Canvas-04-Konva

作者：wei0925
时间：2017/07/04 22:20

#canvas-04-Konva ##4.1 基本概念

- 舞台 Stage
- 层 Layer
- 组 Gourp

##4.2 基本形状 --查看API: https://konvajs.github.io/api/Konva.html

###1、矩形 Rect ###2、圆形 Circle ###3、扇形 Wedge ###4、文字 Text ###5、图片 Image ###6、线 Line ###7、多边形 Star ###8、..... ###9、例子：基本形状 1.html <head> <meta charset="UTF-8"> <title>konva</title> <style> body{ margin:0; } </style> </head> <body>

```
<script src="./konva.min.js"></script>

<script>
        // 创建舞台
        var stage = new Konva.Stage({
                width:window.innerWidth,
                height:window.innerHeight,
                container:"box"
        });

        // 创建层
        var layer = new Konva.Layer();

        // 将层添加到舞台中
        stage.add(layer);

        // 创建矩形
        var rect = new Konva.Rect({
                x:100,
                y:100,
                width:200,
                height:100,
                fill:"red",

                //圆角
                cornerRadius:10,

                stroke:"green",
                strokeWidth:10,
```

```
        // 是否可拖拽
        draggable:true, //true可拖拽的,false不可拖拽

        // 缩放
        scaleX:.8,              //水平缩放
        scaleY:.5               //垂直缩放
});
layer.add(rect);

// 绘制圆形
var circle = new Konva.Circle({
        //获取舞台的中心点
        x:stage.getWidth() / 2,
        y:stage.getHeight() / 2,

        //半径
        radius:100,

        fill:"yellow",
        stroke:"red"
});
layer.add(circle);

// 绘制扇形
var shanxing = new Konva.Wedge({
        x:500,
        y:500,
        radius:100,
        fill:"orange",
        stroke:"green",
        strokeWidth:10,

        // 角度
        angle:100
});
layer.add(shanxing);

// 绘制线
var line = new Konva.Line({
        points:[200,200,400,400],       //[x1,y1,x2,y2]
        strokeWidth:10,
        stroke:"#369",

        //虚线
        // dash:[50,10] //长50隔10，然后一直循环
        dash:[50,10,30,5]       //长50隔10，长30隔5，然后一直循环
});
layer.add(line);

// 绘制五角星
var star = new Konva.Star({
        x:700,
        y:200,
```

```
                // 内圆
                innerRadius:80,

                // 外圆
                outerRadius:160,

                fill:"red",

                // 几个角
                numPoints:2             //默认数字5
            });
            layer.add(star);

            // 绘制文字
            var text = new Konva.Text({
                x:50,
                y:500,
                text:"Hello word",
                stroke:"green",
                fontSize:49
            });
            layer.add(text);

            layer.draw();   //画层
        </script>
    </body>
```

##4.3 事件 ###1、例子：鼠标事件 2.html <head> <meta charset="UTF-8"> <title>鼠标事件</title> <style> body{ margin:0; } </style> </head> <body> <script src="./konva.min.js"></script> <script> // 创建舞台 var stage = new Konva.Stage({ container:"box", width:window.innerWidth, height:window.innerHeight });

```
            // 创建层
            var layer = new Konva.Layer()
            stage.add(layer);

            // 创建组
            var group = new Konva.Group();
            layer.add(group);

            // 绘制矩形
            var rect = new Konva.Rect({
                x:200,
                y:200,

                // 偏移，此时的坐标是（x-offsetX,y-offsetY）
                offsetX:100,
                offsetY:100,

                width:200,
                height:200,
```

```
                        fill:"yellow",
                        stroke:"green",
                        strokeWidth:8
                })
                group.add(rect);

                layer.draw();

                // 给矩形设置鼠标划入划出事件
                rect.on("mouseenter",function(){
                        this.rotation(45);      //鼠标滑入让矩形旋转45度
                        layer.draw();   //改变参数需要重新绘制层
                }).on("mouseleave",function(){
                        this.rotation(0);
                        layer.draw();
                })
        </script>
</body>
```

###2、Tween 过渡

###3、To动画

###4、例子：动画效果-过渡 3.html <head> <meta charset="UTF-8"> <title>tween 过渡</title> <style> body{ margin:0; } </style> </head> <body> <script src="./konva.min.js"></script>

```
        <script>
                // 创建舞台
                var stage = new Konva.Stage({
                        container:"box",
                        width:window.innerWidth,
                        height:window.innerHeight
                });

                // 创建层
                var layer = new Konva.Layer();
                stage.add(layer);

                // 绘制矩形
                var rect = new Konva.Rect({
                        x:100,
                        y:100,
                        width:10,
                        height:20,
                        fill:"red"
                });
                layer.add(rect);

                // 绘制星星
                var star = new Konva.Star({
```

```
                x:200,
                y:300,
                innerRadius:80,
                outerRadius:120,
                nomPoints:11,
                fill:"red"
        });
        layer.add(star);

        layer.draw()

        // 创建tween动画
        var tween = new Konva.Tween({
                node:rect,
                width:600,
                duration:3,
                yoyo:true,
                easing:Konva.Easings.EaseOut,   //先快后慢，先慢后快

                // 动画执行完毕
                onFinish:function(){
                        console.log("啊，结束了");
                }
        });
        tween.play();

        // to动画 tween的简单用法
        star.to({
                rotation:360,   //旋转
                duration:1,             //过渡时间
                yoyo:true               //是否执行循环
        })
    </script>
</body>
```

###5、例子：动画效果-animation、定时 4.html <head> <meta charset="UTF-8"> <title>tween（动画、定时）</title> <style> body{ margin:0; } </style> </head> <body>

```
    <script src="./konva.min.js"></script>
    <script>
        // 创建舞台
        var stage = new Konva.Stage({
                container:"box",
                width:window.innerWidth,
                height:window.innerHeight
        });

        // 创建层
        var layer = new Konva.Layer();
        stage.add(layer);
```

```
                // 绘制星星
                var star = new Konva.Star({
                        x:200,
                        y:300,
                        innerRadius:80,
                        outerRadius:120,
                        numPoints:11,
                        fill:"red"
                })
                layer.add(star);

                layer.draw();

                var angle = 10;
                var animate = new Konva.Animation(function(frame){
                        star.rotation(angle);
                        angle += 10;
                },layer);
                animate.start();

                // 定时
                // var angle = 10;
                // setInterval(function(){
                //      star.rotation(angle);
                //      angle += 10;
                // },layer);
                // animate.start();
        </script>
</body>
```

###6、例子：进度条(animation) 5.html <body> <script src="./konva.min.js"></script> <script> // 创建舞台 var stage = new Konva.Stage({ container:"box", width:window.innerWidth, height:window.innerHeight });

```
                // 创建层
                var layer = new Konva.Layer();
                stage.add(layer);

                // 外面的矩形 边框
                var outerRect = new Konva.Rect({
                        x:100,
                        y:100,
                        width:800,
                        height:60,
                        stroke:"green",
                        strokeWidth:6
                })
                layer.add(outerRect);

                // 里面的矩形
                var innerRect = new Konva.Rect({
                        x:100,
```

```
                y:100,
                width:0,
                height:60,
                fill:"pink"
        })
        layer.add(innerRect);

        layer.draw();

        var animate = new Konva.Animation(function(){
                if(innerRect.width() < 800){
                        // 宽度重复加10
                        innerRect.width(innerRect.width() + 10)
                }else{
                        animate.stop();
                }
        },layer);

        animate.start();
    </script>
</body>
```

###7、例子：钟表 6.html <head> <meta charset="UTF-8"> <title>钟表</title> <style> body{ margin: 0; } </style> </head> <body> <script src="./konva.min.js"> </script> <script> // 定义宽高 var w = window.innerWidth, h = window.innerHeight;

```
        // 圆心坐标
        var x = w/2, y = h/2;

        // 表盘半径
        var radius = 200;

        // 创建舞台
        var stage = new Konva.Stage({
                container:"box",
                width:w,
                height:h
        })

        // 创建表盘的层
        var dialLayer = new Konva.Layer();
        stage.add(dialLayer);

        // 绘制表盘
        var circle = new Konva.Circle({
                x:x,
                y:y,
                radius:radius,
                strokeWidth:10,
                stroke:"#000"
        });
        dialLayer.add(circle);
```

```
// 绘制刻度
var hourDial = new dialSacle({
        x:x,
        y:y,
        outerRadius:radius,
        innerRadius:radius-20,
        strokeWidth:10,
        number:12
});
dialLayer.add(hourDial);

//绘制分钟刻度
var minuteDial = new dialSacle({
        x:x,
        y:y,
        outerRadius:radius,
        innerRadius:radius-10,
        strokeWidth:4,
        number:60
});
dialLayer.add(minuteDial);

/*
* 构造函数  绘制刻度
* [@param](https://my.oschina.net/u/2303379) object options  配置选项
*/
function dialSacle(options){
        options = options || {},
        options.x = options.x || 0;
        options.y = options.y || 0;
        options.outerRadius = options.outerRadius || 0;
        options.innerRadius = options.innerRadius || 0;
        options.number = options.number || 0;
        options.strokeWidth = options.strokeWidth || 0;
        options.stroke = options.stroke || "#000";

        // 创建组
        var group = new Konva.Group({
                x:options.x,
                y:options.y
        });

        // 绘制刻度

        // 刻度间隔
        var angleDiff = 360 / options.number / 180 * Math.PI;
        for(var i = 0;i < options.number;i++){
                var line = new Konva.Line({
                        points:[
                                Math.cos(angleDiff * i) * options.outer
                                Math.sin(angleDiff * i) * options.outer
                                Math.cos(angleDiff * i) * options.inner
                                Math.sin(angleDiff * i) * options.inner
                        ],
```

```
                        strokeWidth:options.strokeWidth,
                        stroke:options.stroke
                });
                group.add(line)
        }
        return group;
}
dialLayer.draw();

// 创建指针层
var handLayer = new Konva.Layer({
        x:x,
        y:y
});
stage.add(handLayer);

// 秒针
var secondHand = new Konva.Line({
        points:[-20,0,180,0],
        stroke:"red",
        strokeWidth:2
});
handLayer.add(secondHand);

//分针
var minuteHand = new Konva.Line({
        points:[-20,0,150,0],
        stroke:"#000",
        strokeWidth:6
});
handLayer.add(minuteHand);

//时针
var hourHand = new Konva.Line({
        points:[-20,0,120, 0],
        stroke:"#000",
        strokeWidth:10
});
handLayer.add(hourHand);

// 修饰的小圆
var smallCircle = new Konva.Circle({
        x:0,
        y:y,
        radius:10,
        fill:"#000"
});
handLayer.add(smallCircle);

// 定时
function run(){
        //获取当前时间
        var date = new Date();

        //秒
```

```
                var seconds = date.getSeconds();
                secondHand.rotation(seconds/60 * 360 - 90);

                //分针
                var minutes = date.getMinutes() + seconds / 60;
                minuteHand.rotation(minutes/60 * 360 - 90);

                //小时
                var hours = date.getHours() % 12 + minutes / 60;
                hourHand.rotation(hours/12 *360 - 90);

                handLayer.draw(); //重新绘制

                setTimeout(run, 1000);
            }

            run();

            handLayer.draw();
        </script>
    </body>
```

##4.4 Echarts表格插件 * 参考Echarts官网：http://echarts.baidu.com/index.html

##作业 ###1、炫酷效果 查看homework-answer ###2、动画的饼状图 查看homework-answer ###3、动画的直方图 查看homework-answer

---