

CSS3转换

今日主打

CSS3 Transform

CSS3 2D转换

CSS3 3D转换

CSS3 Transform与坐标系统

CSS3 矩阵

CSS3 扩展属性

CSS3 Transform

CSS3的变形 (Transform) 属性

让元素在一个坐标系统中变形。这个属性包含一系列变形函数，可以移动、旋转和缩放元素。

语法

`transform : none | <transform-function> [<transform-function>]*`;

默认值

`transform: none;`

兼容性

IE10+、FireFox16+、Chrome36+、Safari16+、Opera23+

CSS3 2D转换

CSS3 rotate()

CSS3 translate()

CSS3 scale()

CSS3 skew()

CSS3 2D转换

旋转rotate

通过指定的角度参数对原元素指定一个2D rotation (2D 旋转)。

语法

```
transform: rotate(<angle>);
```

参数说明

angle指旋转角度，正数表示顺时针旋转，负数表示逆时针旋转。

举个栗子

CSS3 2D转换

移动translate

`translate()`方法，根据左(X轴)和顶部(Y轴)位置给定的参数，从当前元素位置移动。

三种情况

`translateX(x)`仅水平方向移动（X轴移动）；

`translateY(Y)`仅垂直方向移动（Y轴移动）；

`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）。

CSS3 2D转换

translateX(<translation-value>)

通过给定一个X方向上的数目指定一个translation。

语法

transform: translateX(<translation-value>);

举个栗子

CSS3 2D转换

translateY(<translation-value>)

通过给定一个Y方向上的数目指定一个translation。

语法

transform: translateY(<translation-value>);

举个栗子

CSS3 2D转换

translate(<translation-value>[, <translation-value>])

通过矢量[tx, ty]指定一个2D translation, tx是第一个过渡值参数, ty是第二个过渡值参数选项。

语法

transform: translate(<translation-value>[, <translation-value>]);

举个栗子

CSS3 2D转换

缩放scale

scale()方法，指定对象的2D scale（2D缩放）。

三种情况

scaleX(x)元素仅水平方向缩放（X轴缩放）；

scaleY(y)元素仅垂直方向缩放（Y轴缩放）；

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）。

CSS3 2D转换

scaleX(<number>)

使用 [sx, 1] 缩放矢量执行缩放操作，sx为所需参数。

语法

transform: scaleX(<number>);

举个栗子

CSS3 2D转换

scaleY(<number>)

使用 [1, sy] 缩放矢量执行缩放操作，sy为所需参数。

语法

transform: scaleY(<number>);

举个栗子

CSS3 2D转换

scale(<number>[, <number>])

提供执行[sx,sy]缩放矢量的两个参数指定一个2D scale (2D缩放)。

语法

transform: scale(<number>[, <number>]);

举个栗子

CSS3 2D转换

扭曲skew

skew()方法，指定对象skew transformation（斜切扭曲）。

三种情况

skewX(x)仅使元素在水平方向扭曲变形（X轴扭曲变形）；

skewY(y)仅使元素在垂直方向扭曲变形（Y轴扭曲变形）；

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）。

CSS3 2D转换

skewX(<angle>)

按给定的角度沿X轴指定一个skew transformation (斜切变换)。

语法

transform: skewX(<angle>);

举个栗子

CSS3 2D转换

skewY(<angle>)

按给定的角度沿Y轴指定一个skew transformation (斜切变换)。

语法

transform: skewY(<angle>);

举个栗子

CSS3 2D转换

skew(<angle> [, <angle>])

X轴Y轴上的skew transformation (斜切变换)。第一个参数对应X轴，第二个参数对应Y轴。

语法

transform: skew(<angle> [, <angle>]);

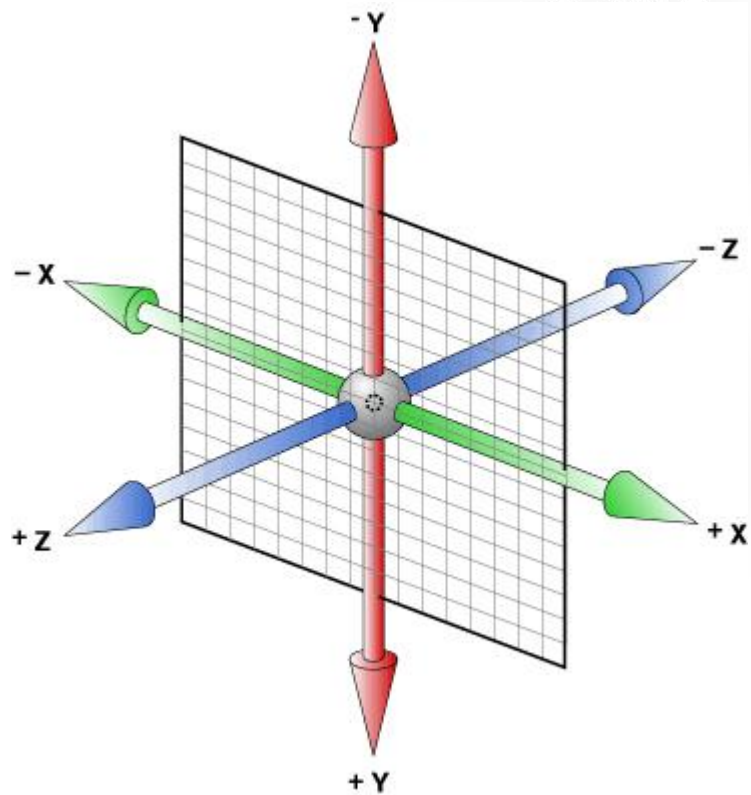
举个栗子

CSS3 3D转换

CSS3 rotate3d()

CSS3 translate3d()

CSS3 scale3d()



CSS3 3D转换

旋转rotateX

rotateX方法指定对象在x轴上的旋转角度。

语法

`transform: rotateX(angle);`

参数说明

angle表示旋转的角度。

举个栗子

CSS3 3D转换

旋转rotateY

rotateY方法指定对象在y轴上的旋转角度。

语法

`transform: rotateY(angle);`

参数说明

angle表示旋转的角度。

举个栗子

CSS3 3D转换

旋转rotateZ

rotateZ方法指定对象在z轴上的旋转角度。

语法

`transform: rotateZ(angle);`

参数说明

angle表示旋转的角度。

举个栗子

CSS3 3D转换

旋转rotate3d

rotate3d方法指定对象的3D旋转角度。

语法

```
transform: rotate3d(x, y, z, angle);
```

参数说明

前3个参数分别表示旋转的方向x,y,z，第4个参数表示旋转的角度，参数不允许省略。

举个栗子

CSS3 3D转换

移动translateZ

translateZ方法指定对象Z轴的平移。

语法

`transform: translateZ(z);`

参数说明

参数对应Z轴，不允许省略。

举个栗子

CSS3 3D转换

移动translate3d

translate3d方法指定对象的3D位移。

语法

`transform: translate3d(x, y, z);`

参数说明

第1个参数对应X轴，第2个参数对应Y轴，第3个参数对应Z轴，参数不允许省略。

举个栗子

CSS3 3D转换

缩放scaleZ

scaleZ方法指定对象的z轴缩放。

语法

`transform: scaleZ(z);`

参数说明

参数对应Z轴，不允许省略。

举个栗子

CSS3 3D转换

缩放scale3d

scale3d方法指定对象的3D缩放。

语法

`transform: scale3d(x, y, z);`

参数说明

第1个参数对应X轴，第2个参数对应Y轴，第3个参数对应Z轴，参数不允许省略。

举个栗子

CSS3 Transform与坐标系统

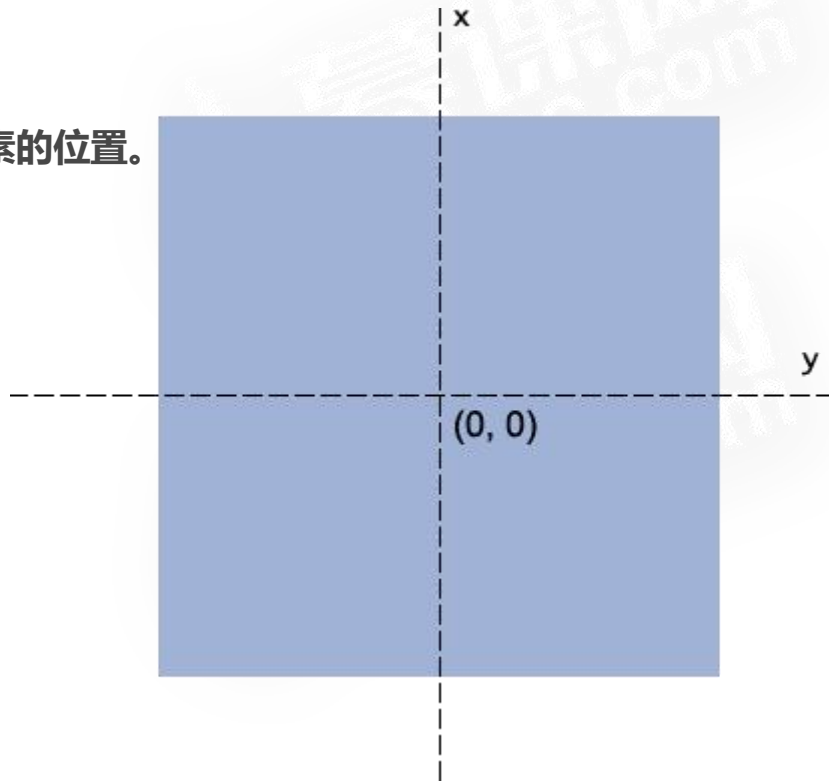
transform-origin属性

transform-origin属性允许您更改转换元素的位置。

语法

transform-origin: x-axis y-axis z-axis;

举个栗子



CSS3 矩阵

矩阵的概念

矩阵可以理解为方阵，只不过，平时方阵里面站的是人，矩阵中是数值。而所谓矩阵的计算，就是两个方阵的人（可以想象成古代的方阵士兵）相互冲杀。

CSS3中的矩阵

CSS3中的矩阵指的是一个方法，书写为`matrix()`和`matrix3d()`；

`matrix`是元素2D平面的移动变换(transform)，2D变换矩阵为3*3；

`matrix3d`是元素3D的移动变换(transform)，3D变换则是4*4的矩阵。

$$\begin{bmatrix} 1 & 2 & 8 \\ 10 & 3 & 9 \\ 7 & 4 & 0 \end{bmatrix}$$

CSS3 矩阵

矩阵matrix

transform: matrix(a, b, c, d, tx, ty);

注意书写方向是竖直方向。

$$\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

转换公式

x、y表示转换元素的所有坐标

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

目标矩阵说明

ax+cy+e为变换后的水平坐标，bx+dy+f表示变换后的垂直位置。

CSS3 矩阵

矩阵matrix举个栗子

`transform: matrix(1, 0, 0, 1, 30, 30);` /* a=1, b=0, c=0, d=1, e=30, f=30 */

根据这个矩阵偏移元素的中心点，假设是(0, 0)，即 $x = 0, y = 0$

变换后， $x = ax + cy + e = 1 * 0 + 0 * 0 + 30 = 30$ ， $y = bx + dy + f = 0 * 0 + 0 * 1 + 30 = 30$

于是，整个元素的中心点从(0, 0)变成了(30, 30)，整个元素就发生了平移

说明

`transform: matrix(1, 0, 0, 1, x, y)`等同于`transform: translate(x, y)`。

注意

matrix在FF浏览器下需要添加单位，webkit内核默认px，translate等方法需要加单位。

CSS3 矩阵

矩阵matrix缩放(scale)

`matrix(sx, 0, 0, sy, 0, 0)` —— `scale(sx, sy)`

矩阵matrix旋转(rotate)

`matrix(cos θ , sin θ , -sin θ , cos θ , 0, 0)` —— `rotate(θ deg)`。

矩阵matrix拉伸(skew)

`matrix(1, tan θ y, tan θ x, 1, 0, 0)` —— `skew(θ xdeg, θ ydeg)`。

CSS3 矩阵

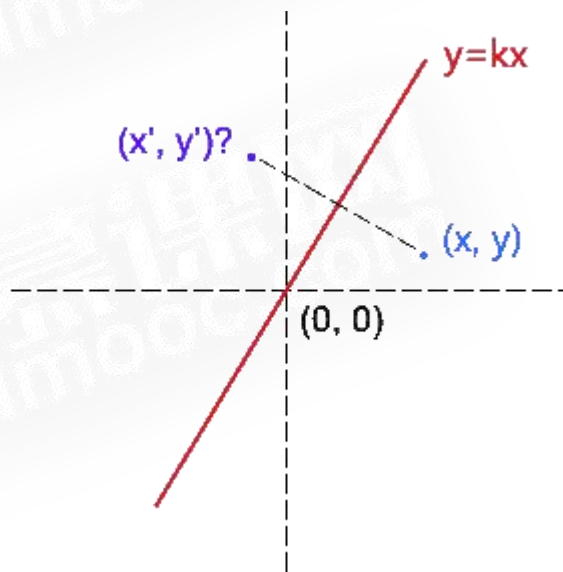
矩阵matrix镜像对称效果

`matrix((1-k*k) / (1+k*k), 2k / (1 + k*k), 2k / (1 + k*k), (k*k - 1) / (1+k*k), 0, 0)`

小小说明

对称轴一定通过元素变换的中心点，k是对称轴的斜率

举个栗子



CSS3 矩阵

3D变换中的矩阵

从二维到三维，是从4到9；而在矩阵里头是从3*3变成4*4，9到16了。

小小例子

`transform: matrix3d(sx, 0, 0, 0, 0, sy, 0, 0, 0, 0, sz, 0, 0, 0, 0, 1);`

$$\begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

CSS3 扩展属性

transform-style属性

transform-style属性指定嵌套元素是怎样在三维空间中呈现。

语法

transform-style: flat|preserve-3d;

默认值

transform-style: flat;

举个栗子

CSS3 扩展属性

perspective属性

指定观察者与「 $z=0$ 」平面的距离，使具有三维位置变换的元素产生透视效果。

语法

`perspective: number|none;`

默认值

`perspective: none;`

举个栗子

CSS3 扩展属性

perspective-origin属性

指定透视点的位置。

语法

`perspective-origin: x-axis y-axis;`

默认值

`perspective-origin: 50% 50%;`

举个栗子

CSS3 扩展属性

backface-visibility属性

指定元素背面面向用户时是否可见。

语法

`backface-visibility: visible|hidden;`

默认值

`backface-visibility: visible;`

举个栗子