**第8章 像素操作**

对于像素操作，Canvas为我们提供了3种操作方法：getImageData()、putImageData()、createImageData()。

**一、getImageData()方法**

在Canvas中，我们可以使用getImageData()方法来获取一张图片的像素数据。

**语法：**

var imgData = cxt.getImageData(x , y , width , height);

var data = imgData.data;

**说明：**

x、y表示所选图片区域的坐标，width、height表示所选图片区域的宽度和高度。

getImageData()方法返回的是一个canvasPixelArray对象，我们可以用一个变量（如imgData）来保存这个对象。canvasPixelArray对象有一个data属性，这个data属性是一个保存了这一张图片像素数据的数组，数组取值如[r1,g1,b1,a1,r2,g2,b2,a2,……]。也就是说，imgData.data这个数组中每4个数存储着1个像素的rgba颜色值，这4个数分别是红（r）、绿（g）、蓝（b）、透明度（a）。

其中，imgData.data表示一个保存像素数据的数组，imgData.data.length表示图片像素总量。

**二、putImageData()方法**

在canvas中，我们可以使用putImageData()方法输出一张图片的像素数据。简单来说，我们可以使用putImageData()方法来在Canvas中显示一张图片。

**语法：**

cxt.putImageData(image , x , y);

**说明：**

image表示重新绘制的图形，也就是使用getImageData()方法获取的canvasPixelArray对象。x、y表示重新绘制图形左上角的横坐标、纵坐标。

getImageData()和putImageData()都是配合使用的，一般都是先用getImageData()获取像素数据，然后利用一定的算法进行像素操作，最后再使用putImageData()方法输出像素数据（即在Canvas中显示一张图片）。

**三、createImageData()方法**

在Canvas中，除了getImageData()和putImageData()，我们还有一种像素操作的方法：createImageData()。

createImageData()用于在画布中创建一个区域，使得这个区域得以进行像素操作。如果没有创建像素操作区域，我们是没办法进行像素操作的。

**语法：**

cxt.createImageData(sw,sh); //第1种格式

cxt.createImageData(imageData); //第2种格式

**说明：**

createImageData()有2种语法格式：

（1）第1种格式：接收2个参数，sw和sh分别表示要创建区域的宽度和高度；

（2）第2种格式：接收一个像素对象，表示要“创建区域”的宽度和高度与“这个像素对象”的宽度和高度相等；

getImageData()和putImageData()配合使用是对“图像”进行像素操作的，而createImageData()和putImageData()配合使用是对“一个区域”进行像素操作的。

**四、像素操作**

在这一章中，我们使用getImageData()和putImageData()方法来对图片进行各种像素操作，包括：

**（1）反转效果；**

**（2）黑白效果；**

**（3）亮度效果；**

**（4）复古效果；**

**（5）红色蒙版；**

**（6）透明处理；**

很多图像处理效果的算法，从本质上来说其实跟大学线性代数中的矩阵有着密切的关系。考虑到各个学习层级的人，本书不对这些知识作详细介绍。如果你不经常搞图形学，忘记这些算法也是很正常。事实上我们也没必要记住，等到需要的时候再回来翻一翻就行了。

通过这一章的学习，我们也可以打造一个属于自己的图像处理小应用了，有兴趣的小伙伴可以自己动手设计一下。因为核心技术已经做了详细介绍，剩下的难点不多了。