

矩阵变换

来自 Safari Developer Library: Safari HTML5 Canvas Guide

本文由 **陶桃猪** 翻□

如需□□及引用

□注明出□

□□

限于本人翻□水平

定有□漏不妥之□

□迎指正

□箱：taotaozhulz@163.com

博客：<http://www.lonemover.com>

2011 年 7 月 25 日

旋转、转移以及放缩都是通过一个变换矩阵实现的，这个 3×3 的矩阵是用来将一个二维的数组转化，比如位图，使用的是线性代数的知识。

对于旋转、转移以及放缩这些常用的转化，已经有一些方便的接口方法，但是你可以使用矩阵变换来实现其他的效果，比如对折或者反射。

设置变换矩阵

你可以通过调用 `context` 的 `setTransform(a,b,c,d,e,f)` 方法来设置变换矩阵，通过将矩阵的前两行新的数值传递进去（第三行永远是 `0 0 1`）。

矩阵中的参数的位置如下图所示：

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

重要：其他的使用变换矩阵的 API 可能放置参数的顺序有所不同。当你从另外一个来源复制矩阵设置的时候一定要检查一下矩阵参数的位置。

当你设置变换矩阵的时候，它会覆盖任何的旋转、转移以及放缩，因为这些设置全部都是使用变换矩阵的。

注意：如果你在设置了变换矩阵之后设置旋转、变换以及转移，变换矩阵被修改了但是没有被重置-操作使用的矩阵数值被修改了，但是其他的矩阵数值并没有改变。

例子：为反射设置矩阵

以 y 轴设置变换矩阵来实现反射，调用 `ctx.setTransform(1,0,0,-1,0,0)`。这样，所有的绘制操作会产生一个上下颠倒的镜像，所有的 y 轴坐标会被乘以 -1 。

以 x 轴设置变换矩阵实现反射，调用 `ctx.setTransform(-1,0,0,1,0,0)`。这样，所有的绘制操作会产生一个镜像，所有的 x 轴坐标会被乘以 -1 。

以 x 和 y 轴设置变换矩阵实现反射，调用 `ctx.setTransform(-1,0,0,-1,0,0)`。这样，所有的绘制操作会产生一个镜像，所有的 x 以及 y 坐标会被乘以 -1 。

下面是一个以 y 轴反射的例子：



The image shows the word "Reflection" written twice. The top instance is in blue, and the bottom instance is in red. The red text is a vertical reflection of the blue text, demonstrating the effect of a y-axis reflection transformation.

```

<html>
<head>
  <title>Reflection Matrix</title>
<script type="text/javascript">
var can;
var ctx;
function init(){
  can = document.getElementById("can");
  ctx = can.getContext("2d");
  ctx.fillStyle="blue";
  ctx.font="48pt Helvetica";
  ctx.fillText("Reflection", 0, 100);
  ctx.setTransform(1,0,0,-1,0,0);
  ctx.fillStyle = "red";
  ctx.fillText("Reflection", 0, -100);
}
</script>
</head>
<body onload="init();">
<canvas id="can" height="200" width="300">
</canvas>
</body>
</html>

```

你可以在设置了反射矩阵之后设置放缩，因为这两个操作归属于矩阵的不同的两个部分。下面的代码加入到之前的例子中，将 y 放大到 1.5 倍并且设置填充的样式为一个渐变的颜色而不是之前固定的颜色，使得镜像变高并且从蓝色渐变到透明白，如下图所示：



增加如下的代码：

```

ctx.scale(1,1.5);
var grad = ctx.createLinearGradient(0,-50,0,-140);
grad.addColorStop(0, 'blue');
grad.addColorStop(1, 'rgba(255,255,255,0)');
ctx.fillStyle = grad;
ctx.fillText("Reflection", 0, -102/1.5);

```

注意：渐变的 y 轴和反射的文本被修改了，是因为变换的矩阵被修改了。y 轴坐标被取负并且除以了 1.5。

元矩阵

如果想要把转换矩阵重置到元矩阵（没有任何变换的矩阵），调用 `ctx.setTransform(1,0,0,1,0,0)`。如下图所示：

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

操作变换矩阵

变换矩阵可以当做一个二维的数组。你可以通过应用另一个矩阵到当前的矩阵来变换当前的矩阵。

为了实现以上的目标，我们可以调用上下文的 `transform(a,b,c,d,e,f)` 方法，将矩阵的前两行数据传递进去（第三行总是 `0 0 1`）。

变换变换矩阵会潜在的改变任何的旋转、放缩以及转移设置。

在变换了变换矩阵之后设置旋转、放缩以及转移会导致矩阵的一些附加的改变-被旋转、放缩、或者转移操作使用过的矩阵数据将会被重写。

变换变换矩阵比较适合给精通线性代数的开发者使用的方法。详细的数学方面的知识已经超出了本文档的范围。