

# 画布Canvas

## 基本用法

[getContext\(contextID\)](#) 返回一个用于在画布上绘图的环境。

要在画布（canvas）上绘图，需要取得绘图上下文对象的引用，即调用 `getContext()` 并传入上下文名字。

参数 `contextID` 指定了您想要在画布上绘制的类型。当前唯一的合法值是 “2d”，它指定了二维绘图。

[toDataURL\(\)](#) 导出在<canvas>上绘制的图像。

## 2D上下文

[fillStyle](#) 填充

[strokeStyle](#) 描边

属性的值可以是：

`color` 颜色值。格式：颜色名、十六进制码、rgb、rgba(红色值，绿色值，蓝色值，Alpha透明度)、hsl或hsla。

`gradient` 用于填充绘图的渐变对象（线性或放射性）。

`pattern` 模式对象。

默认值是 #000000。

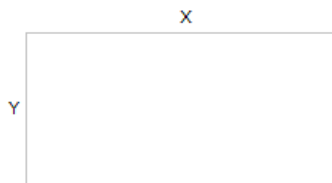
## 绘制矩形

[fillRect\(x, y, width, height\)](#) 从(x, y)开始绘制“被填充”的矩形。绘制前先将fillStyle设置为指定颜色。

[strokeRect\(x, y, width, height\)](#) 从(x, y)开始绘制“被描边”的矩形。绘制前将strokeStyle设置为指定颜色。

[clearRect\(x, y, width, height\)](#) 在给定的矩形内清除指定的像素。

参数单位：像素



## 绘制路径

[beginPath\(\)](#) 开始一条路径，或重置当前的路径。

要绘制路径，首先必须调用beginPath()。

[arc\(x, y, r, startAngle, endAngle, counterclockwise\)](#) 以(x, y)为圆心绘制弧线（用于创建圆或部分圆）。

参数：

`sAngle` 起始角，以弧度计。（弧的圆形的三点钟位置是 0 度）。

`eAngle` 结束角，以弧度计。

`counterclockwise` 可选，规定应该逆时针还是顺时针绘图。False = 顺时针，true = 逆时针。

[arcTo\(x1, y1, x2, y2, r\)](#) 从(x1, y1)开始绘制弧线，到(x2, y2)为止。

[bezierCurveTo\(c1x, c1y, c2x, c2y, x, y\)](#) 从上一点开始绘制三次方贝塞尔曲线，到(x, y)为止，并以(c1x, c1y)和(c2x, c2y)为控制点。

[lineTo\(x, y\)](#) 从上一点开始绘制一条直线，的，到(x, y)为止。

[moveTo\(x, y\)](#) 把路径移动到(x, y)，不画线。

[quadraticCurveTo\(\)](#) 从上一点开始绘制二次方贝塞尔曲线，到(x, y)为止，并以(cx, cy)为控制点。

[rect\(x, y, width, height\)](#) 从(x, y)开始绘制矩形路径。

[fill\(\)](#) 填充当前绘制路径

[stroke\(\)](#) 给当前路径描边

[clip\(\)](#) 在路径上创建一个剪切区域

[isPointInPath\(x, y\)](#) 如果点(x, y)位于当前路径中，则返回 true，否则返回 false

## 绘制文本

[font](#) 设置或返回文本的样式、大小、字体属性。font 属性使用的语法与 [CSS font 属性](#)相同。例如 “bold 10px Arial”

[textAlign](#) 设置或返回文本的对齐方式。可能的值：start(默认)、end、center、left(不建议)、right(不建议)

[textBaseline](#) 设置或返回文本的基线。可能的值：alphabetic(默认)、top、hanging、middle、ideographicbottom

[fillText\(\(text, x, y, maxWidth\)](#) 在画布上绘制填色的文本。

[strokeText\(\(text, x, y, maxWidth\)](#) 在画布上绘制描边的文本。

文本的默认颜色是黑色。

参数：

maxWidth 可选。允许的最大文本宽度，以像素计。

[measureText\(text\)](#) 返回返回一个对象，该对象只有一个width属性，包含文本的宽度。使用方

法：`context.measureText(text).width;`

## 变换

[rotate\(angle\)](#) 将图像围绕原点旋转`angle`角度。

[scale\(scalewidth, scaleheight\)](#) 对图像进行缩放。

参数：

scalewidth 缩放当前绘图的宽度（1.0=100%，0.5=50%，2=200%，依次类推）

scaleheight 缩放当前绘图的高度（1.0=100%，0.5=50%，2=200%，etc.）

默认值：1.0

[translate\(x, y\)](#) 将坐标原点移动到(x, y)。执行后，坐标(0, 0)会变成之前由(x, y)表示的点。

[transform\(a, b, c, d, e, f\)](#)

画布上的每个对象都拥有一个当前的变换矩阵。

transform() 方法替换当前的变换矩阵。它以下面描述的矩阵来操作当前的变换矩阵：

```
a c e  
b d f  
0 0 1
```

换句话说，transform() 允许您缩放、旋转、移动并倾斜当前的环境。

参数：

b 水平倾斜绘图

c 垂直倾斜绘图

d 垂直缩放绘图

e 水平移动绘图

f 垂直移动绘图

[setTransform\(a, b, c, d, e, f\)](#) 将当前转换重置为单位矩阵。然后运行 transform()

## 图像绘制

[drawImage\(\)](#) 向画布上绘制图像

- [drawImage\(img, x, y\)](#) 在画布上绘制图像
- [drawImage\(img, x, y, width, height\)](#) 在画布上绘制图像，并规定图像的宽度和高度
- [drawImage\(img, sx, sy, swidth, sheight, x, y, width, height\)](#) 剪切图像，并在画布上绘制被剪切的部分

## 阴影

[shadowColor](#) 设置或返回阴影颜色。默认为黑色。

[shadowBlur](#) 设置或返回阴影的模糊级别。默认为0，即不模糊。

[shadowOffsetX](#) 设置或返回阴影距形状的水平距离。默认为0。

[shadowOffsetY](#) 设置或返回阴影距形状的垂直距离。默认为0。

## 渐变

[createLinearGradient\(x0, y0, x1, y1\)](#) 创建从(x0, y0)开始(x1, y1)结束的线性渐变

[addColorStop\(stop, color\)](#) 规定渐变对象中的颜色和停止位置

参数:

stop 介于 0.0 与 1.0 之间的值, 表示渐变中开始与结束之间的位置。

color 在结束位置显示的 CSS 颜色值

## 模式

[createPattern\(image, repeat\)](#) 在指定的方向上重复指定的元素

image 规定要使用的图片、画布或视频元素。

repeat 表示如何重复的字符串。可能的值: repeat(默认)、repeat-x、repeat-y、no-repeat

## 像素操作

[width](#) 返回 ImageData 对象的宽度

[height](#) 返回 ImageData 对象的高度

[data](#) 返回一个对象, 其包含指定的 ImageData 对象的图像数据

[getImageData\(x, y, width, height\)](#) 返回 ImageData 对象, 该对象为画布上指定的矩形复制像素数据

[putImageData\(imgData, x, y, dirtyX, dirtyY, dirtyWidth, dirtyHeight\)](#) 把图像数据 (从指定的 ImageData 对象) 放回画布上

## 合成

[globalAlpha](#) 设置或返回绘图的当前 alpha 或透明值

[globalCompositeOperation](#) 设置或返回新图像如何绘制到已有的图像上

## 保存与恢复

[save\(\)](#) 当前所有设置保存进一个栈结构。

[restore\(\)](#) 在保存设置的栈结构中向前返回一级, 恢复之前的状态。