HTML5 Canvas

1 默认画布大小

300px \* 150px

2 设置画布大小

<canvas width="600px" height="30px">

3 缩放画布大小

<style>canvas { width: 600px; height: 300px; }</style>

4 绘制方法

取得绘制上下文对象 var ctx = document.getElementById('myCanvas').getContext('2d');

设置画布宽度 document.getElementById('myCanvas').width = x;

设置画布高度 document.getElementById('myCanvas').height = y;

重置路径（否则重叠） ctx.beginPath();

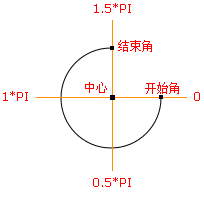
闭合路径 ctx.close();

设置线段起点 ctx.moveTo(x, y);

设置线段终点 ctx.lineTo(x, y);

设置线段宽度 ctx.lineWidth = x;

绘制圆形 ctx.arc(x, y, r, 起始角弧度, 结束角弧度, 绘制时针方向); 正圆弧度：0, 2 \* Math.PI

设置描边样式 ctx.strokeStyle = "#000";

描边 ctx.stroke();

设置填充样式 ctx.fillStyle = "rgba(0, 0, 0, 1)"

填充 ctx.fill();

描边矩形 ctx.strokeRect(x, y, w, h);

填充矩形 ctx.fillRect(x, y, w, h);

描边圆形 ctx.arc(); ctx.stroke();

填充圆形 ctx.arc(); ctx.fill();

平移原点 ctx.translate(x, y);

旋转 ctx.rotate(旋转角弧度);

缩放 ctx.scale(x, y);

保存状态 ctx.save();

恢复状态 ctx.restore();

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

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

设置渐变颜色 gradient.addColorStop(位置百分比（0~1之间小数）, "rgb(0, 0, 0)");

设置渐变填充 ctx.fillStyle = gradient;

填充文字 ctx.fillText(str, x, y);

描边文字 ctx.strokeText(str, x, y);

设置填充文字样式 ctx.font = "50px sans-serif";

填充文字水平居中 ctx.textAlign = "left | center | right";

填充文字垂直居中 ctx.textBaseline = "top | middle | bottom";

获取文本宽度 ctx.measureText(str).width;

填充图片 var img = new Image();

img.src = "logo.png";

img.onload = function() { ctx.drawImage(new Image(), x, y); } 图像加载回调异步处理

缩放图片 ctx.drawImage(new Image(), x, y, w, h);

裁剪图片 ctx.drawImage(new Image(), x1, y1, w1, h1, x2, y2, w2, h2);

图形画刷 var pattern = ctx.createPattern(img, "repeat | norepeat | repeat-x | repeat-y");

ctx.fillStyle = pattern;

区域剪辑 ctx.clip();

绘制阴影 ctx.shadowOffsetX = number;

ctx.shadowOffsetY = number;

ctx.shadowColor = "rgba(0, 0, 0, 0.5)";

ctx.shadowBlur = number;

清除矩形区域 ctx.clearRect(0, 0, canvas.width, canvas.height);

绘制贝塞尔曲线 ctx.moveTo(x1, y1); ctx.quadraticCurveTo(x2, y2, x3, y3);

可使用canvas二次贝塞尔曲线工具生成代码 http://web.chacuo.net/css3beziertool

离屏动画源码（使用ctx.drawImage(offCanvas, ...)）

<body>

<canvas id="myCanvas" width="600px" height="400px">

您的浏览器不支持canvas

</canvas>

<!--创建离屏Canvas-->

<canvas id="offCanvas" width="600px" height="400px">

您的浏览器不支持canvas

</canvas>

<script type="text/javascript">

var canvas = document.getElementById('myCanvas');

var ctx = canvas.getContext('2d');

var offCanvas = document.getElementById('offCanvas');

var offCtx = offCanvas.getContext('2d');

var posx = 0, posy = 0, dir = 1, isMouseInRect = false;

//把一些复杂的绘画操作，画在离屏Canvas上面

var drawALot = function(){

for(var k=0; k<20; k++){

for(var i=0;i<canvas.width;i+=10){

for(var j=0;j<canvas.height;j+=10){

offCtx.beginPath();

offCtx.arc(i,j,5,0,2\*Math.PI,true);

offCtx.stroke();

}

}

}

}

canvas.onmousemove = function(e){

var mouseX = e.offsetX;

var mouseY = e.offsetY;

if(mouseX > posx && mouseY <posx +50 && mouseY > posy && mouseY < posy + 50){

isMouseInRect = true;

}else{

isMouseInRect = false;

}

}

setInterval(function() {

if(!isMouseInRect){

posx += 10 \* dir;

}

//clearRect清空画布的一个矩形区域

ctx.clearRect(0, 0, canvas.width, canvas.height);

// drawALot();

//真正要用到复杂的绘画的时候，直接从离屏Canvas上拷贝过来

ctx.drawImage(offCanvas,0,0,offCanvas.width, offCanvas.height,0,0, canvas.width, canvas.height);

ctx.fillRect(posx, posy, 50, 50);

if(posx + 50 >= canvas.width){

dir = -1;

}else if(posx <= 0){

dir = 1;

}

},100);

drawALot();

</script>

</body>