

SHANXI **UNIQUE**  
TECHNOLOGY CO.,LTD.



优逸客

SHANXI UNIQUE TECHNOLOGY

# html5 canvas路径

- canvas路径指存以贝塞尔曲线为理论基础的区域绘制方式，绘制时产生的线条称为路径。路径由一个或多个直线段或曲线段组成，或者是经过精确计算画出的特殊图形，路径是canvas实现绘图的基础。

方法	描述
<code>fill()</code>	填充当前绘图（路径）
<code>stroke()</code>	绘制已定义的路径
<code>beginPath()</code>	起始一条路径，或重置当前路径
<code>moveTo()</code>	把路径移动到画布中的指定点，不创建线条
<code>closePath()</code>	创建从当前点回到起始点的路径
<code>lineTo()</code>	添加一个新点，然后在画布中创建从该点到最后指定点的线条
<code>clip()</code>	从原始画布剪切任意形状和尺寸的区域
<code>quadraticCurveTo()</code>	创建二次贝塞尔曲线
<code>bezierCurveTo()</code>	创建三次方贝塞尔曲线
<code>arc()</code>	创建弧/曲线（用于创建圆形或部分圆）
<code>arcTo()</code>	创建两切线之间的弧/曲线
<code>isPointInPath()</code>	如果指定的点位于当前路径中，则返回 <b>true</b> ，否则返回 <b>false</b>

- `rect(x,y,w,h)`
- `stroke()`
  - 绘制路径
- `fill()`
  - 填充图形



- moveTo(x,y)
  - 开始绘制一条直线，指定线条的起点
- lineTo(x1,y1)
  - 指定直线要到达的位置
- stroke()
  - 绘制路径



# 路径(线条)修饰

属性	描述
lineCap	设置或返回线条的结束端点样式 butt:默认。向线条的每个末端添加平直的边缘 round:向线条的每个末端添加圆形线帽。 square:向线条的每个末端添加正方形线帽
lineJoin	设置或返回两条线相交时，所创建的拐角类型 miter:默认。创建尖角;bevel:创建斜角。 round:创建圆角。
lineWidth	设置或返回当前的线条宽度 number:当前线条的宽度，以像素计
miterLimit	设置或返回最大斜接长度 number:正数。规定最大斜接长度.5 斜接长度指的是在两条线交汇处内角和外角之间的距离 只有当 lineJoin 属性为 "miter" 时，miterLimit 才有效



- `beginPath()`
  - 开始一条路径，或重置当前的路径。
- `closePath()`
  - 创建从当前点到开始点的路径，关闭当前的绘图路径
- 注意:
  - 如果画完前面的路径没有重新指定`beginPath`，那么画第其他路径的时候会将前面最近指定的`beginPath`后的全部路径重新绘制
  - 每次调用`context.fill ( )`的时候会自动把当次绘制的路径的开始点和结束点相连，接着填充封闭的部分



- `quadraticCurveTo(cpx,cpy,x,y)`
  - 通过使用表示二次贝塞尔曲线的指定控制点，向当前路径添加一个点
  - 示例：
    - 开始点：`moveTo(20,20)`
    - 控制点：`quadraticCurveTo(20,100,200,20)`
    - 结束点：`quadraticCurveTo(20,100,200,20)`

参数	描述
cpx	贝塞尔控制点的 x 坐标
cpy	贝塞尔控制点的 y 坐标
x	结束点的 x 坐标
y	结束点的 y 坐标



- `bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y)`
  - 通过使用表示三次贝塞尔曲线的指定控制点，向当前路径添加一个点
  - 示例:
    - 开始点：`moveTo(20,20)`
    - 控制点 1：`bezierCurveTo(20,100,200,100,200,20)`
    - 控制点 2：`bezierCurveTo(20,100,200,100,200,20)`
    - 结束点：`bezierCurveTo(20,100,200,100,200,20)`

参数	描述
cp1x	第一个贝塞尔控制点的 x 坐标
cp1y	第一个贝塞尔控制点的 y 坐标
cp2x	第二个贝塞尔控制点的 x 坐标
cp2y	第二个贝塞尔控制点的 y 坐标
x	结束点的 x 坐标
y	结束点的 y 坐标

# 绘制弧线（用于创建圆形或部分圆）

- `arc(x,y,r,sAngle,eAngle,counterclockwise);`
  - 创建弧/曲线（用于创建圆或部分圆）

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。（弧的圆形的三点钟位置是 0 度）。
eAngle	结束角，以弧度计。
counterclockwise	可选。规定应该逆时针还是顺时针绘图。False = 顺时针，true = 逆时针。

- `arcTo(x1,y1,x2,y2,r)` 方法在画布上创建介于两个切线之间的弧/曲线.

参数	描述
x1	弧的起点的 x 坐标
y1	弧的起点的 y 坐标
x2	弧的终点的 x 坐标
y2	弧的终点的 y 坐标
r	弧的半径

## ➤ clip() 方法从原始画布中剪切任意形状和尺寸

- 提示：一旦剪切了某个区域，则所有之后的绘图都会被限制在被剪切的区域内（不能访问画布上的其他区域）。您也可以在使用 clip() 方法前通过使用 save() 方法对当前画布区域进行保存，并在以后的任意时间对其进行恢复
- 使用步骤：
  - 使用.save保证裁切前的绘图状态；
  - 通过context.rect()来确定要裁切的区域在位置；
  - 通过context.clip()来执行裁切动作；
  - 在Canvas上画图，只有在裁切区域的图形才能显示出来；
  - 使用context.restore()再恢复到裁切前的状态

## ➤ isPointInPath(x,y)

- 方法返回 true，说明指定的点位于当前路径中；否则返回 false

参数	描述
x	测试的 x 坐标
y	测试的 y 坐标



谢谢观看...

SHANXI **UNIQUE**  
TECHNOLOGY CO.,LTD.