSS3 转换

通过 CSS3 转换，我们能够对元素进行移动、缩放、转动、拉长或拉伸

转换是使元素改变形状、尺寸和位置的一种效果。

`translate()` 方法

通过 `translate()` 方法，元素从其当前位置移动，根据给定的

`left`（x 坐标）和 `top`（y 坐标）位置参数：

1、`translate(x,y)`水平方向和垂直方向同时移动（也就是X轴和Y轴同时移动）

2、translateX(x)仅水平方向移动（X轴移动）

3、translateY(Y)仅垂直方向移动（Y轴移动）

transform: translate(100px, 100px); transform: translateX(100px); transform: translateY(100px);

rotate() 方法

旋转rotate()函数通过指定的角度参数使元素相对原点进行旋转。它主要在二维空间内进行操作，设置一个角度值，用来指定旋转的幅度。如果这个值为正值，元素相对原点中心顺时针旋转；如果这个值为负值，元素相对原点中心逆时针旋转

通过 rotate() 方法，元素顺时针旋转给定的角度。允许负值，元素将逆时针旋转

transform: rotateX(30deg); transform: rotateY(30deg);

scale() 方法

缩放 scale()函数 让元素根据中心原点对对象进行缩放。缩放 scale 具有三种情况： 1、scale(X,Y)使元素水平方向和垂直方向同时缩放（也就是X轴和Y轴同时缩放） 2、scaleX(x)元素仅水平方向缩放（X轴缩放） 3、scaleY(y)元素仅垂直方向缩放（Y轴缩放）

skew() 方法

扭曲skew()函数能够让元素倾斜显示。它可以将一个对象以其中心位置围绕着X轴和Y轴按照一定的角度倾斜。这与rotate()函数的旋转不同，rotate()函数只是旋转，而不会改变元素的形状。skew()函数不会旋转，而只会改变元素的形状。

1、skew(x,y)使元素在水平和垂直方向同时扭曲（X轴和Y轴同时按一定的角度值进行扭曲变形）；

2、skewX(x)仅使元素在水平方向扭曲变形（X轴扭曲变形）；

3、skewY(y)仅使元素在垂直方向扭曲变形（Y轴扭曲变形）

eg: -webkit-transform: skew(45deg); -moz-transform:skew(45deg) transform:skew(45deg);

展示：

<http://fangyexu.com/tool-CSS3Inspector.html>

matrix() 方法

matrix() 方法把所有 2D 转换方法组合在一起。

旋转、缩放、移动以及倾斜元素。matrix() 是一个含六个值的(a,b,c,d,e,f)变换矩阵，用来指定一个2D变换，相当于直接应用一个[a b c d e f]变换矩阵。就是基于水平方向（X轴）和垂直方向（Y轴）重新定位元素,此属性值使用涉及到数学中的矩阵

transform: matrix(水平缩放, 垂直扭曲, 水平扭曲, 垂直缩放, 水平偏移, 垂直偏移)；

<http://blog.csdn.net/henren555/article/details/9699449>

变形--原点 transform-origin 在没有重置transform-origin改变元素原点位置的情况下，CSS变形进行的旋转、位移、缩放，扭曲等操作都是以元素自己中心位置进行变形。但很多时候，我们可以通过transform-origin来对元素进行原点位置改变，使元素原点不在元素的中心位置，以

达到需要的原点位置。当“transition-property”属性设置为all时，表示的是所有中点值的属性。

css3 过渡

它如何工作？

CSS3 过渡是元素从一种样式逐渐改变为另一种的效果。

要实现这一点，必须规定两项内容：

规定您希望把效果添加到哪个 CSS 属性上

规定效果的时长

过渡属性

下面的表格列出了所有的转换属性：

属性 描述

transition 简写属性，用于在一个属性中设置四个过渡属性。

transition-property 规定应用过渡的 CSS 属性的名称。

transition-duration 定义过渡效果花费的时间。默认是 0。

transition-timing-function 规定过渡效果的时间曲线。默认是 "ease"。

transition-delay 规定过渡效果何时开始。默认是 0。

transition-delay属性和transition-duration属性极其类似，不同的是transition-duration是用来设置过渡动画的持续时间，而transition-delay主要用来指定一个动画开始执行的时间，也就是说当改变元素属性值后多长时间开始执行。

有时我们想改变两个或者多个css属性的transition效果时，只要把几个transition的声明串在一起，用逗号（“，”）隔开，然后各自可以有各自不同的延续时间和其时间的速率变换方式。但需要值得注意的一点：第一个时间的值为 transition-duration，第二个为transition-delay。

例如：a{ transition: background 0.8s ease-in 0.3,color 0.6s ease-out 0.3;}