HTML5程序设计基础

第十章 画布（一）

主要内容

01 创建canvas

02 绘制直线、多边形

03 绘制弧和圆

04 绘制矩形

05 透明度

01 创建canvas

Canvas简介

- HTML5 canvas 元素用于绘制图形。canvas 元素只是一块无色透明的图形容器区域，需要利用 JavaScript 脚本来完成绘画。
- 画布是一个空白矩形区域，可以控制其每一像素。
- 可以通过多种方法使用 canvas 绘制路径、矩形、圆形、字符以及添加图像。

创建Canvas元素

```
<canvas id="canvas1" width="200" height="100">
        您的浏览器不支持 canvas 元素，请更换或更新浏览器
</canvas>
```

- canvas默认大小是300*150
- 建议直接设置width和height属性，同时改变canvas元素的大小和元素绘图表面的大小。如通过CSS来设定，则只会改变canvas的大小。当canvas元素的大小和绘图表面的大小不一致时，浏览器会对绘图表面进行缩放，使其符合元素的大小。
- 在设置canvas的宽度和高度时，不推荐使用 px 后缀。

demo10-1.html

getContent()方法

- 使用<canvas>元素，首先需要调用getContext( )方法。
- 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。context被称为绘图环境对象，包含绘图的上下文环境。

```
var elem=document.getElementById( ‘canvas1’ );
var context=elem.getContext( ‘2d’ );
```

注意：该方法可以接受两个值：2d和3d，分别表示二维和三维

demo10-2.html

在Canvas中绘制图形的步骤

1. 在页面添加 canvas 元素，定义 id 属性以便后续调用。

```
<canvas id="myCanvas" width="500" height="500"></canvas>
```

2. 使用id寻找 canvas 元素。

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

3. 通过 canvas 元素的 getContext 方法来获取其上下文，创建 Context 对象，以获取允许进行绘制的 2D 环境。

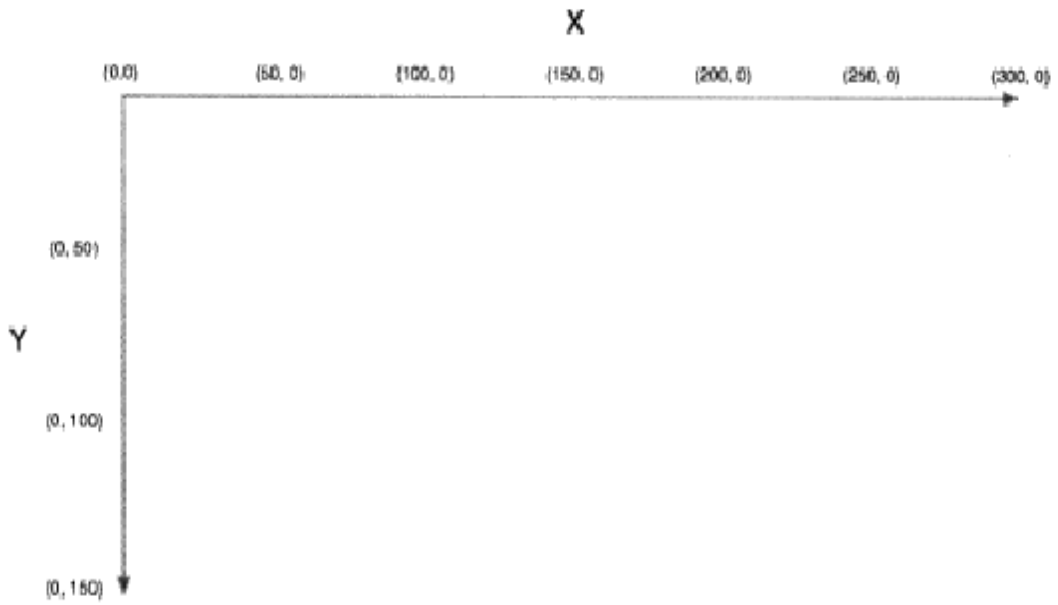
```
var context = c.getContext("2d");
```

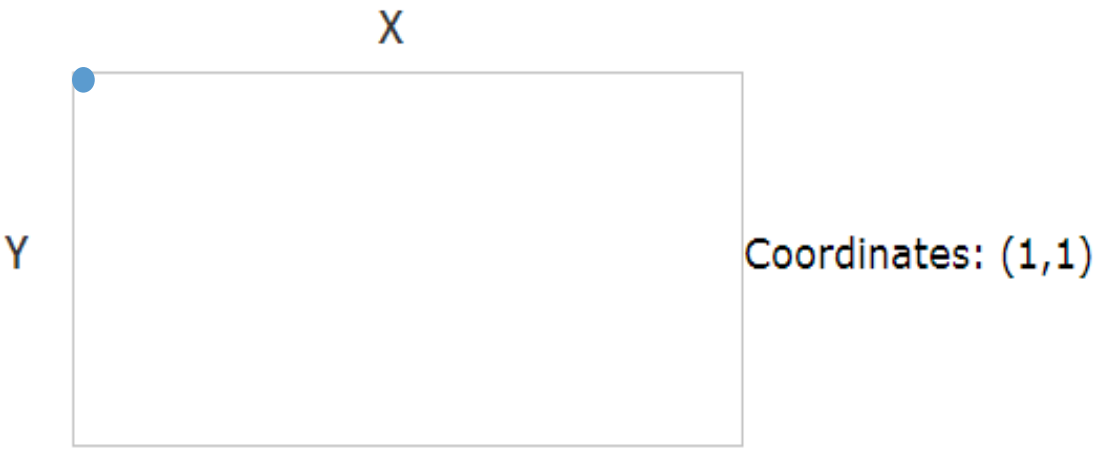
4. 使用 JavaScript 进行绘制。

02 绘制直线、多边形

Canvas坐标及方法

- canvas 是一个二维网格。
- canvas 坐标原点 (0,0) 位于 canvas 的左上角。x 轴水平向右延伸，y轴垂直向下延伸。





路径和描绘

- moveTo(x,y)
  - 用于创建新的子路径，并规定其起始点为（ x,y ）。
- lineTo(x,y)
  - 为当前子路径添加一条直线。这条直线从当前点开始，到 (x, y) 结束。当方法返回时，当前点是 (x,y) 。
- stroke( )
  - 实际地绘制出通过 moveTo() 和 lineTo() 方法定义的路径。默认颜色是黑色。

线条绘制

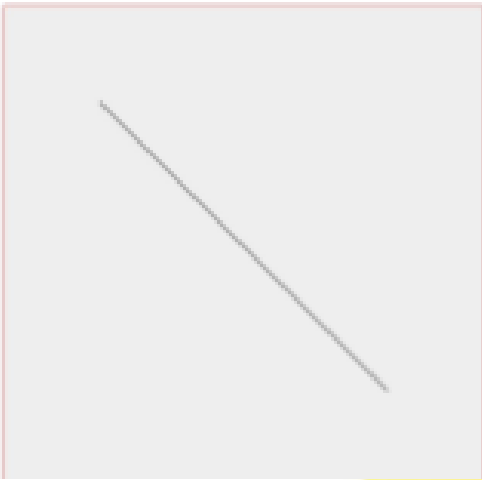
- context. moveTo(100,100)
- context. lineTo(300,300)

} 路径设置

- context. stroke()

} 绘制

- canvas 是基于路径的绘图



demo10-3.html

1. moveTo(x,y) 只是设置起点并不画线。
2. 如果没有用moveTo() 规定子路径的起点，则 lineTo(x,y)等同于moveTo(x,y)。
3. 连完一条线后起点改变，改变为 lineTo(x,y) 中的坐标。

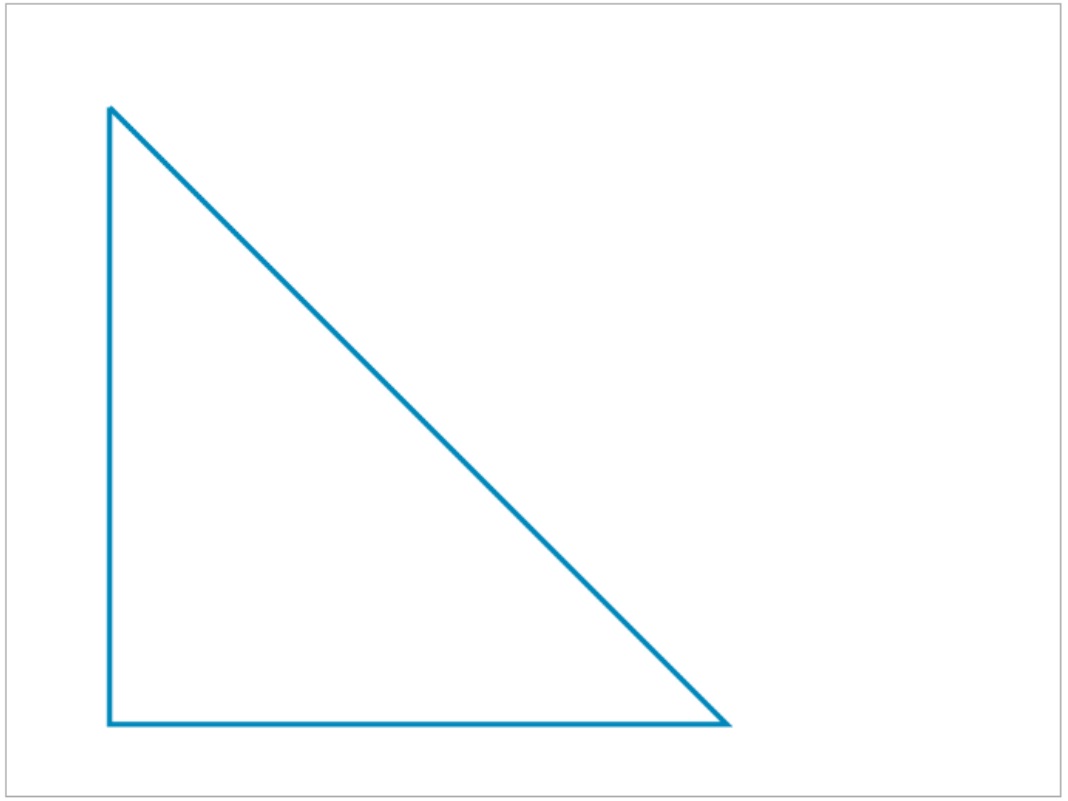
线条样式设置

- lineWidth 属性
  - 设置当前线条的宽度，默认单位为像素
  - context.lineWidth = 5;
- strokeStyle 属性
  - 设置用于绘制描边的颜色
  - context.strokeStyle = "red";

demo10-3.html

练习

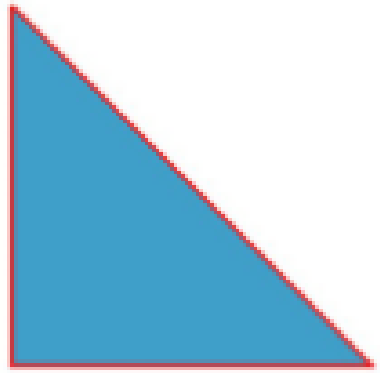
- 绘制如下三角形



demo10-4.html

填充

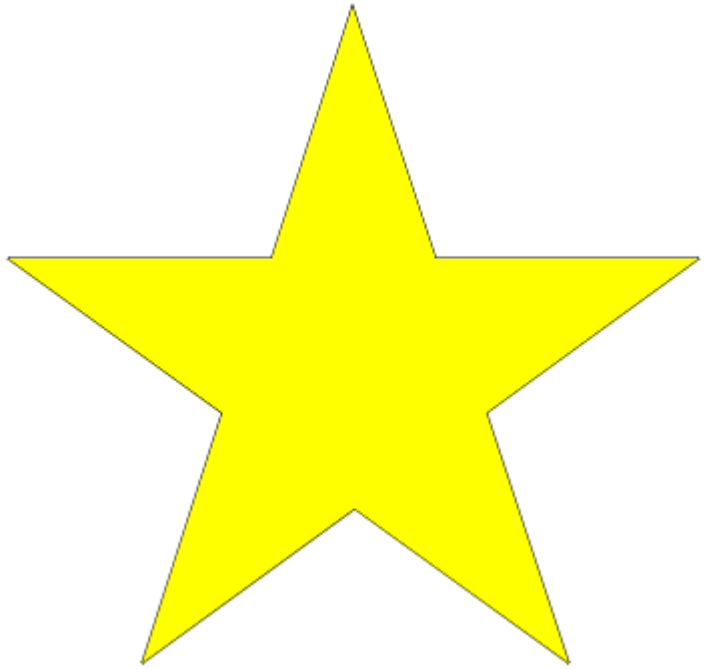
- fill( ) 方法
  - 填充当前绘图（路径）。默认颜色是黑色。
  - 如果路径未关闭，那么 fill() 方法会自动从路径结束点到开始点之间添加一条线，以关闭该路径，然后填充该路径。
- fillStyle 属性
  - 设置用于填充绘画的颜色。
  - context.fillStyle="red";

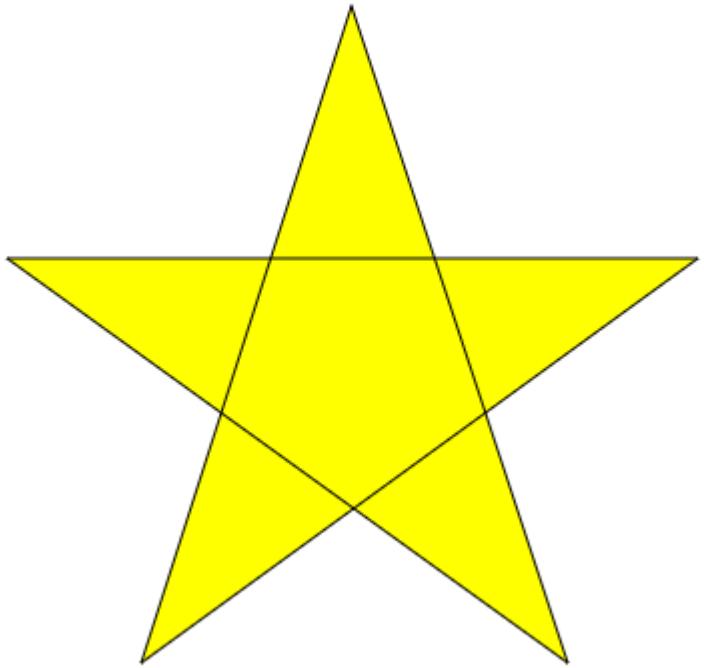


demo10-5.html

填充

- 建议先填充再描边





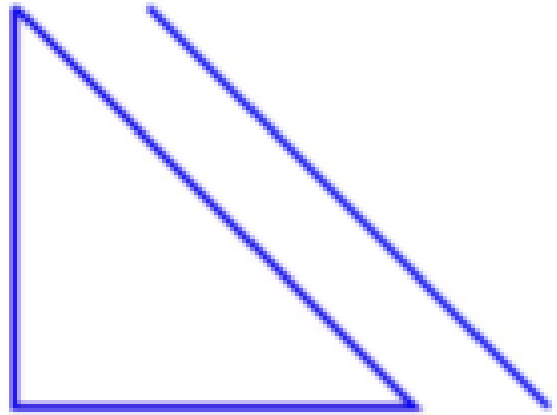
demo10-6.html

深入canvas基于路径绘图

```
context.moveTo(100,100);
context.lineTo(100,400);
context.lineTo(400,400);
context.lineTo(100,100);

context.lineWidth=5;
context.strokeStyle ="red";
context.stroke();

context.moveTo(200,100);
context.lineTo(800,700);
context.strokeStyle = "blue"
context.stroke();
```



demo10-7.html

路径封闭

- beginPath( )
  - 开始一条新的路径（路径开始点）
- closePath( )
  - 创建从当前点到开始点的路径（路径结束点）。
  - 关闭一条打开的子路径。
- canvas 之中只能有一条路径存在，称之为“当前路径”(current path)。

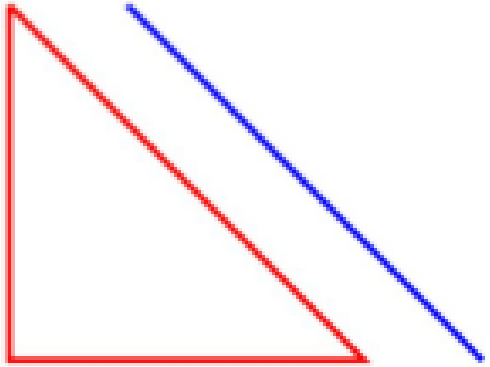
深入canvas基于路径绘图

```
context.moveTo(100,100);
context.lineTo(100,400);
context.lineTo(400,400);
context.lineTo(100,100);

context.lineWidth=5;
context.strokeStyle ="red";
context.stroke();

context.beginPath();
context.moveTo(200,100);
context.lineTo(500,400);
context.strokeStyle = "blue"
context.stroke();
context.closePath();
```

第一条路径可省略



demo10-8.html

小结

content.moveTo(x,y)	把路径移动到画布中的指定点
content.lineTo(x,y)	为当前子路径添加一条直线。这条直线从当前点开始，到 (x, y) 结束。
content.beginPath( )	开始一条新的路径
content.closePath( )	创建从当前点到开始点的路径
content.lineWidth	设置当前线条的宽度
content.strokeStyle	设置用于绘制描边的颜色
content.fillStyle	设置用于填充绘画的颜色
content.stroke( )	绘制路径
content.fill( )	填充当前绘图

练习

- 七巧板

demo10-9.html

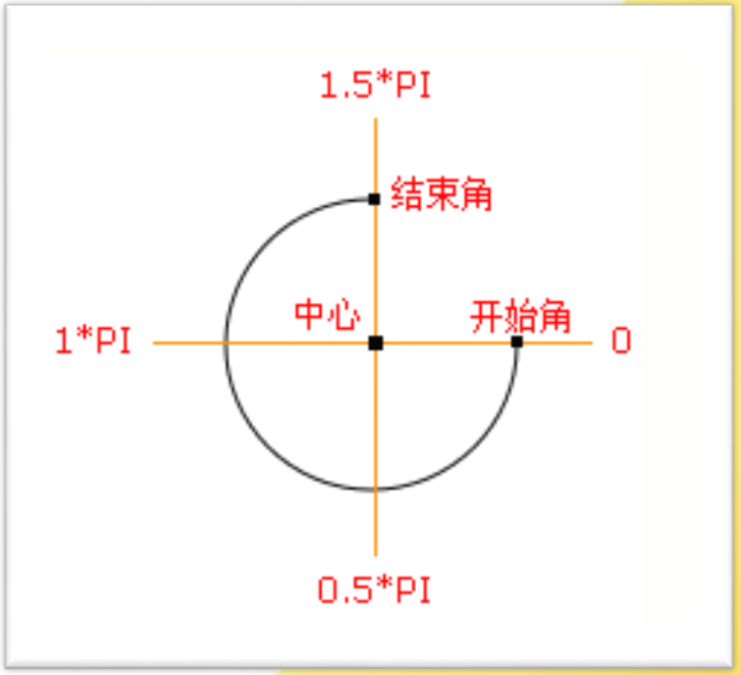
03 绘制弧和圆

弧/曲线

- arc() 方法创建弧/曲线（用于创建圆或部分圆）

```
context.arc(x, y, r, sAngle, eAngle, anticlockwise)
```

参数	描述
x	圆的中心的 x 坐标。
y	圆的中心的 y 坐标。
r	圆的半径。
sAngle	起始角，以弧度计
eAngle	结束角，以弧度计。
anticlockwise	可选，是否按照逆时针方向绘图。 false = 顺时针 true = 逆时针



绘制弧线

```
var c1 = document.getElementById("canvas1");
var context1 = c1.getContext("2d");
context1.lineWidth=5;
context1.strokeStyle = “#005588”;
context1.arc(300,300,200,0,1.5*Math.PI);
context1.stroke();
```

demo10-10.html

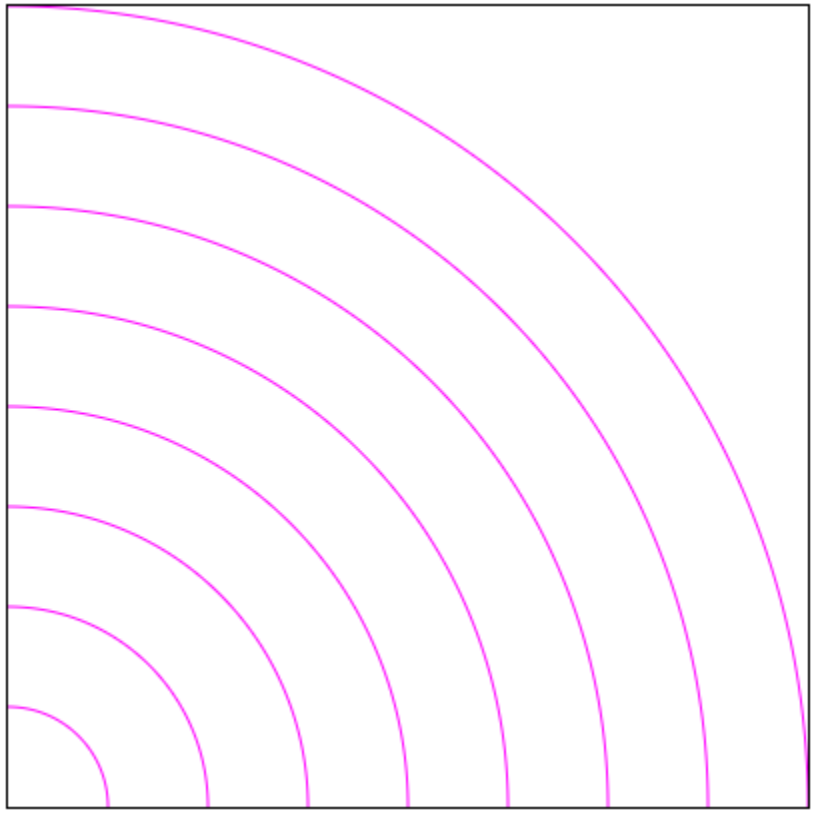
绘制多段弧线

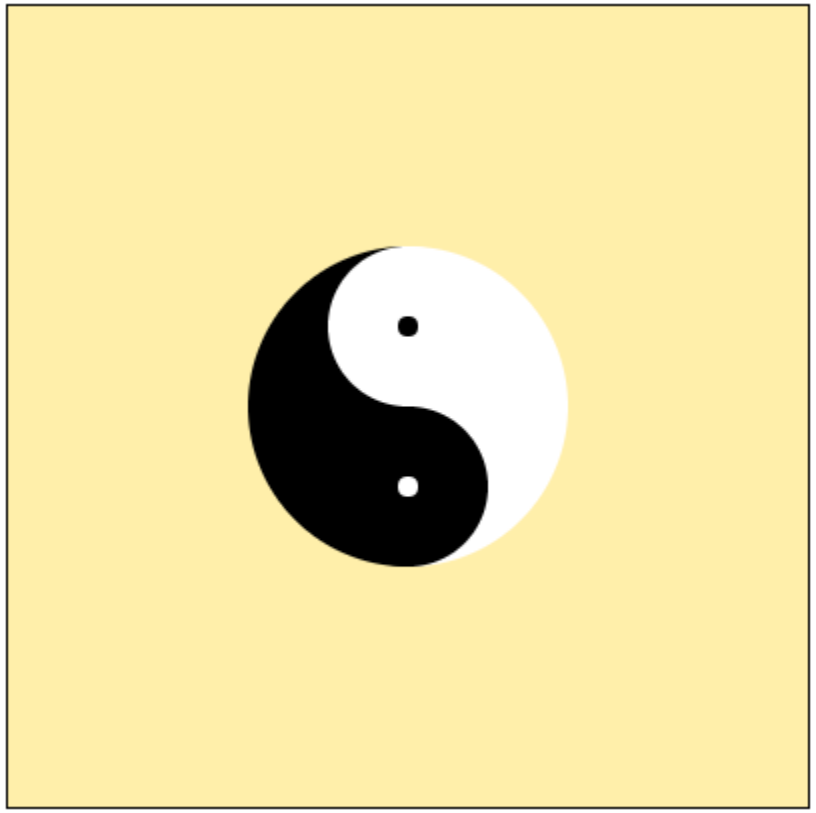
```
for (i = 0; i < 10; i++) {
    context.beginPath(); //开始路径
    context.arc(50 + 100 * i, 60, 40, 0, 2 * Math.PI * (i + 1) / 10, true);
    context.closePath(); //关闭路径
    context.stroke(); //绘制路径
 }
```

demo10-11.html

练习

- 同心圆、八卦





demo10-12.html

04 绘制矩形

绘制矩形

```
context.moveTo(x, y);
context.lineTo(x + width, y);
context.lineTo(x + width, y + height);
context.lineTo(x, y + height);
```

- context.rect(x, y, width, height)
  - x，y 指定矩形左上角的位置
  - width，height 指定矩形的尺寸
  - 与 fill( ) 和 stroke( ) 搭配使用

demo10-13.html

绘制矩形

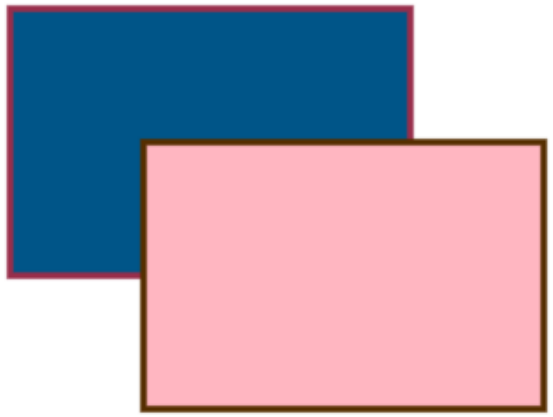
- 矩形相关方法
  - rect(x，y，width，height);　　创建矩形
  - fillRect(x，y，width，height)；　　绘制 "已填色" 的矩形（实心）
  - strokeRect(x，y，width，height); 绘制不填色的矩形（空心）
  - clearRect(x，y，width，height);　清空给定矩形内的指定像素

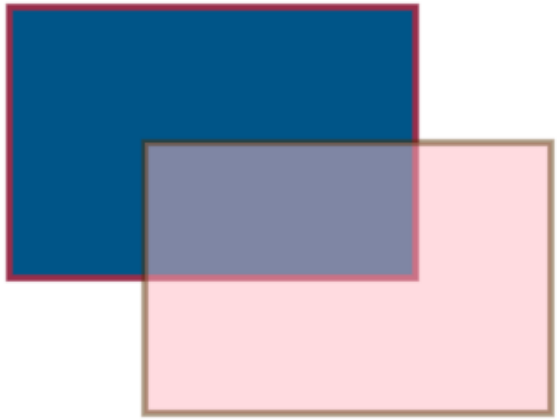
demo10-14.html

05 透明度

透明度

- globalAlpha 属性
  - 设置绘图的当前透明值。



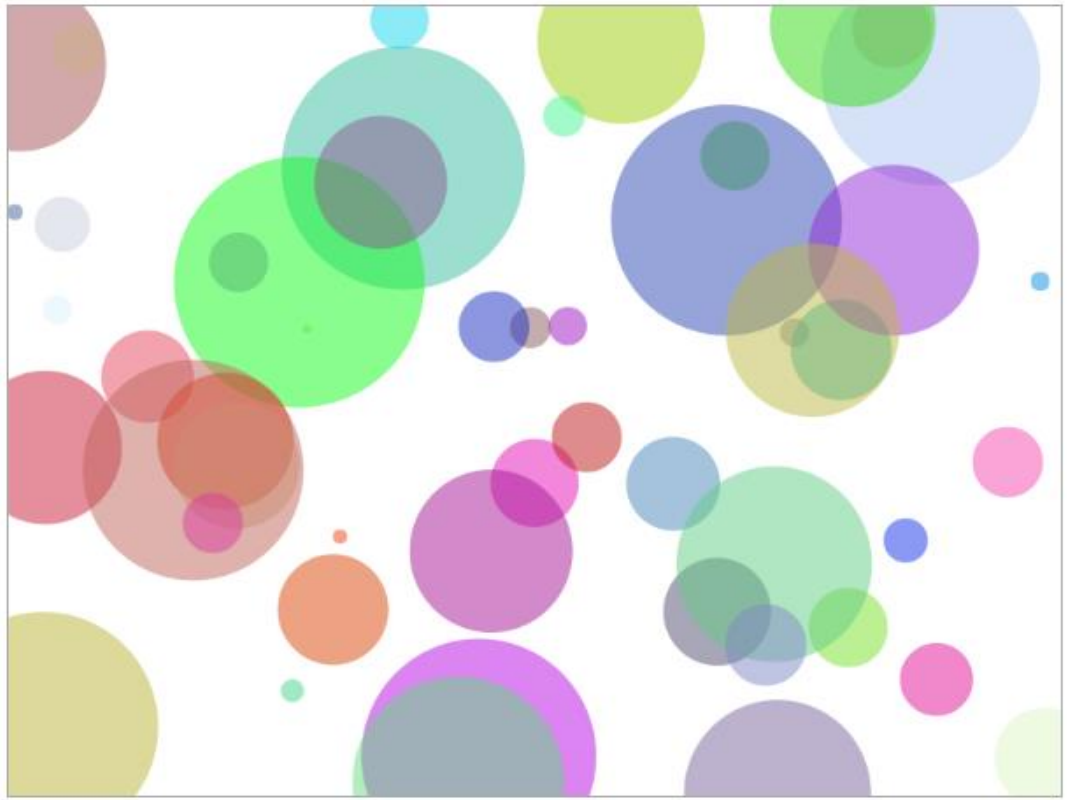


```
context.globalAlpha = 0.5;
context.fillStyle = "rgba(255,0,0,0.5)";
```

demo10-15.html

练习

- 在画布上绘制50个任意大小、任意位置、透明度为0.5的任意颜色的圆形。



demo10-16.html

THANKYOU