绘画板 01——实现绘制矩形功能

作者：crick77
原文链接：https://ld246.com/article/1474350442966
来源网站：链滴

svgjs 本身提供了绘制图形的函数。

为了实现鼠标点击、绘制的功能，需要监听鼠标事件

1. mousedown 修改绘画状态位，开始画图
2. mousemove 修改图形大小，绘制图形
3. mouseup 修改绘画状态位，结束画图

拓展了一个函数，获取鼠标点击时相对画布左上角的相对坐标

根据svgjs提供的函数，将图形的透明度设置为0.0全透明，可以修改边框和高度，后期可以根据上面按钮关联修改。

现在默认绘制矩形，需要向上封装，实现按钮与图形绘制的绑定。

github地址： https://github.com/wangyuheng/painter

DEMO地址： http://painter.crick.wang/

扩展Doc，获取相对坐标

```
(function() {
    SVG.extend(SVG.Doc, {
            transformPoint : function( event ){
                event = event || window.event;
                var touches = event.changedTouches && event.changedTouches[0] || event;
                var node = this.node;
                var point = node.createSVGPoint();
                point.x = touches.pageX-window.scrollX;
                point.y = touches.pageY-window.scrollY;
                var matrix = node.getScreenCTM().inverse();
                return point.matrixTransform( matrix );
            }
        });
})();
```

通过事件监听，绘制矩形

```
if (!SVG.supported) {
    alert('SVG not supported');
}
var svgDraw = SVG( "svgPanel" ).size('100%', 600);
var svgDoc = svgDraw.doc();
var drawing = false;
var element = null;
var startPoint = null;
var parent = null;
function mousemove(event) {
    console.log('rect mousemove');
    if (drawing) {
        var svgPoint = svgDoc.transformPoint(event);
```

```
        var x = svgPoint.x;
        var y = svgPoint.y;
        var newWidth = x - startPoint.x;
        var newHeight = y - startPoint.y;
        var startX = startPoint.x;
        var startY = startPoint.y;
        if (newWidth < 0) {
            startX += newWidth;
        }
        if (newHeight < 0) {
            startY += newHeight;
        }
        newWidth = Math.abs(newWidth);
        newHeight = Math.abs(newHeight);
        element.x(startX).y(startY).width(newWidth).height(newHeight);
    }
};
function mousedown(event) {
    console.log('rect mousedown');
    drawing = true;
    startPoint = svgDoc.transformPoint(event);
    element = parent.rect(0, 0).style("fill-opacity", '0.0').stroke({
        width: '2',
        color: '#000000'
    });
}
function mouseup(event) {
    console.log('rect mouseup ' + element);
    drawing = false;
}
var listener = {
    mousedown: mousedown,
    mousemove: mousemove,
    mouseup: mouseup,
};
var Tool = {};
Tool.Rect = function (parentEle) {
    parent = parentEle;
    var l =listener;
    svgDoc.on( 'mousedown', l.mousedown );
    svgDoc.on( 'mousemove', l.mousemove );
    svgDoc.on( 'mouseup',   l.mouseup );
};
Tool.Rect(svgDraw);
```