# 认识Canvas元素

与video和audio类似．canvas元素完全不需要任何外部插件就能够运行。你只需要使用HTML并使用2D渲染上下文API（2D rendmng context API）编写一些JavaScript代码。

canvas元素的用法很简单，开始只需要添加下面的代码：

<canvas width=”500” height=”500”></canvas>

这段代码实际上并没有实现什么特殊的效果。它仅仅创建了一个新的空白canvas元素，还无法看到任何内容，因为还没有使用2D渲染上下文执行任何操作。

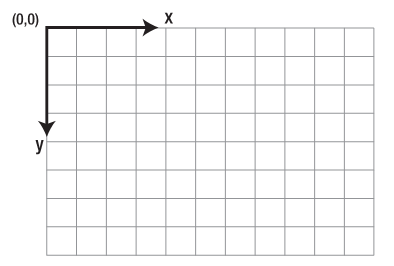
目前，在创建canvas元素时，需要特别注意的是width和height属性．这两个属性明确地定义了canvas元素的尺寸，从而相应地定义了2D渲染上下文的尺寸。如果不使用这些属性定义canvas元素的尺寸，那么2D渲染上下文会被设置为使用默认宽度和高度，分别是300和150像素。

## 2D渲染上下文

canvas元素并非Canvas中最强大的部分，真正的关键部分是2D渲染上下文，这是你真正绘制图形的地方。canvas元素的用途只是作为2D渲染上下文的包装器，它包含绘图和图形操作所需要的全部方法和丰富功能。理解这一点是很重要的，强调一下：绘图是在2D渲染上下文中进行的，而不是在canvas元素中进行。可以通过canvas元素访问和显示2D渲染上下文。

### 1．坐标系统

2D渲染上下文是一种基于屏幕的标准绘图平台。与其他的2D平台类似，它采用平面的笛卡儿坐标系统，左上角为原点（0,0）。向右移动时，x坐标值会增加；向下移动时，y坐标值会增加。如果你想把图形绘制到正确的位置上，一定要理解这个坐标系统。



**图1 2D渲染上下文的笛卡儿坐标系统**

坐标系统的1个单位通常相当于屏幕的1个像素，所以位置（24，30）是向右24像素和向下30像素的位置。有时候坐标系统的1个单位相当于2个像素（例如，在一些高分辨率显示器中），但是一般的经验法则是1个坐标单位等于1个屏幕像素。

### 2．访问2D渲染上下文

我们先创建只有一个空白canvas元素的简单HTML网页：

<! DOCTYPE html>

<html>

<head>

<title>Learning the basics of canvas</title>

<meta charset=”utf-8”>

<script type=”text/javascript” src=”http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js’></script>

<script type=”text/javascript”>

$(document).ready(function(){

})；

</script>

</head>

<body>

<canvas id=”myCanvas” width=”500” height=”500”>

</canvas>

</body>

</html>

如果现在运行，你不会看到任何内容，因此我们要访问2D渲染上下文，这样就可以开始绘图了。将下面的语句放到jQuery语句之中：

var canvas=$(“#mCanvas”);

var context=canvas.get(0).getContext(“2d”);

在这个例子中，我们将这个canvas元素赋值给一个变量，然后再通过调用getContext方法将得到的2D渲染上下文赋给另一个变量。必须强调一点，由于我们使用了jQuery，所以需要调用get方法才能访问DOM中的canvas元素，然后才能够访问Canvas的getContext方法。

有了包含2D渲染上下文的变量之后，就可以开始绘制图形了。在上下文变量声明语句之后添加下面这行代码：

context . fillRect(40,40,100,100);

刷新页面，页面上出现一个黑色的矩形。这个矩形是黑色的，因为Canvas所绘制元素的默认颜色是黑色。