

# canvas 绘图之线条及线条属性

## 一、创建 canvas

首先创建一个 canvas 元素，只需要在 html 文件中加入一句代码：

```
<canvas id="canvas">建议使用 Chrome 浏览器使用！</canvas>
```

可通过 canvas 的标签属性 width 和 height 设置 canvas 画布的大小：

```
<canvas id="canvas" width="800" height="600">建议使用 Chrome 浏览器使用！</canvas>
```

## 二、获取环境

通过 js 中获取到该 canvas 元素，然后设置它的宽高，并获取到上下文的环境：

```
var canvas = document.getElementById("canvas");//获取到 canvas 元素  
var context = canvas.getContext("2d");//获取上下文的环境
```

接下来我们的所有操作都是基于这个 context 的上下文环境。

## 三、画一条简单的直线

```
context.moveTo(100,100);  
context.lineTo(500,500);
```

- moveTo()方法表示一次绘制的起点坐标
- lineTo()表示基于上一个坐标点到这个坐标点之间的直线连接。

注意：canvas 是基于状态的绘制，而不是基于对象的绘制。所以，上面代码都是

状态的设置，还需要使用 stroke()方法进行绘制：

```
context.stroke();//绘制
```

除此之外，我们还可以设置线条的一些基本属性：

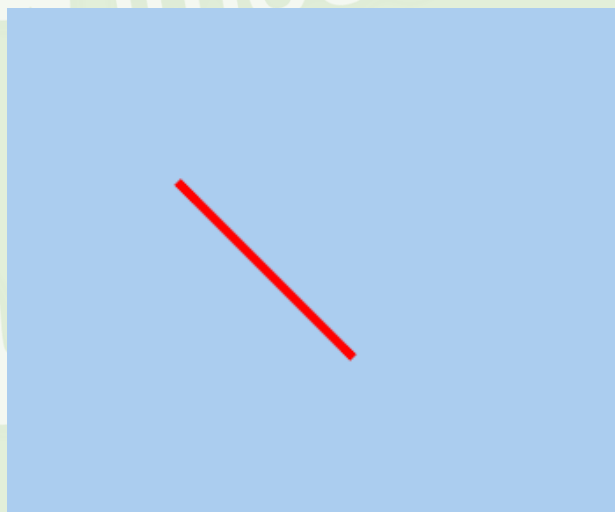
```
context.lineWidth = 8;//线条的宽度
```

```
context.strokeStyle = "#333";//线条的颜色
```

一个简单的绘制一条直线的完整例子

```
var canvas = document.getElementById("canvas");//获取到 canvas 元素  
var context = canvas.getContext("2d");//获取上下文的环境  
context.moveTo(100,100);  
context.lineTo(200,200);  
context.lineWidth = 5;//线条的宽度  
context.strokeStyle = "red";//线条的颜色  
context.stroke();//绘制
```

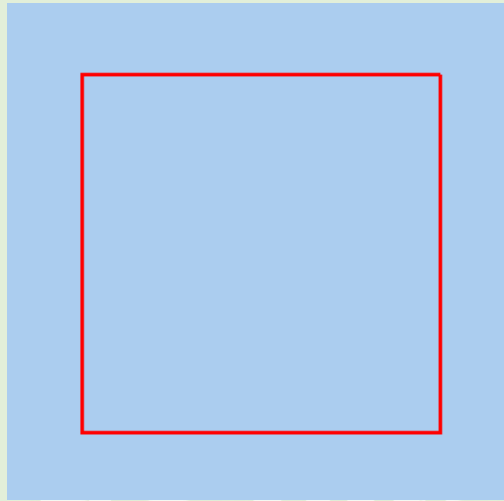
运行结果如下图：



#### 四、绘制一个连续折线

```
context.moveTo(100,100);  
context.lineTo(600,100);  
context.lineTo(600,600);  
context.lineTo(100,600);  
context.lineTo(100,100);
```

运行结果如下：



上面简单的例子我们是连续绘制的折线，也就是说可以一笔连续画完的折线。如果是多条间断的折线，那么我们就需要使用 `context.moveTo()` 来重新绘制一条折线的起点坐标：

```
context.moveTo(100,200);  
context.lineTo(300,400);  
context.lineTo(100,600);  
  
context.moveTo(300,200);  
context.lineTo(500,400);  
context.lineTo(300,600);  
  
context.moveTo(500,200);  
context.lineTo(700,400);  
context.lineTo(500,600);
```

运行结果如下：

