


canvas 1

介绍

- 是HTML5重要元素，不需要外部插件支持
- 提供了强大的图形处理功能
- 常用于移动端web的开发
- 基本现代浏览器都支持，IE8以下不支持
  - <canvas>您的浏览器不支持canvas</canvas>
- 进行绘制，是通过context环境进行操作的
  - var canvas = document.querySelector(“canvas”)
  - var context = canvas.getContext(“2d”)
- 坐标系
  - 横轴向右为正
  - 纵轴向下为正

beginPath()	开始绘制
moveTo(x,y)	设置绘制起点
lineTo(x,y)	设置下一个点
closePath()	结束绘制,会从当前点回到结束点,形成一个封闭的图形
strokeStyle	设置绘制的样式
stroke()	绘制点之间的路线
fillStyle	设置填充样式
fill()	填充当前绘图
lineWidth	线宽

基本图形绘制

- 1. 绘制直线
  - context.beginPath() 开始绘制
  - context.moveTo(0,0) 设置起点
  - context.lineTo(100,100) 设置下一个点
  - context.closePath() 结束绘制
- 2. 绘制矩形
  - context.strokeRect(x,y,width,height)
  - context.fillRect(x,y,width,height)
- 3. 绘制圆形
  - context.arc(x,y,radius,startAngle,endAngle,anticlockwise);
    - x/y 圆心坐标
    - radius 半径
    - startAngle 起始角度 说白了就是确定起点位置
    - endAngle 结束角度 确定终点位置
    - anticlockwise 是否逆时针
      - true 逆时针
      - false 顺时针
  - context.lineWidth = 4;
  - context.strokeStyle = "red";
  - context.stroke();
  - context.fillStyle = "green";
  - context.fill();
- 4. 绘制文字
  - var text = "Hello World";
  - context.font = "italic 50px Zapfino"; 设置字体
  - var gradient = context.createLinearGradient(x1,y1,x2,y2); 渐变范围从(x1,y1)到(x2,y2)
  - gradient.addColorStop("0","blue"); 添加渐变色
  - gradient.addColorStop("0.5","green"); 添加渐变色
  - gradient.addColorStop("1","red"); 添加渐变色
  - context.strokeStyle = gradient;
  - context.textBaseline = "top"; 调整基线
  - 
  - context.strokeText(text,x,y,maxWidth);

图形变换

- 平移 context.translate(x,y)
- 旋转 context.rotate(Math.PI/2)
- 缩放 context.scale(x1,y1)

阴影设置

- |               |                           |
|---------------|---------------------------|
| shadowColor   | 阴影颜色                      |
| shadowOffsetX | X方向偏移量                    |
| shadowOffsetY | Y方向偏移量                    |
| closePath()   | 结束绘制,会从当前点回到结束点,形成一个封闭的图形 |
| shadowBlur    | 设置阴影的模糊级别                 |
- context.shadowOffsetX = 2;
- context.shadowOffsetY = -2;
- context.shadowColor = "yellow";
- context.shadowBlur = 50;

贝塞尔曲线路径

- 二次贝塞尔 context.quadraticCurveTo(dx,dy,x,y)
- 三次贝塞尔 context.bezierCurveTo(500,0,0,500,500,500)