

canvas

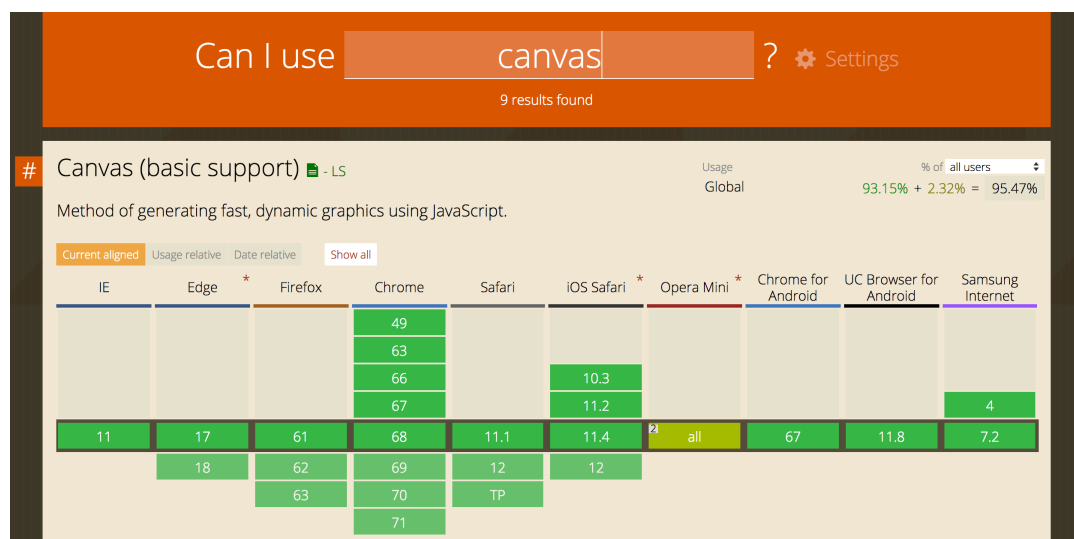
- 简介

`<canvas>` 是 HTML5 新增的，一个可以使用脚本(通常为JavaScript)在其中绘制图像的 HTML 元素。它可以用来制作照片集或者制作简单(也不是那么简单)的动画，甚至可以进行实时视频处理和渲染。

Canvas是由HTML代码配合高度和宽度属性而定义出的可绘制区域。JavaScript代码可以访问该区域，类似于其他通用的二维API，通过一套完整的绘图函数来动态生成图形。

Mozilla 程序从 Gecko 1.8 (Firefox 1.5)开始支持 `<canvas>`，Internet Explorer 从IE9开始`<canvas>`。Chrome和Opera 9+ 也支持`<canvas>`。

方法兼容性：



- 历史来源

苹果内部使用自己MacOS X WebKit推出，供应用程序使用像仪表盘的构件和 Safari 浏览器使用。后来，有人通过Gecko内核的浏览器(尤其是Mozilla和Firefox)，Opera和Chrome和超文本网络应用技术工作组建议为下一代的网络技术使用该元素

- canvas基本使用

1. canvas标签的使用

```

1 <!-- canvas默认宽高300px 150px, css属性同样可以更改宽高但如果和初始化宽高比不统一会出现变形的问 建议不要使用css定宽高!! -->
2 <canvas width="300" height="150">
3   不支持canvas标签会显示该内容~
4 </canvas>

```

2. canvas方法检查支持性

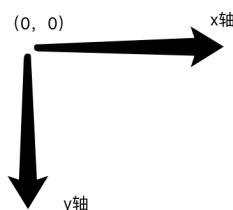
```

1 //canvas标签需要考虑浏览器版本兼容性、同样其对应的方法也应该考虑不支持的处理方式!!
2 var canvas = document.getElementById("target");
3 if( canvas.getContext ){
4   var ctx = canvas.getContext("2d");
5 }else{
6   alert("该浏览器版本过低, 请更换~")
7 }

```

3.canvas 绘制图形

☐ 坐标系



☐ 矩形

`fillRect(x , y , width , height)` //填充以(x,y)为起点宽高分别为width、height的矩形 默认为黑色

`strokeRect(x , y , width , height)` //绘制一个空心以(x,y)为起点宽高分别为width、height的矩形

`clearRect(x , y , width , height)` // 清除以(x,y)为起点宽高分别为width、height的矩形 为透明

☐ 路径

`beginPath()` 新建一条路径一旦创建成功 绘制命令将转移到新建的路径上

`moveTo(x , y)` 移动画笔到(x , y) 点开始后面的绘制工作

`closePath()` 关闭该路径 将绘制指令重新转移到上下文

`stroke()` 将绘制的路径进行描边

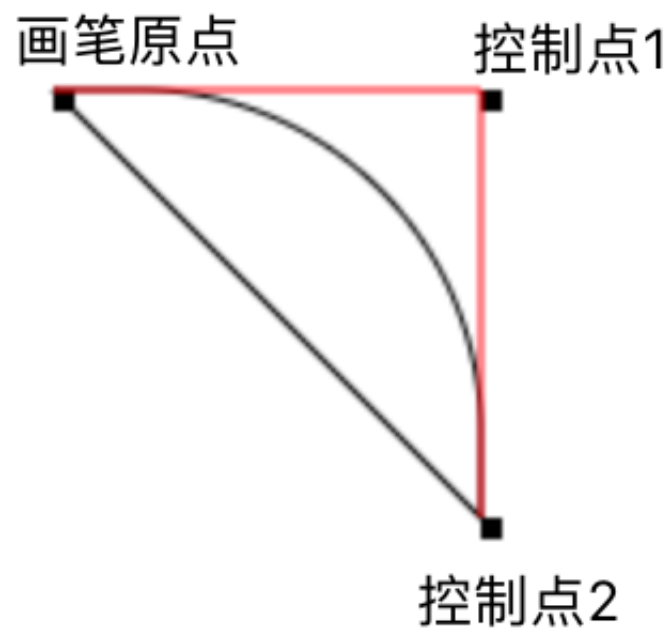
`fill()` 将绘制的封闭区域进行填充

☐ 圆弧

`arc(x , y , r , startAngle , endAngle , anticlosewise)` // 以(x,y)为圆心 r为半径的圆 绘制startAngle弧度 到endAngle弧度的圆弧 anticlosewise默认为false 即顺时针方向 true为逆时针方向

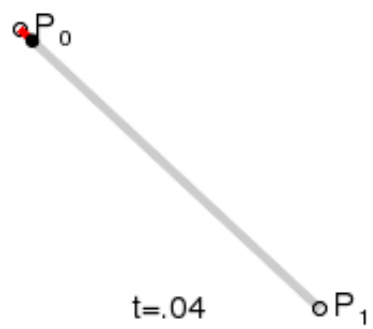
`arcTo(x1 , y1 , x2 , y2 , radius)` //根据 两个控制点 (x1,y1) 和 (x2, y2)以及半径绘

制弧线 同时连接两个控制点

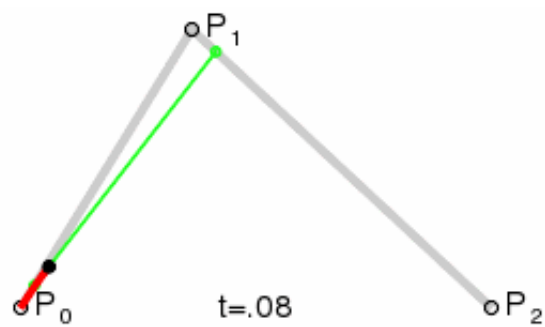


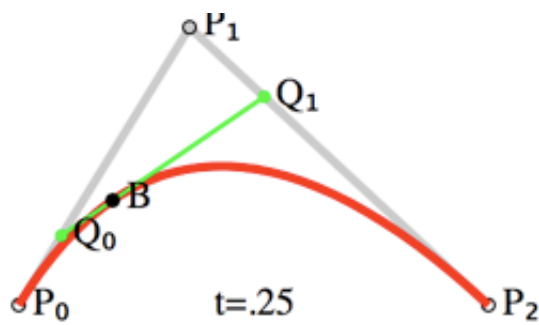
□ 贝塞尔曲线

一次贝塞尔曲线其实是一条直线。

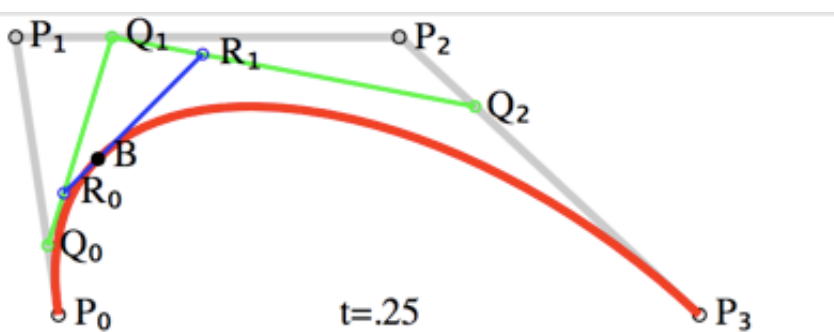
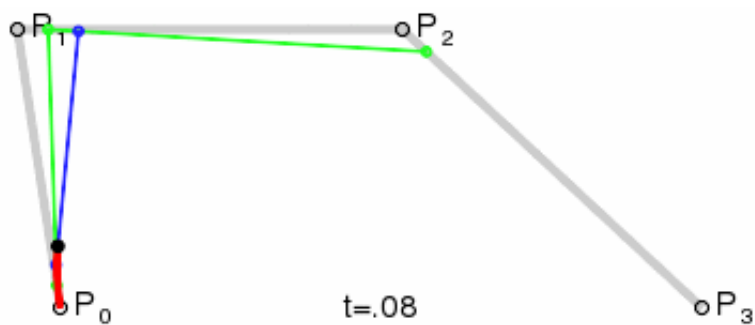


二次贝塞尔曲线





三次贝塞尔曲线



二次贝塞尔曲线

`quadraticCurveTo(cp1x, cp1y , x , y)` // (cp1x,cp1y) 控制点 (x,y)结束点

三次贝塞尔曲线

`bezierCurveTo(cp1x, cp1y ,cp2x , cp2y ,x , y)`// (cp1x,cp1y) 控制点1

(cp2x,cp2y) 控制点2 (x,y)结束点

□ 样式添加

`fillStyle = color`

`strokeStyle = color`

color 可以为颜色值、渐变对象(并非样式!!!!)、图片对象([createPattern](#) 目

前没有试验成功!!!)

lineWidth = value 线宽

lineCap = type (butt 、 round 、 square) 线条末端样式 依次是方形、圆形&突出、方形&突出



lineJoin = type (round 、 bevel 、 miter) 线条交汇处样式 依次是圆形、平角、三角形



ctx.setLineDash([实际长度 , 间隙长度]) & ctx.lineDashOffset 虚线

setLineDash接受数组 而lineDashOffset来设置偏移量

☐ 渐变

var gradient = ctx.createLinearGradient(x1 ,y1 ,x2 ,y2); //线性渐变

var gradient = ctx.createRadialGradient(x1 ,y1 ,r1 ,x2 ,y2 ,r2); //径向渐变

gradient.addColorStop(position , color)// position:相对位置0~1 color:该位置下的颜色

☐ 透明度

ctx.globalAlpha = value (0~1)

☐ 文本

fillText(text , x , y ,[,maxWidth]) 在(x,y)位置绘制text文本 最大宽度为maxWidth(可选)

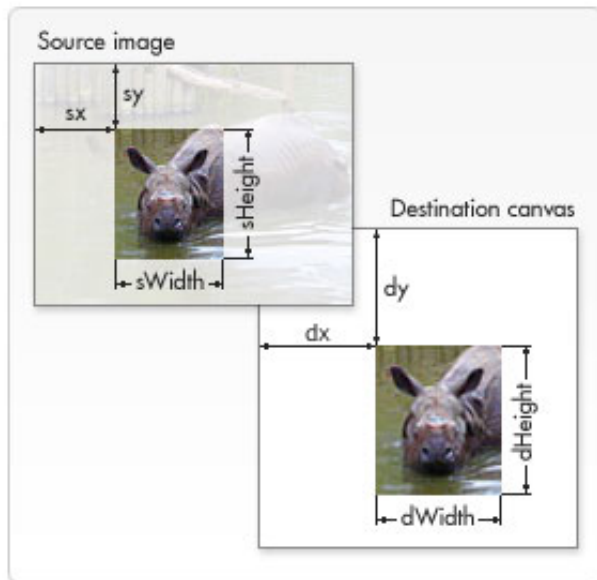
strokeText(text ,x ,y ,[,maxWidth]) 在(x,y)位置绘制text文本边框 最大宽度为maxWidth(可选)

font = value eg:"100px sans-serif"

☐ 绘制图片

`drawImage(image , x , y , width , height)` image为图片对象、从(x,y)处放置宽高分别为width height的图片

`drawImage(image , sx , sy , sWidth , sheight ,dx ,dy ,dwidth ,dheight)` 切片
前四个是定义图像源的切片位置和大小 后四个是定期切片的目标显示位置大小



☐ 状态保存 恢复

`save()` `restore()`

☐ `translate(x , y)` 将canvas原点的移动到 (x,y) (save&restore保存初始状态!!!)

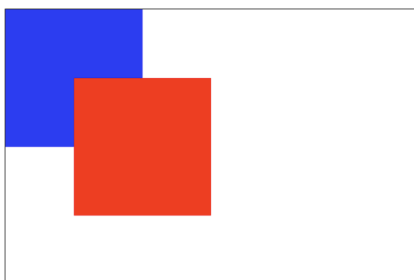
☐ `rotate(angle)` 顺时针方向旋转坐标轴 angle弧度

☐ `scale(x,y)` 将图形横向缩放x倍、纵向缩放y倍 (x、y大于1是放大 小于1为缩放!!!)

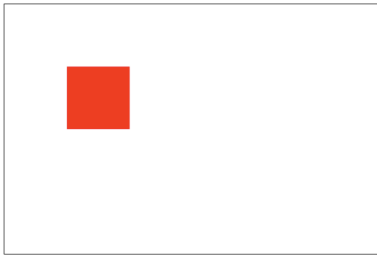
☐ 全局合成操作

`globalCompositeOperation = type;`

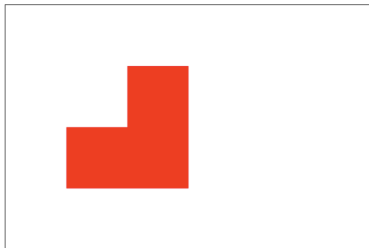
`source-over`



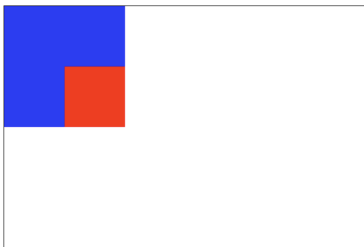
source-in



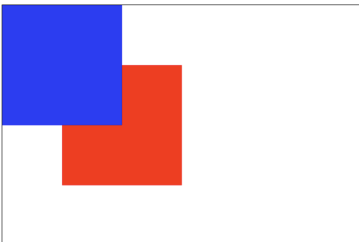
source-out



source-atop



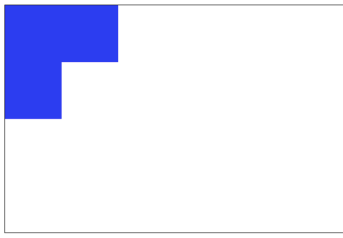
destination-over



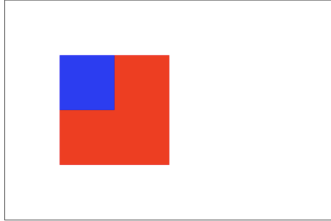
destination-in



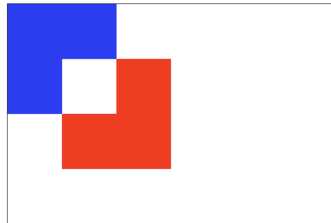
destination-out



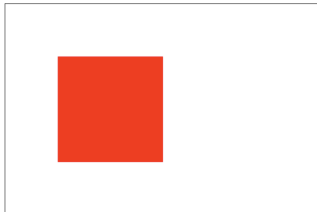
destination-atop



xor



copy



☐ 裁剪

clip 只显示裁剪区域内部区域（使用save & restore 存储canvas状态！！！）

☐ 动画

clearRect() 清空画布

save&restore 保存恢复canvas状态

定时执行

setInterval()

setTimeout()

requestAnimationFrame()

区别详见：<https://www.cnblogs.com/xiaohuochai/p/5777186.html>

svg

- 标签

矩形 rect

圆形 circle

椭圆 ellipse

线 line

折线 polyline

多边形 polygon

路径 path

文本 text

详细标签：<http://www.runoob.com/svg/svg-reference.html>

- 属性

宽度 width

高度 height

css样式 style

填充色 fill

边框颜色 stroke

1. 边框宽度 stroke-width

2. 边框首尾 stroke-linecap

3. 线条样式 stroke-dasharray (需要填写线条宽度以及线条间隙 依据顺序填写一个循环)

4. 线条每一段的起始偏移量 stroke-dashoffset

填充透明度 fill-opacity

边框 stroke-opacity

图形填充规则 fill-rule (nonzero evenodd)

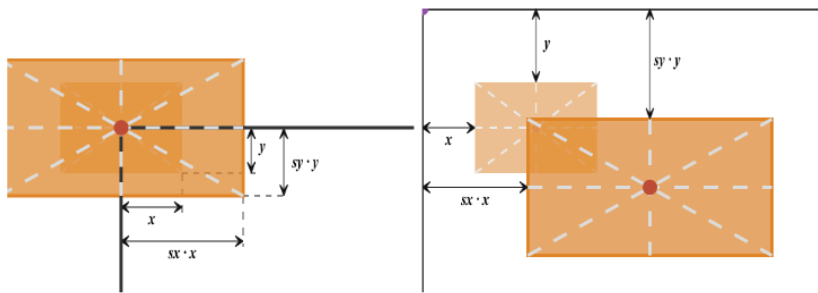
动作 transform (中心点为图像左上角,并且只支持2d变换) 详

情：<https://www.zhangxinxu.com/wordpress/2015/10/understand-svg-transform/>

1. translate(10 ,10) or translate(10 10)

2. rotate(20) or rotate(20 , x y) x,y为旋转中心点 并且只能是默认deg单位

3. scale(x , y)



viewBox 属性 详情: <https://www.zhangxinxu.com/wordpress/2014/08/svg-viewport-viewbox-preserveaspectratio/>

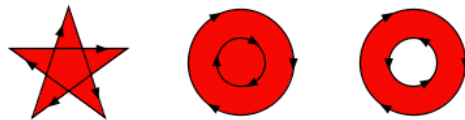
1. viewBox = "x , y , width , height"

preserveAspectRatio属性 详

情: <https://www.zhangxinxu.com/wordpress/2014/08/svg-viewport-viewbox-preserveaspectratio/>

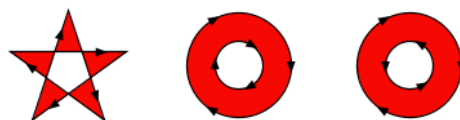
nonzero

字面意思是“非零”。按该规则，要判断一个点是否在图形内，从该点作任意方向的一条射线，然后检测射线与图形路径的交点情况。从0开始计数，路径从左向右穿过射线则计数加1，从右向左穿过射线则计数减1。得出计数结果后，如果结果是0，则认为点在图形外部，否则认为在内部。下图演示了nonzero规则:



evenodd

字面意思是“奇偶”。按该规则，要判断一个点是否在图形内，从该点作任意方向的一条射线，然后检测射线与图形路径的交点的数量。如果结果是奇数则认为点在内部，是偶数则认为点在外部。下图演示了evenodd规则:



矩形 rect

1. 距离左侧位置 x
2. 距离顶部位置 y
3. 矩形产生圆角 rx

4. 矩形产生圆角 ry

圆形 circle

1. 圆点 cx cy
2. 半径 r

椭圆 ellipse

1. 椭圆中心坐标 cx cy
2. 水平半径 rx
3. 垂直半径 ry

直线 line

1. 开始点 x1 y1
2. 结束点 x2 y2

多边形 polygon

1. 定义多边形每个角的x y 坐标 points (eg:points="200,10 250,190 160,210")

曲线 polyline

1. 定义折线起点 拐点 重点 points(eg: points="0,40 40,40 40,80 80,80 80,120 120,120 120,160")

□ 路径path

M = moveto 移动画笔

L = lineto 画线

H = horizontal lineto 水平线

V = vertical lineto 垂直线

C = curveto 曲线

S = smooth curveto 光滑曲线

Q = quadratic Bezier curve 贝塞尔曲线

T = smooth quadratic Bezier curveto 光滑贝塞尔曲线

A = elliptical Arc 椭圆

Z = closepath 结束路径

注意： 上面的命令大写表示绝对定位，小写表示相对定位

□ 文字

```
<svg>
```

```
<text x="10" y="20" fill="red" transform="rotate(30,20,40)"
```

```
</svg>
```

其超链接、tspan textPath等标签的使用不再详细的讲解

详情: <http://www.runoob.com/svg/svg-text.html>

这里会发现svg标签多并且path路径等又很难计算等 推荐使用svg编辑器来生成路径 这里我们以Method Draw编辑器为例

使用教程网址: <https://blog.csdn.net/q1056843325/article/details/54563750>

英文在线工具网址: <https://editor.method.ac/>

中文在线网址: <https://c.runoob.com/more/svgeditor/>

相关链接:

<https://blog.csdn.net/u012468376/article/details/73350998>

<https://www.w3cplus.com/canvas/gradient.html>

<http://www.runoob.com/svg/svg-tutorial.html>