HTML DOM CanvasRenderingContext2D 对象

**CanvasRenderingContext2D 对象**

CanvasRenderingContext2D 对象提供了一组用来在画布上绘制的图形函数。可用的函数非常丰富，它们可以分为以下几类：

**绘制矩形**

可以使用 [strokeRect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_strokerect.asp) 和 [fillRect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_fillrect.asp) 来绘制矩形的边框和填充矩形。此外，可以使用 [clearRect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_clearrect.asp) 来清除矩形所定义的区域。

strokeRect(左上角点X，左上角点Y，宽，高);

fillRect() 参数和strokeRect()相同

**绘制图像**

在 Canvas API 中，图像通过表示 [HTML <img> 元素](http://www.w3school.com.cn/tags/tag_img.asp)的 [Image 对象](http://www.w3school.com.cn/jsref/dom_obj_image.asp)来指定，或者通过使用 Image() 构造函数所创建的屏幕外图像来指定。一个 [Canvas 对象](http://www.w3school.com.cn/jsref/dom_obj_canvas.asp)也可以用作图像来源。

可以使用 [drawImage() 方法](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_drawimage.asp)在一个画布上绘制图像；而更为常见的形式是，允许源图像的任意矩形区域缩放或绘制到画布上。

**创建和渲染路径**

画布的一项强大功能是，它能够从基本的绘图操作来构建图形，然后，绘制这些图形的框架（勾勒它们）或者给这些图形的内容涂色（填充它们）。累计起来的操作统一叫做**当前路径**。一个画布只保持一条当前路径。

为了构建一个多条线段所构成的一个连续的封闭形状，绘制操作之间需要一个连接点。为此，画布保存了一个**当前位置**。画布绘制操作显式地使用这个位置作为它们的起始点，并且更新它直到到达终点。可以把这看做是使用钢笔或纸来绘画：当您完成了一条具体的线段或曲线，当前位置就是完成这一操作之后钢笔所停留的点。

可以在当前路径中创建一系列互相不连续的形状，它们都使用同样的绘制参数一起渲染。要分隔开这些形状，请使用 [moveTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_moveto.asp)方法，这个方法把当前的位置移动到一个新的位置而不添加一条连接线段。当您这么做的时候，就创建了一条新的路径，这是用来表示一组连接起来的操作的画布术语。

一旦您所想要的路径形成了，可以使用 [stroke()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_stroke.asp) 绘制其边框，使用 [fill()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_fill.asp) 来绘制其内容；或者两件事都做。

可用的图形操作有：用来绘制直线的 [lineTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_lineto.asp)，用于绘制矩形的 [rect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_rect.asp)，用于绘制部分圆形的 [arc()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_arc.asp) 或 [arcTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_arcto.asp)，以及用于绘制曲线的 [bezierCurveTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_beziercurveto.asp) 或 [quadraticCurveTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_quadraticcurveto.asp)。

除了勾勒和填充，还可以使用当前路径来指定绘制时画布所使用的剪切区域。这个区域中的像素是显示的，区域之外的像素是不显示的。剪切区域是累加性的；调用 [clip()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_clip.asp) 可以将当前路径和当前绘制区域取交集，产生一个新的区域。不幸的是，没有直接的方法把当前的剪切区域设置为画布的范围；要做到这一点，必须保存和恢复画布的整个图形状态。

如果任何子路径中的线段没有形成一个闭合的图形，[fill()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_fill.asp) 和 [clip()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_clip.asp) 操作通过添加一条从子路径的起点到终点的、虚拟（勾勒的时候看不到）线段来闭合它。可选的是，也可以调用 [closePath()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_closepath.asp) 来显示地添加这条线段。

**颜色、渐变和模式**

在填充和勾勒路径时，可用 fillStyle 和 strokeStyle 属性来指定线段或者绘制区域如何绘制。CSS 样式颜色字符串，以及描述渐变和模式的 [CanvasGradient](http://www.w3school.com.cn/jsref/dom_obj_canvasgradient.asp) 或 [CanvasPattern](http://www.w3school.com.cn/jsref/dom_obj_createpattern.asp) 都是可以接受的。要创建一个渐变，请使用 [createLinearGradient()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createlineargradient.asp) 或 [createRadialGradient()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createradialgradient.asp)。要创建一个模式，请使用 [createPattern()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createpattern.asp)。

要用 CSS 表示法来指定不透明的颜色，就采用 "#RRGGBB" 形式的字符串，其中 RR、GG 和 BB 分别是指定颜色的红色、绿色和蓝色成分的十六进制，其值都在 00 和 FF 之间。例如，完全红色的值是 "#FF0000"。要指定部分透明的颜色，请使用一个 "rgba(R,G,B,A)" 形式的字符串。在这种形式中，R、G 和 B 将颜色的红色、绿色和蓝色成分指定为 0 到 255 之间的十进制整数，并且 A 把 alpha（不透明）成分指定为 0.0 （完全透明）和 1.0 （完全不透明）之间的一个浮点数值。例如，半透明的完全红色为 "rgba(255,0,0,0.5)"。

createLinearGradient(x1, y1, x2, y2): 参数表示开始点到结束点

createRadialGradient(x1, y1, r1, x2, y2, r2): 参数表示开始圆心及半径，后结束圆心及半径

**线条宽度、线帽和线条连接**

画布为调整各种线条显示提供了几个选项。可以使用 lineWidth 属性来指定线条的宽度，用 [lineCap 属性](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_linecap.asp)来指定的端点如何绘制，并且用 [lineJoin 属性](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_linejoin.asp)来指定线条如何连接。

**坐标空间和转换**

默认情况下，一个画布的坐标空间使用画布的左上角 (0,0) 作为原点，x 值向右增加，y 值向下增加。这个坐标空间中的一个单位通常转换为像素。

然后，可以转换坐标空间，产生你在绘图操作中所指定的用来移动、缩放或旋转的任何坐标或范围。这通过 [translate()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_translate.asp)、[scale()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_scale.asp) 和 [rotate() 方法](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_rotate.asp)来实现，它们会对画布的变换矩阵产生影响。由于坐标空间可以像这样转换，您传递给 [lineTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_lineto.asp) 这样方法的坐标可能无法用像素来度量。因此，Canvas API 使用浮点数而不是整数。

变换按照它们被指定的顺序相反的顺序来处理。例如，调用 scale() 之后，紧接着调用 translate()，这会首先变换坐标系统，然后再缩放。

**组合**

通常，图形是一个绘制于另一个的上面，新的图形使得在它之前绘制在其下方的图形变得模糊。这是一个画布中的默认行为。然而，您可以通过为 [globalCompositeOperation 属性](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_globalcompositeoperation.asp)指定不同的值来执行很多有趣的操作，范围包括从 XOR 操作到增量或减暗图形区域。

**阴影**

Canvas API 包含了可以自动为您所绘制的任何图形添加下拉阴影的属性。然而，在编写本参考页时，Safari 是唯一实现了这一 API 的浏览器。阴影的颜色可用 shadowColor 属性来指定，并且可以通过 shadowOffsetX 和 shadowOffsetY 属性来改变。另外，应用到阴影边缘的羽化量也可以使用 shadowBlur 属性来设置。

**保存图形状态**

[save()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_save.asp) 和 [restore()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_restore.asp) 方法允许你保存和恢复一个 CanvasRenderingContext2D 对象的状态。save() 把当前状态推入到栈中，而 restore() 从栈的顶端弹出最近保存的状态，并且根据这些存储的值来设置当前绘图状态。

CanvasRenderingContext2D 对象的所有属性（除了画布的属性是一个常量）都是保存的状态的一部分。变换矩阵和剪切区域也是这个状态的一部分，但是当前路径和当前点并不是。

**CanvasRenderingContext2D 对象的属性**

**canvas 属性**

这个环境可以绘制于其上的 Canvas 元素。

**fillStyle 属性**

用来填充路径的当前的颜色、模式或渐变。这个属性可以设置为一个字符串或者一个 [CanvasGradient 对象](http://www.w3school.com.cn/jsref/dom_obj_canvasgradient.asp) 或 [CanvasPattern 对象](http://www.w3school.com.cn/jsref/dom_obj_createpattern.asp)。当设置为一个字符串时，它被解析为一个 CSS 颜色值并且用来进行实心填充。当设置为一个 CanvasGradient 或 CanvasPattern 对象，通过使用指定的渐变或模式来完成填充。

**globalAlpha 属性**

指定在画布上绘制的内容的不透明度。这个值的范围在 0.0（完全透明）和 1.0（完全不透明）之间。默认值为 1.0。

**globalCompositeOperation 属性**

指定颜色如何与画布上已有的颜色组合（合成）。如需更多细节，请参阅 [globalCompositeOperation 属性参考页](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_globalcompositeoperation.asp)。

**lineCap 属性**

指定线条的末端如何绘制。合法的值是 "butt"、"round" 和 "square"。默认值是 "butt"。如需更多细节，请参阅 [lineCap 属性参考页](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_linecap.asp)。

**lineJoin 属性**

指定两条线条如何连接。合法的值是 "round"、"bevel" 和 "miter"。默认值是 "miter"。如需更多细节，请参阅 [lineJoin 属性参考页](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_linejoin.asp)。

**lineWidth 属性**

指定了画笔（绘制线条）操作的线条宽度。默认值是 1.0，并且这个属性必须大于 0.0。较宽的线条在路径上居中，每边有线条宽的一半。

**miterLimit 属性**

当 lineJoin 属性为 "miter" 的时候，这个属性指定了斜连接长度和线条宽度的最大比率。如需更多细节，请参阅 [miterLimit 属性参考页](http://www.w3school.com.cn/jsref/prop_canvasrenderingcontext2d_miterlimit.asp)。

**shadowBlur 属性**

指定羽化阴影的程度。默认值是 0。阴影效果得到 safari 的支持，但是并没有得到 FireFox 1.5 或 Opera 9 的支持。

shadowOffsetX: 正数表示右边阴影，负数表示左边阴影

shadowOffsetY: 正数表示下边阴影，负数表示上边阴影

shadowBlur: 阴影的长度，一般小于shadowOffsetX 和shadowOffsetY

shadowColor: 阴影的颜色

**shadowColor 属性**

把阴影的颜色指定为一个 CSS 字符串或 Web 样式字符串，并且可以包含一个 alpha 部分来表示透明度。默认值是 black。阴影效果得到 Safari 的支持，但是并没有得到 FireFox 1.5 或 Opera 9 的支持。

**shadowOffsetX, shadowOffsetY 属性**

指定阴影的水平偏移和垂直偏移。较大的值使得阴影化的对象似乎漂浮在背景的较高位置上。默认值是 0。阴影效果得到 Safari 的支持，但是并没有得到 FireFox 1.5 或 Opera 9 的支持。

**strokeStyle 属性**

指定了用于画笔（绘制）路径的颜色、模式和渐变。这个属性可能是一个字符串，或者一个 [CanvasGradient 对象](http://www.w3school.com.cn/jsref/dom_obj_canvasgradient.asp) 或 [CanvasPattern 对象](http://www.w3school.com.cn/jsref/dom_obj_createpattern.asp)。如果是一个字符串，它被解析为一个 CSS 颜色值，并且画笔用所得的实色来绘制。如果这个属性的值是一个 CanvasGradient 对象或 CanvasPattern 对象，画笔使用这个渐变或模式来实现。

**CanvasRenderingContext2D 对象的方法**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [arc()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_arc.asp) | 用一个中心点和半径，为一个画布的当前子路径添加一条弧线。  （圆心X，圆心Y，半径，起始圆弧位置，终点圆弧位置，0顺时针或1逆时针） |
| [arcTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_arcto.asp) | 使用目标点和一个半径，为当前的子路径添加一条弧线。  （x1,y1,x2,y2,r） |
| [beginPath()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_beginpath.asp) | 开始一个画布中的一条新路径（或者子路径的一个集合）。 |
| [bezierCurveTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_beziercurveto.asp)  [quadraticCurveTo()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_quadraticcurveto.asp) | 添加一个三次贝塞尔曲线（两个控制点坐标，结束点）  添加一个二次贝塞尔曲线 （一个控制点坐标，结束点） | 为当前路径添加一条贝塞尔曲线。 |
| [clearRect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_clearrect.asp) | 在一个画布的一个矩形区域中清除掉像素。  （左上角X，左上角Y，宽，高） |
| [clip()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_clip.asp) | 使用当前路径作为连续绘制操作的剪切区域。 |
| [closePath()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_closepath.asp) | 如果当前子路径是打开的，就关闭它。 |
| [createLinearGradient()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createlineargradient.asp) | 返回代表线性颜色渐变的一个 CanvasGradient 对象。 |
| [createPattern()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createpattern.asp) | 返回代表贴图图像的一个 CanvasPattern 对象。 |
| [createRadialGradient()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_createradialgradient.asp) | 返回代表放射颜色渐变的一个 CanvasGradient 对象。 |
| [drawImage(img, )](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_drawimage.asp) | 绘制一幅图像。 |
| [fill()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_fill.asp) | 使用指定颜色、渐变或模式来绘制或填充当前路径的内部。 |
| [fillRect(x, y, width, height)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_fillrect.asp) | 绘制或填充一个矩形。 |
| [lineTo(x, y)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_lineto.asp) | 为当前的子路径添加一条直线线段。  （移动点X，移动点Y） |
| [moveTo(x, y)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_moveto.asp) | 设置当前位置并开始一条新的子路径。  （移动点X，移动点Y） |
|  |  |
| [rect()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_rect.asp) | 为当前路径添加一条矩形子路径。  （左上角X，左上角Y，宽，高） |
| [restore()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_restore.asp) | 为画布重置为最近保存的图像状态。 |
| [rotate(角度值)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_rotate.asp) | 旋转画布。 |
| [save()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_save.asp) | 保存 CanvasRenderingContext2D 对象的属性、剪切区域和变换矩阵。 |
| [scale(x倍数, y倍数)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_scale.asp) | 标注画布的用户坐标系统。 |
| [stroke()](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_stroke.asp) | 沿着当前路径绘制或画一条直线。 |
| [strokeRect(x, y, width, height)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_strokerect.asp) | 绘制（但不填充）一个矩形。 |
| [translate(x, y)](http://www.w3school.com.cn/jsref/met_canvasrenderingcontext2d_translate.asp) | 转换画布的用户坐标系统。 |
| fillText(string, x, y, maxWidth) | 添加文字 |
| StrokeText() |  |

**drawImg():**

canvas.getComtext(“2d”).src 可以设置图片：

context.drawImage(img, x, y) 在画布上的定位

context.drawImage(img, x, y, width, height) 在画布上定位，宽高

context.drawImage(img, sx, sy, swidth, sheight, x, y, width, height) 剪切位置以及定宽高，在画布上的位置及宽高

用图片填充背景：

bgImg = new Img();

bgImg.src = “”;

fillStyle = myCanvas.createPattern(bgImg, “repeat or repeat-x or repeat-y or no-repeat”)

context.setTranform(a,b,c,d,e,f);分别表示水平旋转绘图，水平倾斜绘图，垂直倾斜绘图，垂直缩放绘图，水平缩放绘图，垂直移动绘图，初始化时可以用content.setTransform(1, 0, 0, 1, 0, 0)

重置画板的方法：

1：重新设置画布的宽高

2：用fillRect(0, 0, content.width, content.height)直接清空画布