Snapsvg 组件使用：

**查找元素：**

Elm.select(“#id”)  通过 id 查找元素

**查找所有元素**

f.selectAll(“#id tag”) 查找 id 为 id 下的所有 tag 标签元素

**元素属性**

Elm.attr(“attribute”)  获取元素属性

element.attr({ opacity: value, rotate: (value \* 360) });

**绑定点击事件**

Elem.click(function(){})

**添加拖动事件**

Elem.drag(onmove, onstart, onend, [mcontext],[scontext],[econtext])

正在拖动方法、拖动开始方法、拖动结束方法、正在拖动 context 、开始拖动 context 、结束拖动 context

**变换元素**

Elem.transform( str) 变换的字符串

eg: transform: ‘s2r45,250,250’ ( 同时缩放水平和垂直的所标需要指定中心点 )

t 平移坐标 x,y

s 放大倍数

r 旋转角度增加到多少度

? 中心点坐标

eg: transform: ‘t100,-100s2,3’

s 水平和垂直缩放

t 移动

**元素编组**

var r = s.rect(100,100,100,100,20,20).attr({ stroke: '#123456’, 'strokeWidth’: 20, fill: ‘red’ });

var c = s.circle(50,50,50).attr({ stroke: '#123456’, 'strokeWidth’: 20, fill: ‘blue’ });

var g = s.group(r,c);

**组追加元素**

var r0 = s.rect(150,150,100,100,20,20).attr({ fill: "orange", opacity: "0.8", stroke: "black", strokeWidth: “2” });

var c0 = s.circle(225,225,10).attr({ fill: "silver", stroke: “black” });

var g0 = s.group( r0,c0 );

var r1 = s.rect(100,100,100,100,20,20).attr({ fill: "red", opacity: "0.8", stroke: "black", strokeWidth: “2” });

var c1 = s.circle(175,175,10).attr({ fill: "silver", stroke: “black” });

var g1 = s.group( r1,c1 );

g1.appendTo( g0 );

**动画效果**

Elem.animate(attr ， duration ， easing ， callback) 属性，持续时间，释放，回调

                                持续时间，弹性释放

g.animate({ transform: ‘s1r0,150,150’ }, 1000,     mina.bounce )

           起始  结束    改变值的方法                                持续时间

Snap.animate(0,    100,   function (value) {t.attr({text: Math.round(value)}); }, 1000);

**创建path**

Elem.path(pathstring) 元素使用 pathstring 创建 path

**检查点是否在path路径中（闭合？）**

Snap.path.isPointInside( myPathString, x,y )

**创建椭圆**

elem.ellipse(x ， y ， rx ， ry) 中心点 x ，中心点 y ，横向半径，纵向半径

**svg path用法**

M = moveto(M X,Y) ：将画笔移动到指定的坐标位置

L = lineto(L X,Y) ：画直线到指定的坐标位置

H = horizontal lineto(H X) ：画水平线到指定的 X 坐标位置

V = vertical lineto(V Y) ：画垂直线到指定的 Y 坐标位置

C = curveto(C X1,Y1,X2,Y2,ENDX,ENDY) ：三次贝赛曲线

S = smooth curveto(S X2,Y2,ENDX,ENDY)

Q = quadratic Belzier curve(Q X,Y,ENDX,ENDY) ：二次贝赛曲线

T = smooth quadratic Belzier curveto(T ENDX,ENDY) ：映射

A = elliptical Arc(A RX,RY,XROTATION,FLAG1,FLAG2,X,Y) ：弧线

Z = closepath() ：关闭路径

**A指令**

允许不闭合。可以想像成是椭圆的某一段，共七个参数：

A RX,RY,XROTATION,FLAG1,FLAG2,X,Y

RX,RY 指所在椭圆的半轴大小  
XROTATION 指椭圆的 X 轴与水平方向顺时针方向夹角，可以想像成一个水平的椭圆绕中心点顺时针旋转 XROTATION 的角度。  
FLAG1 只有两个值， 1 表示大角度弧线， 0 为小角度弧线。  
FLAG2 只有两个值，确定从起点至终点的方向， 1 为顺时针， 0 为逆时针  
X,Y 为终点坐标

**创建矩阵对象**

var myMatrix = new Snap.Matrix();

myMatrix.scale(4,2);                 // 横纵缩放

myMatrix.translate(100,0);          // 平移

myMatrix.rotate(45);              // 旋转