

<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 canvas 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	

## 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 »

## 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

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

### 尝试一下 »

#### 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 »](#)

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 »](#)

---

## 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 »](#)

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 »](#)

---

## 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");

// 创建渐变
var grd=ctx.createLinearGradient(0,0,200,0);
```

```
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// 填充渐变
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
```

[尝试一下 »](#)

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");

// 创建渐变
var grd=ctx.createRadialGradient(75,50,5,90,60,100);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// 填充渐变
ctx.fillStyle=grd;
ctx.fillRect(10,10,150,80);
```

[尝试一下 »](#)

---

## 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:



The Scream

## 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 »](#)