

# 1、变形transform

2016年7月16日 星期六 下午7:29

css3制作动画的几个属性：变形(transform)，过渡(transition)和动画(animation)。

首先介绍transform变形。

transform英文意思：改变，变形。

css3中transform注意包括以下几种：旋转(rotate),扭曲(skew), 缩放(scale)、移动(translate)和矩阵变形matrix。

语法：

```
transform : none | <transform-function> [ <transform-function> ]*  
transform: rotate | scale | skew | translate |matrix;
```

none就是默认值，不进行变形。

**<transform-function>**:表示一个或多个变换函数，以空格分开。即可同时对一个元素进行transform的多种属性操作，例如同时用rotate,scale和translate三种。

```
rotate(<rotate-angle> [<cx> <cy>])  
skewX(<skew-angle>)  
skewY(<skew-angle>)  
scale(<sx> [<sy>])  
translate(<tx> [<ty>])  
matrix(<a> <b> <c> <d> <e> <f>)
```

## 一、旋转rotate

rotate(<angle>) :通过指定的角度参数对元素指定一个2D rotation(2D旋转)，需先有**transform-origin**属性的定义。

transform-origin定义的是旋转的基点，其中angle是指选择角度，正顺时针旋转，负逆时针旋转。





`rotate`是一个很有用的动画，可配合[animation](#)做出非常有意思的效果，[看综合实例效果](#)。

## 二、`translate`平移

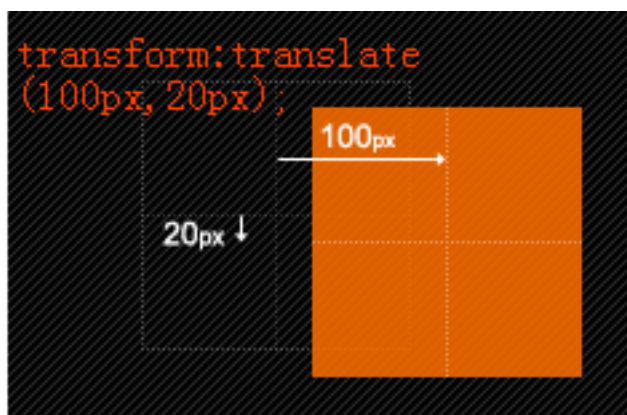
`translate()`函数可以把元素从原来的位置移动，而不影响在x，y轴上的任何web组件，类似于`position:relative`。

`translate()`分三种情况：

### 1、`translate(x,y)`水平和垂直方向同时移动。

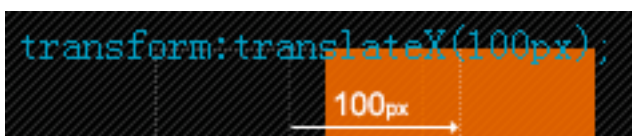
Note:`translate`移动的基点默认为元素中心点，可以根据`transform-origin`改变基点。

如果第二个值没设置，默认为0。



### 2、`translateX(x)`仅水平方向移动。

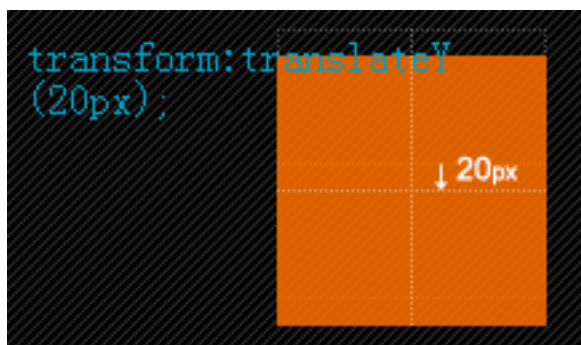
相当于`translate(x,0,)`的简写，基点为元素中心点。





### 3、translate $\mathbf{Y(y)}$ 仅垂直方向移动。

相当于translate(0,y)的简写，基点为元素在中心。



## 三、scale缩放

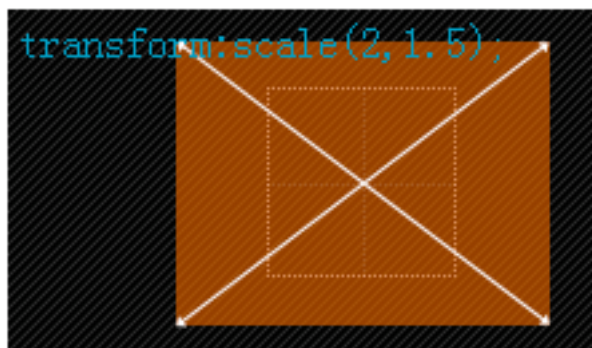
scale缩放和translate移动非常相似，也有三种情况。

缩放中心点：即元素的中心位置

基数：缩放就是既可以缩小，也可以放大；缩放基数为1，大于1放大，小于1缩小。

### 1、scale(x,y)元素在水平和垂直方向同时缩放。

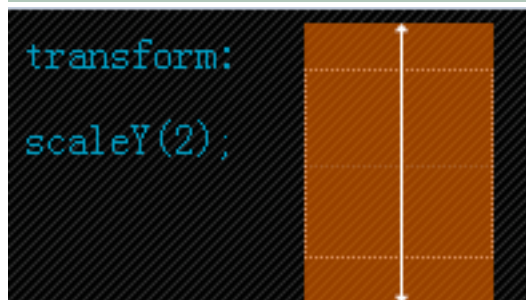
Note: 第二个参数未提供则取与第一个一样的值。



### 2、scale $\mathbf{X(x)}$ x轴缩放。



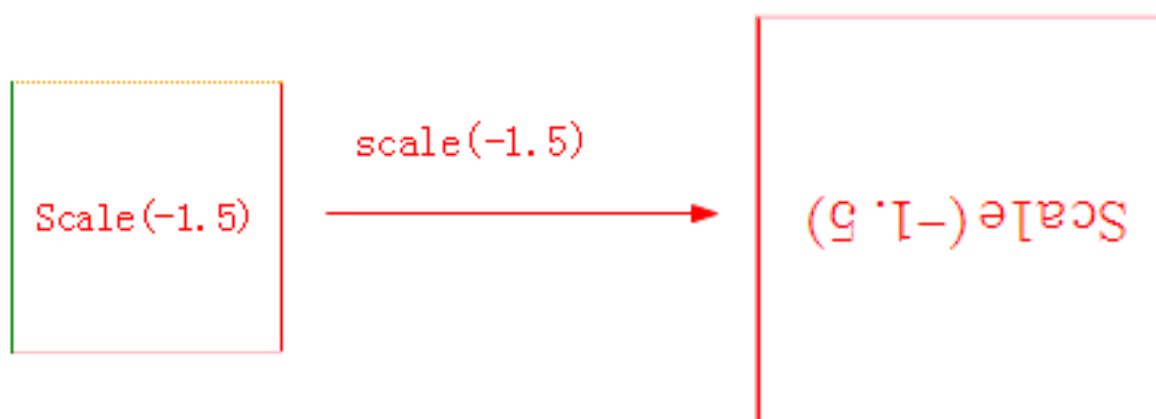
### 3、**scaleY(y)**y轴缩放。



**scale**可以取负值，负值会让元素翻转并缩放。



View Code



## 四、**skew**切变

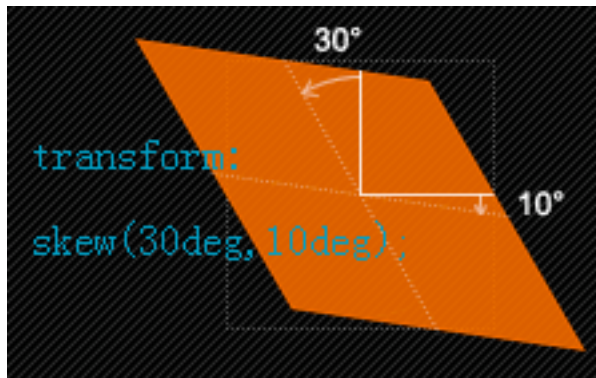
这个属性是用来制作倾斜度的,设计时在2d里面创建3d透视图的时候必须的属性。

**skew**和**translate**、**scale**一样有三种情况。

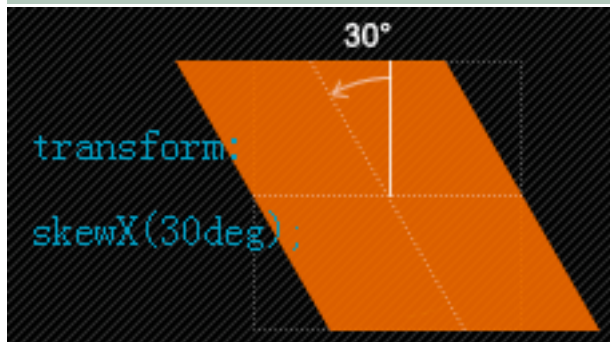
### 1、**skew(x,y)**:x轴和y轴上的**skew transformation**（斜切变换）。

即x轴和y轴同时按照一定的角度值进行扭曲变形。

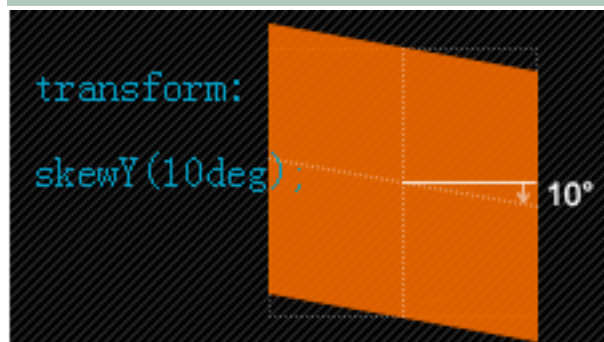
如果第二个参数未提供，则值为0，也就是y轴无斜切。



2、 **skewX(x)**:按给定角度沿x轴指定一个[skew transformation](#)（斜切变换）。



3、 **skewY(y)**:按给定的角度沿Y轴指定一个[skew transformation](#)（斜切变换）。



## 五、矩阵matrix

`matrix(<number>, <number>, <number>, <number>, <number>, <number>)`: 以一个含六值的(a,b,c,d,e,f)变换矩阵的形式指定一个2D变换, 相当于直接应用一个[a,b,c,d,e,f]变换矩阵。就是基于水平方向和垂直方向重新定位元素。

SVG, css3, html5的canvas中都有矩阵变换, 接下来简单说说。

一个元素渲染后就可以得到一张位图, 然后对这个位图上每一点进行变换, 就可以得到新的一张位图, 从而产生平移, 缩放, 旋转, 切变及镜像反射灯效果。

## 1、几个概念

- 矩阵乘法中, 首先要确认两个矩阵是否可以相乘: 只有第一个矩阵的列数等于第二个矩阵的行数, 这样的两个矩阵才能相乘。
- 左乘【前乘】: 即乘在左边, A左乘E即AE。
- 一个m\*n的矩阵左乘一个n\*p的矩阵, 将得到一个m\*p的矩阵。

2D矩阵变换都提供6个参数a,b,c,d,e,f, 基本公式为:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + cy + e \\ bx + dy + f \\ 1 \end{bmatrix}$$

其中, x和y是元素最开始的坐标, x'和y'是矩阵变换后得到的新坐标。

Note: 变换矩阵中a,b,c,d,e,f 6个参数是竖着排的。

$$\begin{aligned} x' &= ax + cy + e \\ y' &= bx + dy + f \end{aligned}$$

## 2、矩阵变换和transform的关系

### a、矩阵和translate平移

$x' = ax + cy + e$ , 我们设 $a=1, c=0$ , 则 $x' = x + e$ ,

$y' = bx + dy + f$ , 同样设 $b=0, d=1$ , 则 $y' = y + f$ 。

这就是`translate(e,f)`了。

所以说`translate(e,f)`就是简化了的变换矩阵`matrix(1,0,0,1,e,f)`,

(x,y) 平移(tx,ty), 就意味着做了一个【1 0 0 1 tx,ty】的矩阵变换。

### b、矩阵和scale缩放



$x' = ax + cy + e$ , 我们设  $c=0, e=0$ , 则  $x' = ax$ ,

$y' = bx + dy + f$ , 我们设  $b=0, f=0$ , 则  $y' = dy$ 。

这就是 `scale(a,d)` 了。

所以说 `scale(a,d)` 就是简化了的变换矩阵 `matrix(a,0,0,d,0,0)`。

$(x,y)$  缩放  $(sx,sy)$ , 就意味着做了一个 **【sx 0 0 sy 0 0】** 的矩阵变换。

### c、矩阵和 **rotate** 旋转

`rotate(a deg)` 等价于 **【cos(a) sin(a) -sin(a) cos(a) 0 0】** 矩阵变换。

### d、矩阵和 **skew** 切变

`skewX(a deg)` 等价于 **【1 0 tan(a) 1 0 0】** 的矩阵变换。

`skewY(a deg)` 等价于 **【1 tan(a) 0 1 0 0】** 的矩阵变换。

所以说 `Matrix` 就是将所有的 2D 效果全部组合在了一起使用。

## 六、transform-origin

前面说了, 元素默认的基点是其中心位置, 可用 `transform-origin` 改变其基点。

使用:

`transform-origin(x,y)`: 用来设置元素的基点 (参考点)。默认点是元素的中心点。x, y 的值可以是百分比, `em`, `px`, 其中 x 也可以是 `left`, `center`, `right`, y 可以是 `top`, `center`, `bottom`, 这点和 [background-position](#) 一样。

关键词	百分比
<code>top = top center = center top</code>	50% 0
<code>right = right center = center right</code>	100%或(100% 50%)
<code>bottom = bottom center = center bottom</code>	50% 100%
<code>left = left center = center left</code>	0 或(0 50%)
<code>center = center center</code>	50%或 (50% 50%)
<code>top left = left top</code>	0 0
<code>bottom right = right bottom</code>	100% 100%

right top = top right	100% 0
bottom right = right bottom	100% 100%
bottom left = left bottom	0 100%

<http://www.cnblogs.com/starof/p/4560076.html>