



canvas 图形样式



李伟

课堂目标

1. 掌握canvas 所提供的样式接口
2. 能够绘制丰富多彩的图形

知识点综述

1. 图形着色的区域
2. 图形着色的方式
3. 描边的样式
4. 投影

图形的着色区域有两种

- 描边区域：strokeStyle 代表了描边样式，描边区域的绘制方法是 stroke()、strokeRect() 或者strokeText()。
- 填充区域：fillStyle 代表了填充样式，填充区域的绘制方法是 fill()、fillRect() 或者fillText()。

图形的着色方式有3种

- 纯色
- 渐变
- 纹理

纯色

1. 书写方式：

- red
- #000000
- rgb(r,g,b)
- rgba(r,g,b,a)

2. 赋值方式

- `ctx.fillStyle= 'red'`
- `ctx.strokeStyle= 'rgb(r,g,b)'`

渐变

1. 建立渐变对象的方式：

- 线性渐变 `gradient=createLinearGradient(x1, y1, x2, y2)`
- 径向渐变 `gradient=createRadialGradient(x1, y1, r1, x2, y2, r2)`

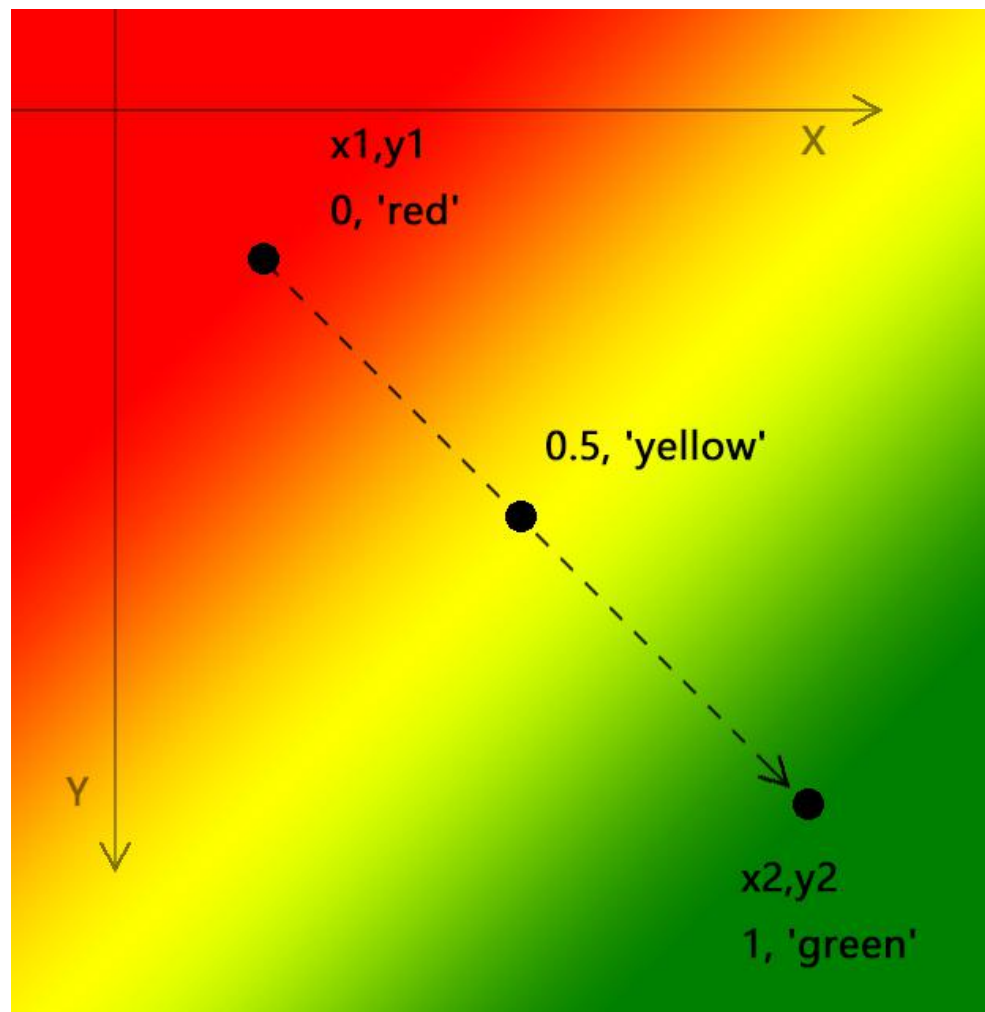
2. 定义渐变的颜色节点

- `gradient.addColorStop(position, color)`

3. 赋值方式

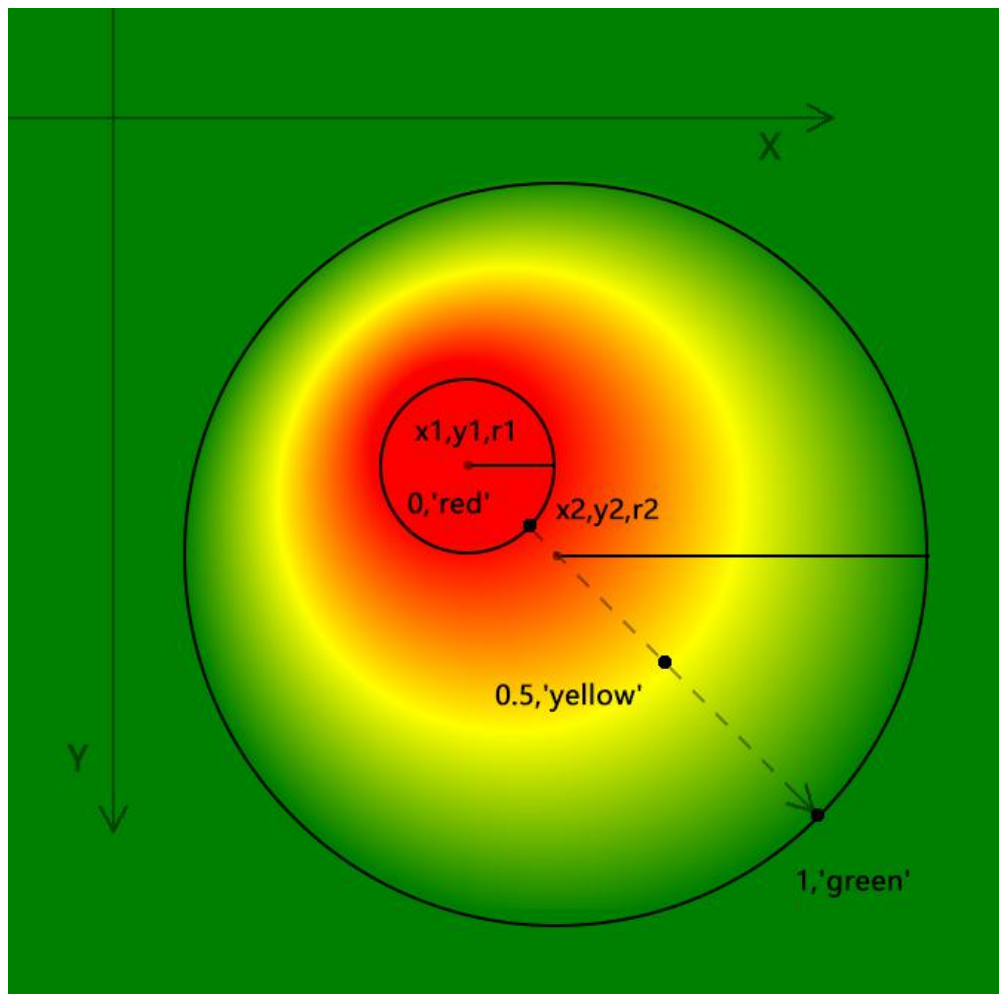
- `ctx.fillStyle= gradient`
- `ctx.strokeStyle= gradient`

线性渐变



```
const linerGradient=context.createLinearGradient(x1, y1, x2, y2);  
linerGradient.addColorStop(0, 'red');  
linerGradient.addColorStop(.5, 'yellow');  
linerGradient.addColorStop(1, 'green');
```


径向像渐变

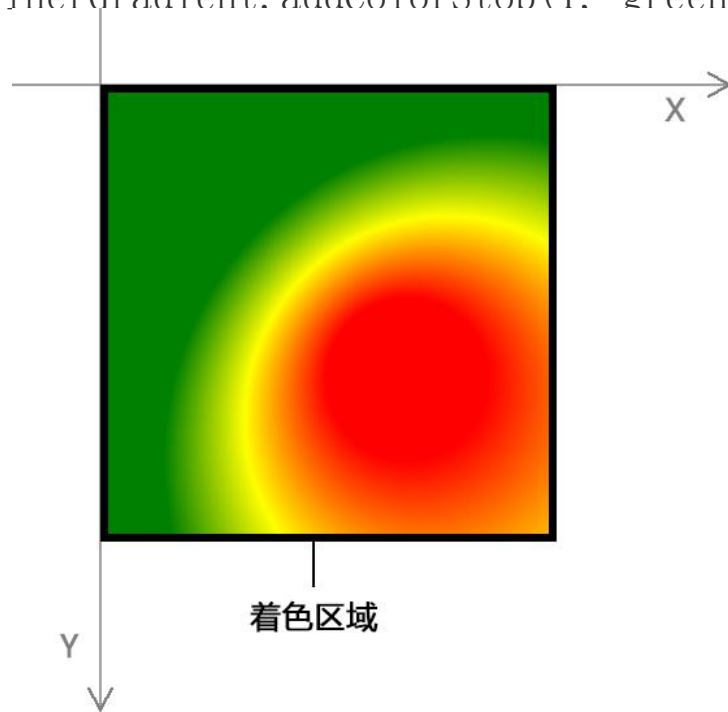


```
const linerGradient=ctx.createLinearGradient(x1, y1, r1, x2, y2, r2);
```

```
linerGradient.addColorStop(0, 'red');
```

```
linerGradient.addColorStop(.5, 'yellow');
```

```
linerGradient.addColorStop(1, 'green');
```



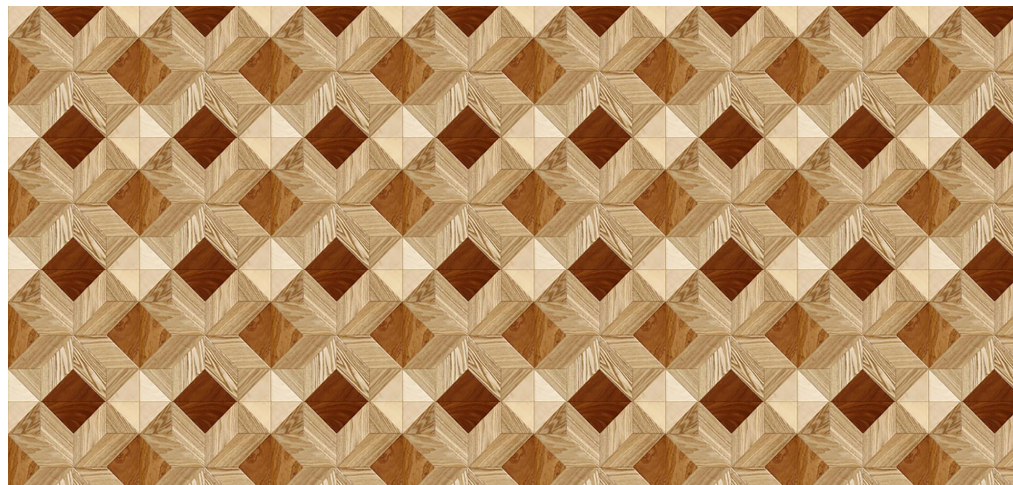
纹理

1. 建立纹理对象：

- `pattern=context.createPattern(image,"repeat|repeat-x|repeat-y|no-repeat");`

2. 为着色样式赋值

- `ctx.fillStyle= pattern`
- `ctx.strokeStyle= pattern`

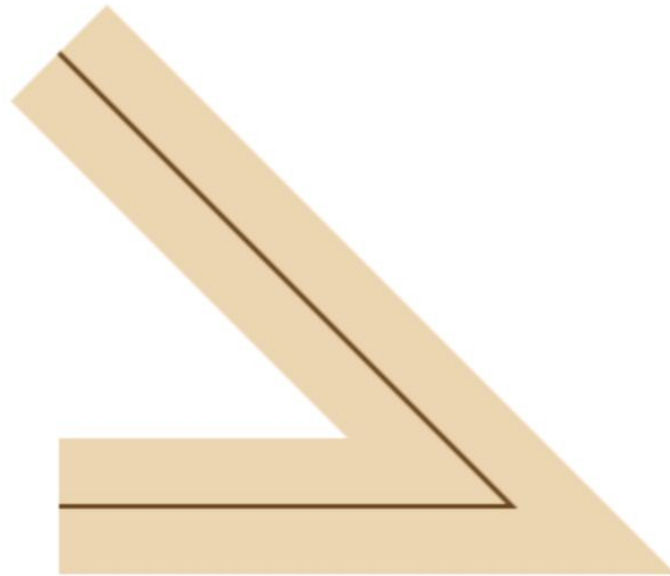


影响描边样式的因素

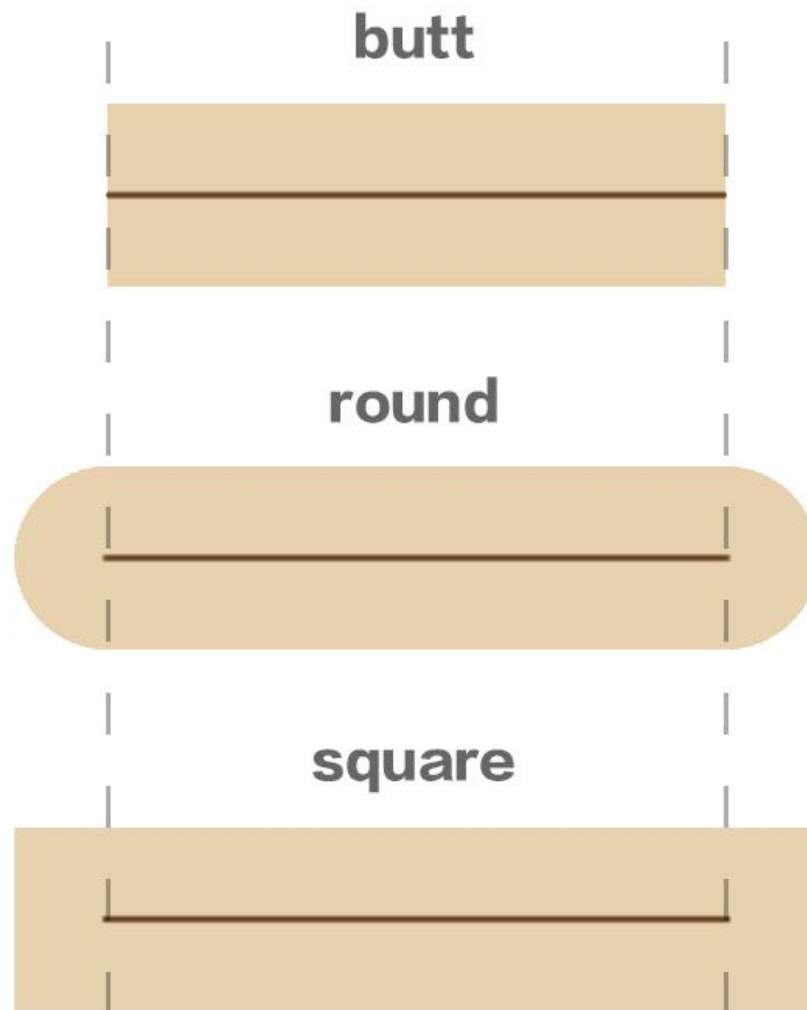
- strokeStyle : 描边的颜色
- lineWidth : 描边宽
- lineCap : 描边端点样式
- lineJoin : 描边拐角类型
- miterLimit : 拐角最大厚度 (只适用于lineJoin='miter' 的情况)
- setLineDash(segments) : 将描边设置为虚线 , 可以通过getLineDash() 方法获取虚线样式
- lineDashOffset : 虚线的偏移量

| lineWidth 线宽

lineWidth 定义描边的宽度，它是从路径的中心开始绘制的，内外各占宽度的一半。



lineCap 线条端点样式



lineJoin 拐角类型



miter



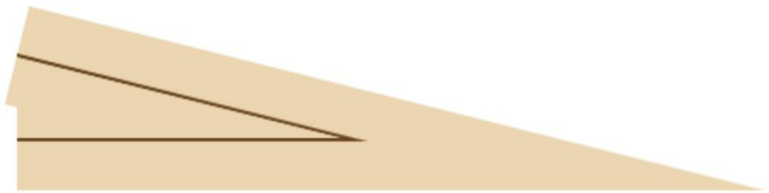
round



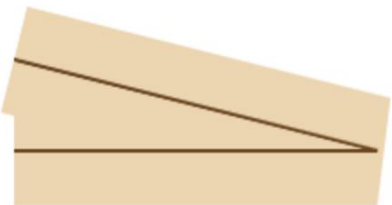
bevel

miterLimit

当lineJoin 为miter 时，若拐角过小，拐角的厚度就会过大。



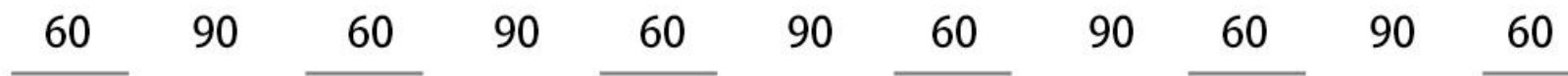
ctx.miterLimit=1 后，可避免此问题



setLineDash(segments) 虚线

```
ctx.setLineDash([ 60, 90 ]);
```

60 90 60 90 60 90 60 90 60 90 60

A horizontal line with a repeating pattern of solid and dashed segments. The solid segments are labeled '60' and the dashed segments are labeled '90'. The pattern is: solid (60), dashed (90), solid (60), dashed (90), solid (60), dashed (90), solid (60), dashed (90), solid (60), dashed (90), solid (60).

```
ctx.setLineDash([ 60, 90, 120 ]);
```

60 90 120 60 90 120 60 90 120

A horizontal line with a repeating pattern of solid and dashed segments. The solid segments are labeled '60', '90', and '120'. The pattern is: solid (60), dashed (90), solid (120), dashed (60), solid (90), dashed (120), solid (60), dashed (90), solid (120).

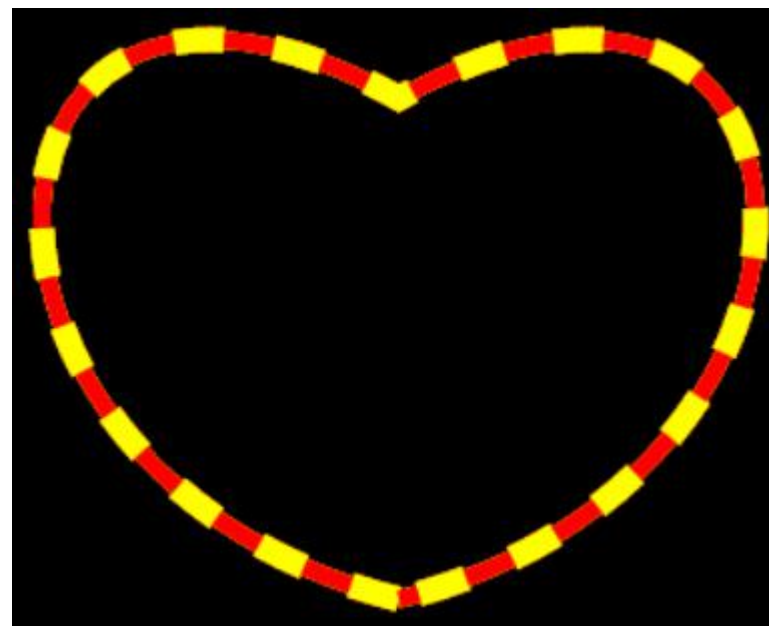
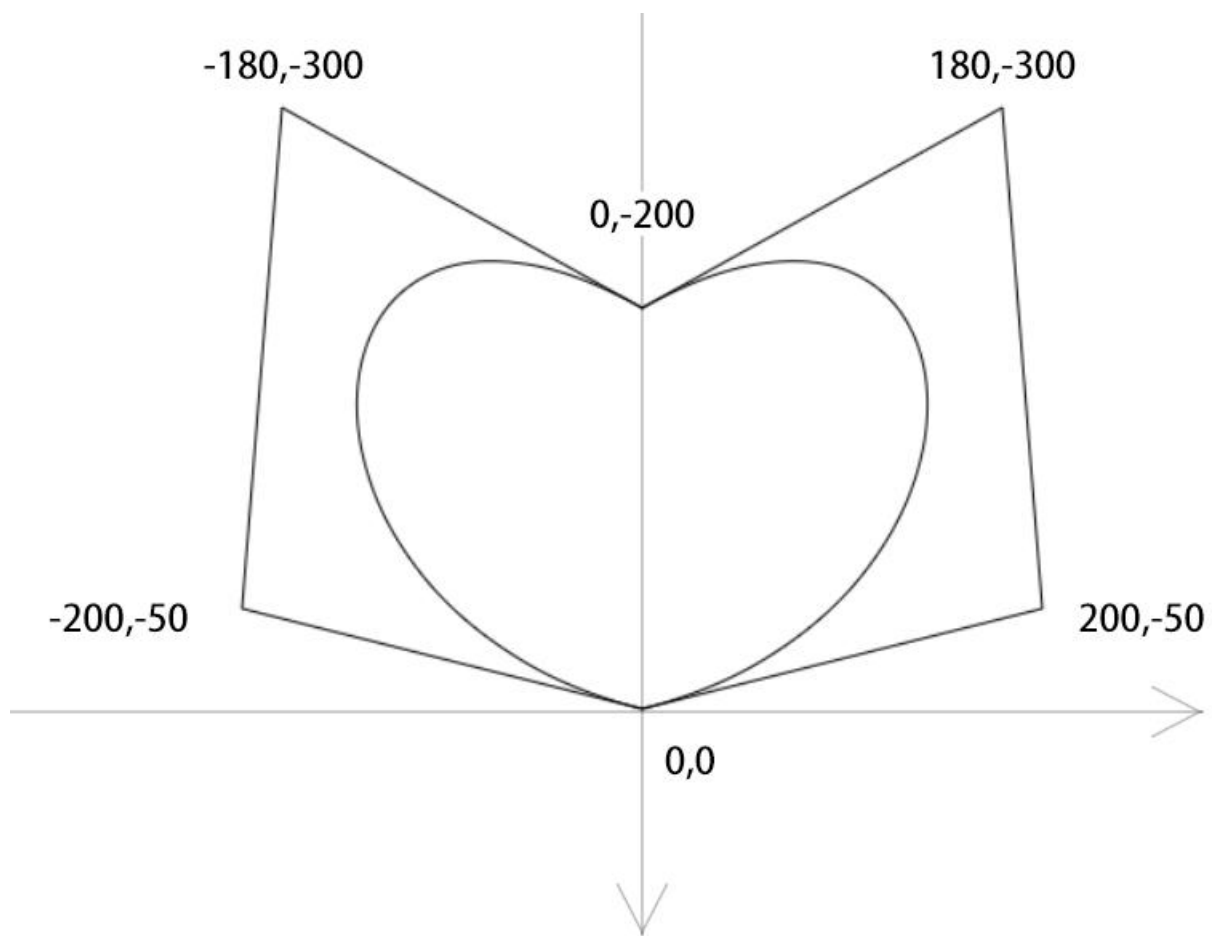
lineDashOffset 虚线偏移

ctx.lineWidth=10
ctx.lineDashOffset=(60 90 60 90 60 90 60 90 60 90 60)

ctx.lineDashOffset=6 (60 90 60 90 60 90 60 90 60 90)

ctx.lineDashOffset=15 (60 90 60 90 60 90 60 90 60 90)

霓虹灯实例

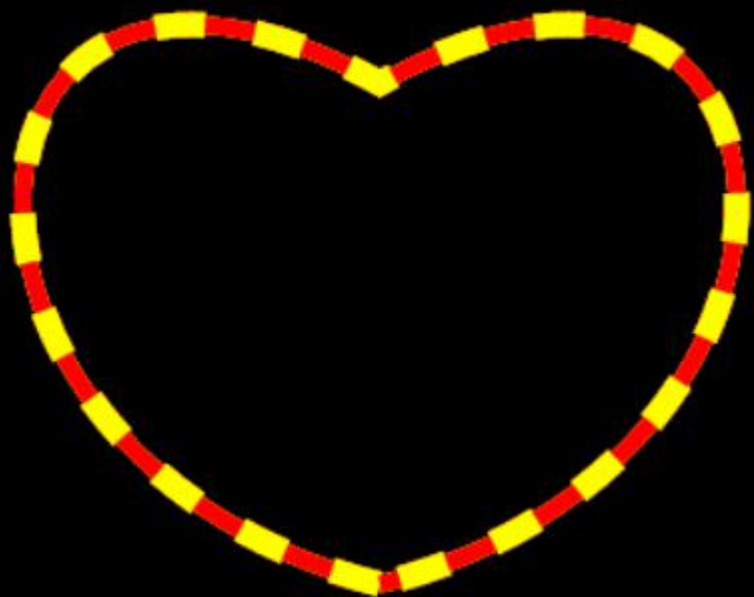


投影的属性

- 位置：
 - shadowOffsetX = float
 - shadowOffsetY = float
- 模糊度： shadowBlur = float
- 颜色： shadowColor = color

注意：投影是上下文对象的一种属性，在绘制图形时，无论执行的是描边方法，还是填充方法，都会在其所绘图形的后面添加投影。

霓虹灯+光晕



注意

在绘制描边图形和填充图形的时候，都需要执行相应的方法，比如路径的fill()、stroke() 方法。

而绘制投影时，则不需要任何方法，只要设置相应属性即可，这样在之后绘制图形的时候都会自带投影，无论这个图形是描边图形还是填充图形。

总结

canvas 的样式相当于衣服，canvas 图形就相当于衣服架子，无论多漂亮的样式都需要图形来承载。