**HTML5 <canvas> 参考手册**

<http://www.runoob.com/tags/ref-canvas.html>

**描述**

HTML5 <canvas> 标签用于绘制图像（通过脚本，通常是 JavaScript）。

不过，<canvas> 元素本身并没有绘制能力（它仅仅是图形的容器） - 您必须使用脚本来完成实际的绘图任务。

getContext() 方法可返回一个对象，该对象提供了用于在画布上绘图的方法和属性。

本手册提供完整的 getContext("2d") 对象的属性和方法，可用于在画布上绘制文本、线条、矩形、圆形等等。

**浏览器支持**

Internet Explorer Firefox Opera Google Chrome Safari

Internet Explorer 9、Firefox、Opera、Chrome 和 Safari 支持 <canvas> 标签的属性及方法。

**注意:**Internet Explorer 8 及更早的IE版本不支持 <canvas> 元素。

**颜色、样式和阴影**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [fillStyle](http://www.runoob.com/tags/canvas-fillstyle.html) | 设置或返回用于填充绘画的颜色、渐变或模式。 |
| [strokeStyle](http://www.runoob.com/tags/canvas-strokestyle.html) | 设置或返回用于笔触的颜色、渐变或模式。 |
| [shadowColor](http://www.runoob.com/tags/canvas-shadowcolor.html) | 设置或返回用于阴影的颜色。 |
| [shadowBlur](http://www.runoob.com/tags/canvas-shadowblur.html) | 设置或返回用于阴影的模糊级别。 |
| [shadowOffsetX](http://www.runoob.com/tags/canvas-shadowoffsetx.html) | 设置或返回阴影与形状的水平距离。 |
| [shadowOffsetY](http://www.runoob.com/tags/canvas-shadowoffsety.html) | 设置或返回阴影与形状的垂直距离。 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [createLinearGradient()](http://www.runoob.com/tags/canvas-createlineargradient.html) | 创建线性渐变（用在画布内容上）。 |
| [createPattern()](http://www.runoob.com/tags/canvas-createpattern.html) | 在指定的方向上重复指定的元素。 |
| [createRadialGradient()](http://www.runoob.com/tags/canvas-createradialgradient.html) | 创建放射状/环形的渐变（用在画布内容上）。 |
| [addColorStop()](http://www.runoob.com/tags/canvas-addcolorstop.html) | 规定渐变对象中的颜色和停止位置。 |

**线条样式**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [lineCap](http://www.runoob.com/tags/canvas-linecap.html) | 设置或返回线条的结束端点样式。 |
| [lineJoin](http://www.runoob.com/tags/canvas-linejoin.html) | 设置或返回两条线相交时，所创建的拐角类型。 |
| [lineWidth](http://www.runoob.com/tags/canvas-linewidth.html) | 设置或返回当前的线条宽度。 |
| [miterLimit](http://www.runoob.com/tags/canvas-miterlimit.html) | 设置或返回最大斜接长度。 |

**矩形**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [rect()](http://www.runoob.com/tags/canvas-rect.html) | 创建矩形。 |
| [fillRect()](http://www.runoob.com/tags/canvas-fillrect.html) | 绘制"被填充"的矩形。 |
| [strokeRect()](http://www.runoob.com/tags/canvas-strokerect.html) | 绘制矩形（无填充）。 |
| [clearRect()](http://www.runoob.com/tags/canvas-clearrect.html) | 在给定的矩形内清除指定的像素。 |

**路径**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [fill()](http://www.runoob.com/tags/canvas-fill.html) | 填充当前绘图（路径）。 |
| [stroke()](http://www.runoob.com/tags/canvas-stroke.html) | 绘制已定义的路径。 |
| [beginPath()](http://www.runoob.com/tags/canvas-beginpath.html) | 起始一条路径，或重置当前路径。 |
| [moveTo()](http://www.runoob.com/tags/canvas-moveto.html) | 把路径移动到画布中的指定点，不创建线条。 |
| [closePath()](http://www.runoob.com/tags/canvas-closepath.html) | 创建从当前点回到起始点的路径。 |
| [lineTo()](http://www.runoob.com/tags/canvas-lineto.html) | 添加一个新点，然后在画布中创建从该点到最后指定点的线条。 |
| [clip()](http://www.runoob.com/tags/canvas-clip.html) | 从原始画布剪切任意形状和尺寸的区域。 |
| [quadraticCurveTo()](http://www.runoob.com/tags/canvas-quadraticcurveto.html) | 创建二次贝塞尔曲线。 |
| [bezierCurveTo()](http://www.runoob.com/tags/canvas-beziercurveto.html) | 创建三次贝塞尔曲线。 |
| [arc()](http://www.runoob.com/tags/canvas-arc.html) | 创建弧/曲线（用于创建圆形或部分圆）。 |
| [arcTo()](http://www.runoob.com/tags/canvas-arcto.html) | 创建两切线之间的弧/曲线。 |
| [isPointInPath()](http://www.runoob.com/tags/canvas-ispointinpath.html) | 如果指定的点位于当前路径中，则返回 true，否则返回 false。 |

**转换**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [scale()](http://www.runoob.com/tags/canvas-scale.html) | 缩放当前绘图至更大或更小。 |
| [rotate()](http://www.runoob.com/tags/canvas-rotate.html) | 旋转当前绘图。 |
| [translate()](http://www.runoob.com/tags/canvas-translate.html) | 重新映射画布上的 (0,0) 位置。 |
| [transform()](http://www.runoob.com/tags/canvas-transform.html) | 替换绘图的当前转换矩阵。 |
| [setTransform()](http://www.runoob.com/tags/canvas-settransform.html) | 将当前转换重置为单位矩阵。然后运行 transform()。 |

**文本**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [font](http://www.runoob.com/tags/canvas-font.html) | 设置或返回文本内容的当前字体属性。 |
| [textAlign](http://www.runoob.com/tags/canvas-textalign.html) | 设置或返回文本内容的当前对齐方式。 |
| [textBaseline](http://www.runoob.com/tags/canvas-textbaseline.html) | 设置或返回在绘制文本时使用的当前文本基线。 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [fillText()](http://www.runoob.com/tags/canvas-filltext.html) | 在画布上绘制"被填充的"文本。 |
| [strokeText()](http://www.runoob.com/tags/canvas-stroketext.html) | 在画布上绘制文本（无填充）。 |
| [measureText()](http://www.runoob.com/tags/canvas-measuretext.html) | 返回包含指定文本宽度的对象。 |

**图像绘制**

|  |  |
| --- | --- |
| **方法** | **描述** |
| [drawImage()](http://www.runoob.com/tags/canvas-drawimage.html) | 向画布上绘制图像、画布或视频。 |

**像素操作**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [width](http://www.runoob.com/tags/canvas-imagedata-width.html) | 返回 ImageData 对象的宽度。 |
| [height](http://www.runoob.com/tags/canvas-imagedata-height.html) | 返回 ImageData 对象的高度。 |
| [data](http://www.runoob.com/tags/canvas-imagedata-data.html) | 返回一个对象，其包含指定的 ImageData 对象的图像数据。 |

|  |  |
| --- | --- |
| **方法** | **描述** |
| [createImageData()](http://www.runoob.com/tags/canvas-createimagedata.html) | 创建新的、空白的 ImageData 对象。 |
| [getImageData()](http://www.runoob.com/tags/canvas-getimagedata.html) | 返回 ImageData 对象，该对象为画布上指定的矩形复制像素数据。 |
| [putImageData()](http://www.runoob.com/tags/canvas-putimagedata.html) | 把图像数据（从指定的 ImageData 对象）放回画布上。 |

**合成**

|  |  |
| --- | --- |
| **属性** | **描述** |
| [globalAlpha](http://www.runoob.com/tags/canvas-globalalpha.html) | 设置或返回绘图的当前 alpha 或透明值。 |
| [globalCompositeOperation](http://www.runoob.com/tags/canvas-globalcompositeoperation.html) | 设置或返回新图像如何绘制到已有的图像上。 |

**其他**

|  |  |
| --- | --- |
| **方法** | **描述** |
| save() | 保存当前环境的状态。 |
| restore() | 返回之前保存过的路径状态和属性。 |
| createEvent() |  |
| getContext() |  |
| toDataURL() |  |

← [HTML 事件](http://www.runoob.com/tags/ref-eventattributes.html)

[HTML 音频/视频](http://www.runoob.com/tags/ref-av-dom.html) →

# HTML5 Canvas

# <http://www.runoob.com/html/html5-canvas.html>

<canvas> 标签定义图形，比如图表和其他图像，您必须使用脚本来绘制图形。

在画布上（Canvas）画一个红色矩形，渐变矩形，彩色矩形，和一些彩色的文字。

## 什么是 Canvas?

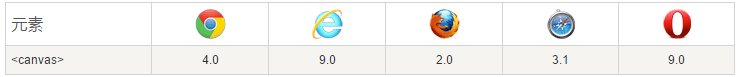
HTML5 <canvas> 元素用于图形的绘制，通过脚本 (通常是JavaScript)来完成.

<canvas> 标签只是图形容器，您必须使用脚本来绘制图形。

你可以通过多种方法使用Canva绘制路径,盒、圆、字符以及添加图像。

## 浏览器支持

表格中的数字表示支持 <canvas> 元素的第一个浏览器版本号。



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 元素 |  |  |  |  |  |
| <canvas> | 4.0 | 9.0 | 2.0 | 3.1 | 9.0 |

## 创建一个画布（Canvas）

一个画布在网页中是一个矩形框，通过 <canvas> 元素来绘制.

**注意:** 默认情况下 <canvas> 元素没有边框和内容。

<canvas>简单实例如下:

<canvas id="myCanvas" width="200" height="100"></canvas>

**注意:** 标签通常需要指定一个id属性 (脚本中经常引用), width 和 height 属性定义的画布的大小.

**提示:**你可以在HTML页面中使用多个 <canvas> 元素.

使用 style 属性来添加边框:

## 实例

<canvas id="myCanvas" width="200" height="100"  
style="border:1px solid #000000;">  
</canvas>

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_empty)

## 使用 JavaScript 来绘制图像

canvas 元素本身是没有绘图能力的。所有的绘制工作必须在 JavaScript 内部完成：

## 实例

<script>  
var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.fillStyle="#FF0000";  
ctx.fillRect(0,0,150,75);  
</script>

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_first)

**实例解析:**

首先，找到 <canvas> 元素:

var c=document.getElementById("myCanvas");

然后，创建 context 对象：

var ctx=c.getContext("2d");

getContext("2d") 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

下面的两行代码绘制一个红色的矩形：

ctx.fillStyle="#FF0000";  
ctx.fillRect(0,0,150,75);

设置fillStyle属性可以是CSS颜色，渐变，或图案。fillStyle 默认设置是#000000（黑色）。

fillRect(*x,y,width,height*) 方法定义了矩形当前的填充方式。

## Canvas 坐标

canvas 是一个二维网格。

canvas 的左上角坐标为 (0,0)

上面的 fillRect 方法拥有参数 (0,0,150,75)。

意思是：在画布上绘制 150x75 的矩形，从左上角开始 (0,0)。

**坐标实例**

如下图所示，画布的 X 和 Y 坐标用于在画布上对绘画进行定位。鼠标移动的矩形框上，显示定位坐标。

X

Y

## Canvas - 路径

在Canvas上画线，我们将使用以下两种方法：

* moveTo(*x,y*) 定义线条开始坐标
* lineTo(*x,y*) 定义线条结束坐标

绘制线条我们必须使用到 "ink" 的方法，就像stroke().

## 实例

定义开始坐标(0,0), 和结束坐标 (200,100). 然后使用 stroke() 方法来绘制线条:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.moveTo(0,0);  
ctx.lineTo(200,100);  
ctx.stroke();

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_path)

在canvas中绘制圆形, 我们将使用以下方法:

* arc(x,y,r,start,stop)

实际上我们在绘制圆形时使用了 "ink" 的方法, 比如 stroke() 或者 fill().

## 实例

使用 arc() 方法 绘制一个圆:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95,50,40,0,2\*Math.PI);  
ctx.stroke();

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_path2)

## Canvas - 文本

使用 canvas 绘制文本，重要的属性和方法如下：

* font - 定义字体
* fillText(*text,x,y*) - 在 canvas 上绘制实心的文本
* strokeText(*text,x,y*) - 在 canvas 上绘制空心的文本

使用 fillText():

## 实例

使用 "Arial" 字体在画布上绘制一个高 30px 的文字（实心）：

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.font="30px Arial";  
ctx.fillText("Hello World",10,50);

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_text)

使用 strokeText():

## 实例

使用 "Arial" 字体在画布上绘制一个高 30px 的文字（空心）：

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
ctx.font="30px Arial";  
ctx.strokeText("Hello World",10,50);

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_text2)

## Canvas - 渐变

渐变可以填充在矩形, 圆形, 线条, 文本等等, 各种形状可以自己定义不同的颜色。

以下有两种不同的方式来设置Canvas渐变：

* createLinearGradient(*x,y,x1,y1*) - 创建线条渐变
* createRadialGradient(*x,y,r,x1,y1,r1*) - 创建一个径向/圆渐变

当我们使用渐变对象，必须使用两种或两种以上的停止颜色。

addColorStop()方法指定颜色停止，参数使用坐标来描述，可以是0至1.

使用渐变，设置fillStyle或strokeStyle的值为 渐变，然后绘制形状，如矩形，文本，或一条线。

使用 createLinearGradient():

## 实例

创建一个线性渐变。使用渐变填充矩形:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
// Create gradient  
var grd=ctx.createLinearGradient(0,0,200,0);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle=grd;  
ctx.fillRect(10,10,150,80);

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_grad)

使用 createRadialGradient():

## 实例

创建一个径向/圆渐变。使用渐变填充矩形：

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
// Create gradient  
var grd=ctx.createRadialGradient(75,50,5,90,60,100);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle=grd;  
ctx.fillRect(10,10,150,80);

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_grad2)

## Canvas - 图像

把一幅图像放置到画布上, 使用以下方法:

* drawImage(*image,x,y*)

## 使用图像:



## 实例

把一幅图像放置到画布上:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
var img=document.getElementById("scream");  
ctx.drawImage(img,10,10);

[**尝试一下 »**](http://www.runoob.com/try/try.php?filename=tryhtml5_canvas_tut_img)

## HTML Canvas 参考手册

标签的完整属性可以参考[Canvas 参考手册.](http://www.runoob.com/tags/ref-canvas.html)

## HTML <canvas> 标签

|  |  |
| --- | --- |
| **Tag** | **描述** |
| [<canvas>](http://www.runoob.com/tags/tag-canvas.html) | HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。 |

← [HTML5 新元素](http://www.runoob.com/html/html5-new-element.html)

[HTML5 内联 SVG](http://www.runoob.com/html/html5-svg.html)