此文章已于 21:55:59 2016/1/25 重新发布到 Net | Linux | 安全

06. Web大前端时代之：HTML5+CSS3入门系列~HTML5 画布

Web大前端时代之：HTML5+CSS3入门系列：<http://www.cnblogs.com/dunitian/p/5121725.html>

**05.HTML5 画布**

**初始画布**

**canvas默认是宽300px，高150px;**

**绘制步骤**

**1.定义一个id**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**2.获取canvas对象**

**var canvasObj = document.getElementById('canvasOne');**

**3.通过getContext获取上下文**

**var context = canvasObj.getContext("2d");**

**目前支持2d绘图**

**4.通过javascript进行绘制**

**context.fillStyle = "red";**

**设置样式为红色**

**context.fillRect(125, 125, 50, 50);**

**在x坐标为125，y坐标为125的地方绘制一个长为50宽为50的正方形**

**绘制案例**

**常见几何**

**绘制直线**

**绘制300\*300画布的对角线**

**beginPath()**

**开始绘制**

**moveTo(x,y)**

**直线起点**

**lineTo(x,y)**

**直线终点**

**stroke()**

**绘制直线**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext("2d");**

**//context.strokeStyle = "#ff6a00";**

**context.beginPath();**

**context.moveTo(0, 0);**

**context.lineTo(300, 300);**

**context.stroke();**

**context.moveTo(0, 300);**

**context.lineTo(300, 0);**

**context.stroke();**

**</script>**

**绘制矩形**

**在画布中间绘制一个100\*50的矩形**

**绘制**

**strokeStyle**

**strokeRect(x,y,w,h)**

**填充**

**fillStyle**

**fillRect(x,y,w,h)**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext("2d");**

**context.strokeStyle = 'rgb(100%,0%,100%)';**

**context.strokeRect(100, 125, 100, 50);**

**</script>**

**绘制圆形**

**绘制300\*300画布的内切圆**

**beginPath()**

**开始绘制路径**

**arc(x, y, r, s, e, b)**

**x，y 坐标这次是代表圆心**

**r 代表半径**

**s 代表开始弧度**

**e 代表结束弧度**

**b 代表是否顺时针方向画图**

**默认逆时针**

**closePath()**

**结束绘制路径**

**扩充案例**

**fill()**

**填充**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext("2d");**

**context.fillStyle = 'rgb(0,50,0)';**

**context.beginPath();//开始绘制路径**

**context.arc(150, 150, 150, 0, Math.PI \* 2,true);//注意：x，y 坐标这次是代表圆心**

**context.closePath();//结束绘制路径**

**context.fill();//填充**

**</script>**

**绘制弧线**

**context.closePath();**

**路径不闭合的时候会自动画一条直线（代码看注释）**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<canvas id="canvasTwo" width="300" height="300"></canvas>**

**<canvas id="canvasThree" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**//第一个画布**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext("2d");**

**for (var i = 0; i <= 15; i++) {**

**context.fillStyle = 'rgb(0,50,0)';**

**context.beginPath();//开始绘制路径**

**context.arc(0, 150, i \* 10, 0, Math.PI \* 1.5);//注意：x，y 坐标这次是代表圆心**

**//context.closePath();//结束绘制路径**

**context.stroke();//填充**

**}**

**//第二个画布**

**var canvasObj2 = document.getElementById('canvasTwo');**

**var context2 = canvasObj2.getContext("2d");**

**for (var i = 0; i <= 30; i++) {**

**context2.fillStyle = 'rgb(0,0,50)';**

**context2.beginPath();//开始绘制路径**

**context2.arc(0, 0, i \* 10, 0, Math.PI \* 1.5);//注意：x，y 坐标这次是代表圆心**

**//context2.closePath();//结束绘制路径**

**context2.stroke();//填充**

**}**

**//第三个画布-搞怪来袭**

**var canvasObj3 = document.getElementById('canvasThree');**

**var context3 = canvasObj3.getContext("2d");**

**for (var i = 0; i <= 30; i++) {**

**context3.fillStyle = 'rgb(0,0,50)';**

**context3.beginPath();//开始绘制路径**

**context3.arc(0, 150, i \* 10, 1, 3);//注意：x，y 坐标这次是代表圆心**

**context3.closePath();//结束绘制路径,路径不闭合的情况下会自动补上一个直线，所以就搞怪了**

**context3.stroke();//填充**

**}**

**</script>**

**绘制三角形**

**绘制三角形**

**<canvas id="myCanvas" width="200" height="200"></canvas>**

**<script type="text/javascript">**

**var c = document.getElementById("myCanvas");**

**var context = c.getContext("2d");**

**context.strokeStyle = "red";**

**context.beginPath();**

**context.moveTo(25, 25);**

**context.lineTo(150, 25);**

**context.lineTo(25, 150);**

**context.closePath();**

**context.stroke();**

**</script>**

**常用技能**

**透 明 度**

**rgba(r,g,b,a)**

**a代表透明度，取值范围在 0~1**

**清 除 画 布**

**context.clearRect(x,y,w,h)**

**（圆形橡皮擦）**

**案例**

**保存与恢复状态**

**状态**

**指当前画面所有样式，变形，裁切的快照**

**举个例子：你先用红色样式画一个矩形，然后保存状态，然后再用蓝色样式画一个矩形。**

**恢复状态画个矩形看看，发现====> 矩形是红色的====>说明====>状态保存下来了。**

**图例**

**保存**

**context.save();**

**恢复**

**context.restore()**

**移动坐标空间**

**context.translate(60,50);**

**x轴方向右移60，y轴方向下移50**

**for (var i = 1; i < 10; i++) {**

**context.save();//保存一下状态**

**context.translate(60 \* i, 50);//x轴方向右移60\*i，y轴方向下移50**

**//画伞的顶部**

**context.fillStyle = 'rgb(' + i \* 25 + ',' + 0 + ',' + 255 + ')';**

**context.beginPath();**

**context.arc(0, 0, 30, 0, Math.PI, true);//在起点位置顺时针画一个半圆**

**context.closePath();**

**context.fill();**

**//画伞的底部**

**context.strokeStyle = "red";**

**context.strokeRect(-0.2, 0, 0.4, 30);**

**//画伞的根部**

**context.beginPath();**

**context.arc(-5, 30, 5, 0, Math.PI);//圆心左移五个单位，圆的右边点就在根部了**

**context.stroke();**

**context.restore();//恢复一下状态（不然每次都移动坐标原点，就变成天女散花了）**

**}**

**for (var i = 1; i < 10; i++) {**

**//x轴方向右移60+i，y轴方向下移50**

**context.translate(60 + i, 50);**

**//画伞的顶部**

**context.fillStyle = 'rgb(' + 255 + ',' + i \* 25 + ',' + 0 + ')';**

**context.beginPath();**

**context.arc(0, 0, 30, 0, Math.PI, true);//在起点位置顺时针画一个半圆**

**context.closePath();**

**context.fill();**

**//画伞的底部**

**context.strokeStyle = "red";**

**context.strokeRect(-0.2, 0, 0.4, 30);**

**//画伞的根部**

**context.beginPath();**

**context.arc(-5, 30, 5, 0, Math.PI);//圆心左移五个单位，圆的右边点就在根部了**

**context.stroke();**

**}**

**旋转坐标空间**

**rotate(angle)**

**angle 代表旋转角度**

**弧度为单位**

**在坐标原点顺时针方向旋转**

**<canvas id="canvasOne" width="600" height="600"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**context.translate(300, 300);**

**//画伞**

**function drawUmbrella(i) {**

**//画伞的顶部**

**context.fillStyle = 'rgb(' + i \* 25 + ',' + 0 + ',' + 255 + ')';**

**context.beginPath();**

**context.arc(0, 0, 30, 0, Math.PI, true);//在起点位置顺时针画一个半圆**

**context.closePath();**

**context.fill();**

**//画伞的底部**

**context.strokeStyle = "red";**

**context.strokeRect(-0.2, 0, 0.4, 30);**

**//画伞的根部**

**context.beginPath();**

**context.arc(-5, 30, 5, 0, Math.PI);//圆心左移五个单位，圆的右边点就在根部了**

**context.stroke();**

**}**

**function draw() {**

**for (var i = 1; i <= 10; i++) {**

**context.save();//保存一下状态**

**context.rotate(Math.PI \* (0.2 \* i));//2Pi 弧度是一个圆**

**context.translate(0, 150);//越小越紧凑**

**drawUmbrella(i);//画伞（画伞代码未变）**

**context.restore();//恢复一下状态**

**}**

**}**

**window.onload = function () {**

**draw();**

**}**

**</script>**

**缩 放 图 形**

**context.scale(1.1, 1.1)**

**扩大1.1倍**

**rgba(r,g,b,a)**

**a代表透明度，取值范围在 0~1**

**<canvas id="canvasOne" width="600" height="600"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**context.translate(300, 300);**

**for (var i = 1; i < 50; i++) {**

**context.rotate(Math.PI / 9);//旋转**

**context.scale(1.1, 1.1);//扩大1.1倍**

**context.translate(0.5, 0.5);//平移**

**context.fillStyle = 'rgba(' + i \* 5 + ',' + 0 + ',' + 200 + ',' + 0.5 + ')';**

**context.beginPath();**

**context.arc(0, 0, 1, 0, Math.PI \* 2, true);**

**context.closePath();**

**context.fill();**

**}**

**</script>**

**矩 阵 变 化**

**其实像 translate（移动），scale（缩放），rotate（旋转）都是特殊的矩阵变换**

**transform(m11,m12,m21,m22,dx,dy)**

**替换当前的变换矩阵（transform() 允许您缩放、旋转、移动并倾斜当前的环境）**

**http://www.w3school.com.cn/tags/canvas\_transform.asp**

**参数图解**

**本质公式**

**参数详解**

**水平缩放绘图**

**m11**

**水平倾斜绘图**

**m12**

**垂直倾斜绘图**

**m21**

**垂直缩放绘图**

**m22**

**水平移动绘图**

**dx**

**垂直移动绘图**

**dy**

**setTransform(m11,m12,m21,m22,dx,dy)**

**重置并创建新的变换矩阵**

**http://www.w3school.com.cn/tags/canvas\_settransform.asp**

**小案例**

**<canvas id="canvasOne" width="600" height="600"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**context.translate(200, 20);**

**for (var i = 1; i < 90; i++) {**

**context.save();**

**//参数：水平缩放绘图，水平倾斜绘图，垂直倾斜绘图，垂直缩放绘图，水平移动绘图，垂直移动绘图**

**context.transform(0.95, 0, 0, 0.95, 30, 30);**

**context.rotate(Math.PI / 12);//旋转角度**

**context.beginPath();**

**context.fillStyle = "rgba(255,0,0,0.5)";**

**context.arc(0, 0, 50, 0, Math.PI \* 2, true);**

**context.closePath();**

**context.fill();**

**}**

**//参数：水平缩放绘图，水平倾斜绘图，垂直倾斜绘图，垂直缩放绘图，水平移动绘图，垂直移动绘图**

**context.setTransform(1, 0, 0, 1, 10, 10);**

**//检验一下是否变化过来了**

**context.fillStyle = "blue";**

**context.fillRect(0, 0, 50, 50);**

**context.fill();**

**</script>**

**扩 展 样 式**

**线形属性**

**lineWidth**

**设置线条粗细默认为1，为正数**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**for (var i = 1; i < 12; i++) {**

**context.strokeStyle = 'rgb(255,0,0)';**

**context.lineWidth = i;**

**context.beginPath();**

**context.moveTo(i \* 20, 0);**

**context.lineTo(i \* 20, 300);**

**//context.closePath();**

**context.stroke();**

**}**

**</script>**

**lineCap**

**设置端点样式**

**butt**

**平头**

**默认**

**round**

**圆头**

**square**

**方头**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**//定义数组**

**var lineCap = ['butt', 'round', 'square'];**

**// 绘制参考线。**

**context.strokeStyle = 'red';**

**context.beginPath();**

**context.moveTo(10,10);**

**context.lineTo(10,150);**

**context.moveTo(150,10);**

**context.lineTo(150,150);**

**context.stroke();**

**// 绘制直线段。**

**context.strokeStyle = 'blue';**

**for (var i = 0; i < lineCap.length; i++) {**

**context.lineWidth = 20;**

**context.lineCap = lineCap[i];**

**context.beginPath();**

**context.moveTo(10, 30 + i \* 50);**

**context.lineTo(150, 30 + i \* 50);**

**context.stroke();**

**}**

**</script>**

**lineJoin**

**设置连接处样式**

**round**

**圆**

**bevel**

**斜面**

**miter**

**默认**

**<canvas id="canvasOne" width="500" height="200"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**var lineJoin = ['round', 'bevel', 'miter'];**

**context.strokeStyle = 'rgb(0,0,0)';**

**for (var i = 0; i < lineJoin.length; i++) {**

**context.lineWidth = 25;**

**context.lineJoin = lineJoin[i];**

**context.beginPath();**

**context.moveTo(10 + i \* 150, 30);**

**context.lineTo(100 + i \* 150, 30);**

**context.lineTo(100 + i \* 150, 100);**

**context.stroke();**

**}**

**</script>**

**miterLimit**

**设置或返回最大斜接长度**

**前提**

**lineJoin使用默认属性（miter）**

**<canvas id="canvasOne" width="1600" height="300"></canvas>**

**<script type="text/javascript">**

**var context = document.getElementById('canvasOne').getContext('2d');**

**for (var i = 1; i < 10; i++) {**

**context.strokeStyle = 'blue';**

**context.lineWidth = 10;**

**context.lineJoin = 'miter';**

**context.miterLimit = i \* 10;**

**context.beginPath();**

**context.moveTo(10, i \* 30);**

**context.lineTo(100, i \* 30);**

**context.lineTo(10, 33 \* i);**

**context.stroke();**

**}**

**</script>**

**渐 变 系 列**

**线形渐变**

**createLinearGradient(x0, y0, x1, y1)**

**请使用该对象作为 strokeStyle 或 fillStyle 属性的值**

**x0,y0**

**渐变起点**

**x1,y1**

**渐变终点**

**addColorStop(position, color);**

**一般都是设置多个色标**

**position**

**色相偏移值**

**取值 0~1**

**color**

**颜色**

**并非一定从0开始，1结束**

**<canvas id="canvasOne" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**var context = document.getElementById('canvasOne').getContext('2d');**

**var lingrad = context.createLinearGradient(0, 0, 0, 200);**

**lingrad.addColorStop(0, '#ff0000');**

**lingrad.addColorStop(1 / 7, '#ff9900');**

**lingrad.addColorStop(2 / 7, '#ffff00');**

**lingrad.addColorStop(3 / 7, '#00ff00');**

**lingrad.addColorStop(4 / 7, '#00ffff');**

**lingrad.addColorStop(5 / 7, '#0000ff');**

**lingrad.addColorStop(6 / 7, '#ff00ff');**

**lingrad.addColorStop(1, '#ff0000');**

**context.fillStyle = lingrad;**

**context.fillRect(10, 10, 200, 200);**

**</script>**

**var c = document.getElementById('myCanvas');**

**var ctx = c.getContext('2d');**

**var grd = ctx.createLinearGradient(0, 0, 170, 0);**

**grd.addColorStop(0.1, 'rgb(255,255,255)');**

**grd.addColorStop(0.5, 'rgb(0,100,0)');**

**grd.addColorStop(0.9, 'rgb(0,0,0)');**

**ctx.fillStyle = grd;**

**ctx.fillRect(20, 20, 150, 100);**

**径向渐变**

**createRadialGradient(x1,y1,r1,x2,y2,r2)**

**以（x1,y1）为原点，r1为半径的圆**

**以（x2,y2）为原点，r2为半径的圆**

**addColorStop(position, color);**

**一般都是设置多个色标**

**position**

**色相偏移值**

**取值 0~1**

**color**

**颜色**

**并非一定从0开始，1结束**

**<canvas id="myCanvas" width="300" height="240"></canvas>**

**<script language="javascript">**

**var ctx = document.getElementById('myCanvas').getContext('2d');**

**var radgrad = ctx.createRadialGradient(55, 55, 20, 100, 100, 90);**

**radgrad.addColorStop(0, 'rgb(255,255,0)');**

**radgrad.addColorStop(0.75, 'rgb(255,0,0)');**

**radgrad.addColorStop(1, 'rgb(255,255,255)');**

**ctx.fillStyle = radgrad;**

**ctx.fillRect(10, 10, 200, 200);**

**</script>**

**创 建 阴 影**

**参数**

**shadowOffsetX**

**shadowOffsetX 属性设置或返回形状与阴影的水平距离**

**shadowOffsetX=0 指示阴影位于形状的正下方。**

**shadowOffsetX=20 指示阴影位于形状 left 位置右侧的 20 像素处。**

**shadowOffsetX=-20 指示阴影位于形状 left 位置左侧的 20 像素处。**

**shadowOffsetY**

**shadowOffsetY 属性设置或返回形状与阴影的垂直距离。**

**shadowOffsetY=0 指示阴影位于形状的正下方。**

**shadowOffsetY=20 指示阴影位于形状 top 位置下方的 20 像素处。**

**shadowOffsetY=-20 指示阴影位于形状 top 位置上方的 20 像素处。**

**shadowBlur**

**shadowBlur 属性设置或返回阴影的模糊级数。**

**shadowColor**

**shadowColor 属性设置或返回用于阴影的颜色**

**<canvas id="canvasOne" width="600" height="600"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**//设置前画个图**

**context.fillStyle = '#0094ff';**

**context.fillRect(410, 310, 50, 50);**

**//设置阴影**

**context.shadowOffsetX = 3;**

**context.shadowOffsetY = 3;**

**context.shadowBlur = 1;**

**context.shadowColor = '#808080';**

**//画矩形**

**context.fillRect(200, 200, 200, 100);**

**//绘图**

**var img = new Image();**

**img.src = '/images/1.jpg';**

**img.onload = function () {**

**context.fillStyle = context.createPattern(img, 'no-repeat');**

**context.fillRect(0, 0, 600, 600);**

**}**

**</script>**

**绘制文字**

**绘制填充文字**

**fillText（str,x,y,[mw]）**

**str**

**文字内容**

**x,y**

**起点坐标**

**mw**

**最大宽度**

**可选参数**

**绘制文字轮廓**

**strokeText（str,x,y,[mw]）**

**str**

**文字内容**

**x,y**

**起点坐标**

**mw**

**最大宽度**

**可选参数**

**测量文字宽度**

**measureText(str)**

**context.measureText(str).width**

**文字宽度**

**str**

**文字内容**

**文字系列属性**

**context.font**

**语法**

**context.font="italic small-caps bold 12px arial";**

**font-style**

**规定字体样式。可能的值：**

**normal**

**italic**

**斜体字**

**oblique**

**倾斜**

**font-variant**

**规定字体变体。可能的值：**

**normal**

**small-caps**

**大写**

**font-weight**

**规定字体的粗细。可能的值：**

**normal**

**bold**

**bolder**

**lighter**

**100**

**font-family**

**规定字体系列。**

**font-size / line-height**

**规定字号和行高，以像素计。**

**icon**

**使用用于标记图标的字体。**

**menu**

**使用用于菜单中的字体（下拉列表和菜单列表）。**

**caption**

**使用标题控件的字体（比如按钮、下拉列表等）。**

**status-bar**

**使用用于窗口状态栏中的字体。**

**message-box**

**使用用于对话框中的字体。**

**small-caption**

**使用用于标记小型控件的字体。**

**案例**

**<canvas id="canvasOne" width="500" height="300"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**var str = 'http://dnt.dkill.net';**

**context.fillStyle = 'rgba(255,0,0,0.9)';**

**context.strokeStyle = 'rgba(255,0,0,0.9)';**

**context.font = '30px 微软雅黑';**

**context.strokeText(str, 100, 40);**

**console.log(context.measureText(str).width);**

**context.fillText(str, 100, 80);**

**console.log(context.measureText(str).width);**

**context.font = 'bold 30px 微软雅黑';**

**context.strokeText(str, 100, 150);**

**console.log(context.measureText(str).width);**

**context.fillText(str, 100, 180);**

**console.log(context.measureText(str).width);**

**</script>**

**图像系列**

**1.图像来源**

**路径图片**

**直接对 src 赋值**

**var img=new Image();**

**img.src='xxx';**

**页面图片**

**来源于页面，如果已知id则可通过**

**document.images 集合**

**document.getElementsByTagName**

**document.getElementsById**

**其他canvas元素**

**document.getElementsByTagName**

**document.getElementsById**

**2.drawImage绘图**

**context.drawImage(img,x,y)**

**在画布上定位图像**

**img 规定要使用的图像、画布或视频。**

**x 在画布上放置图像的 x 坐标位置。**

**y 在画布上放置图像的 y 坐标位置。**

**//获取路径图片**

**document.getElementById('btn1').onclick = function () {**

**clearCanvas(context);**

**var img1 = new Image();**

**img1.src = '/images/1.jpg';**

**context.drawImage(img1, 150, 150);**

**}**

**context.drawImage(img,x,y,w,h)**

**在画布上定位图像，并规定图像的宽度和高度**

**img 规定要使用的图像、画布或视频。**

**x 在画布上放置图像的 x 坐标位置。**

**y 在画布上放置图像的 y 坐标位置。**

**w 要使用的图像的宽度。（伸展或缩小图像）**

**h 要使用的图像的高度。（伸展或缩小图像）**

**//获取页面图片**

**document.getElementById('btn2').onclick = function () {**

**clearCanvas(context);**

**var img2 = document.getElementById('imgOne');**

**context.drawImage(img2, 150, 150, 200, 200);**

**}**

**context.drawImage(img,sx,sy,sw,sh,x,y,w,h)**

**剪切图像，并在画布上定位被剪切的部分**

**img 规定要使用的图像、画布或视频。**

**sx 开始剪切的 x 坐标位置。**

**sy 开始剪切的 y 坐标位置。**

**sw 被剪切图像的宽度。**

**sh 被剪切图像的高度。**

**x 在画布上放置图像的 x 坐标位置。**

**y 在画布上放置图像的 y 坐标位置。**

**w 要使用的图像的宽度。（伸展或缩小图像）**

**h 要使用的图像的高度。（伸展或缩小图像）**

**//从Canvas获取**

**document.getElementById('btn3').onclick = function () {**

**clearCanvas(context);**

**var img3 = document.getElementById('canvasTwo');**

**context.drawImage(img3, 150, 150, 300, 250, 50, 50, 400, 300);**

**}**

**扩展（不完美）**

**createPattern（image，type）**

**image**

**规定要使用的图片、画布或视频元素**

**var img=new Image();**

**type**

**是否重发**

**repeat|repeat-x|repeat-y|no-repeat**

**<canvas id="canvasOne" width="600" height="600"></canvas>**

**<script type="text/javascript">**

**var canvasObj = document.getElementById('canvasOne');**

**var context = canvasObj.getContext('2d');**

**var img = new Image();**

**img.src = '/images/1.jpg';**

**img.onload = function () {**

**context.fillStyle = context.createPattern(img, 'repeat');**

**context.fillRect(0, 0, 600, 600);**

**}**

**</script>**

**注意：**

**context.fillRect(100, 0, 600, 600);**

**这里的fillRect（x,y,w,h）。x,y既指的是坐标原点，也指的是图片原点**

**扩展部分**

**绘制贝塞尔曲线**

**二次方贝塞尔曲线**

**quadraticCurveTo(cp1x,cp1y,x,y)**

**cp1x,cp1y是控制点坐标**

**x,y是终点坐标**

**<canvas id="myCanvas" style="border:1px solid;" width="300" height="200"></canvas>**

**<script type="text/javascript">**

**var c=document.getElementById("myCanvas");**

**var context=c.getContext("2d");**

**// 下面开始绘制二次方贝塞尔曲线。**

**context.strokeStyle="dark";**

**context.beginPath();**

**context.moveTo(0,200);**

**context.quadraticCurveTo(75,50,300,200);**

**context.stroke();**

**context.globalCompositeOperation="source-over";**

**// 下面绘制的直线用于表示上面曲线的控制点和控制线，控制点坐标即两直线的交点（75,50）。**

**context.strokeStyle="#ff00ff";**

**context.beginPath();**

**context.moveTo(75,50);**

**context.lineTo(0,200);**

**context.moveTo(75,50);**

**context.lineTo(300,200);**

**context.stroke();**

**</script>**

**三次方贝塞尔曲线**

**bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y)**

**cp1x,cp1y是控制点坐标**

**cp2x,cp2y是第二个控制点坐标**

**x,y是终点坐标**

**<canvas id="myCanvas" width="300" height="200"></canvas>**

**<script type="text/javascript">**

**var c=document.getElementById("myCanvas");**

**var context=c.getContext("2d");**

**// 下面开始绘制三次方贝塞尔曲线。**

**context.strokeStyle="dark";**

**context.beginPath();**

**context.moveTo(0,200);**

**context.bezierCurveTo(25,50,75,50,300,200);**

**context.stroke();**

**context.globalCompositeOperation="source-over";**

**// 下面绘制的直线用于表示上面曲线的控制点和控制线，控制点坐标为（25,50）和（75,50）。**

**context.strokeStyle="#ff00ff";**

**context.beginPath();**

**context.moveTo(25,50);**

**context.lineTo(0,200);**

**context.moveTo(75,50);**

**context.lineTo(300,200);**

**context.stroke();**

**</script>**

**组合裁切**

**组合**

**globalCompositeOperation**

**设置或返回如何将一个源（新的）图像绘制到目标（已有）的图像上**

**裁切**

**clip()**

**从原始画布中剪切任意形状和尺寸**

**案例**

**从画布中剪切 200\*120 像素的矩形区域。然后，绘制绿色矩形。只有被剪切区域内的绿色矩形部分是可见的**

**<p>不使用 clip()：</p>**

**<canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;"></canvas>**

**<script>**

**var c = document.getElementById("myCanvas");**

**var ctx = c.getContext("2d");**

**ctx.rect(50, 20, 200, 120);**

**ctx.stroke();**

**ctx.fillStyle = "green";**

**ctx.fillRect(0, 0, 150, 100);**

**</script>**

**<br />**

**<p>使用 clip()：</p>**

**<canvas id="myCanvas2" width="300" height="150" style="border:1px solid #d3d3d3;"></canvas>**

**<script>**

**var c = document.getElementById("myCanvas2");**

**var ctx = c.getContext("2d");**

**// Clip a rectangular area**

**ctx.rect(50, 20, 200, 120);**

**ctx.stroke();**

**ctx.clip();**

**// Draw red rectangle after clip()**

**ctx.fillStyle = "green";**

**ctx.fillRect(0, 0, 150, 100);**

**</script>**

**案例2**

**<canvas id="myCanvas" style="border:1px solid;" width="300" height="300"></canvas>**

**<script type="text/javascript">**

**function draw() {**

**var ctx = document.getElementById('myCanvas').getContext("2d");**

**// 绘制背景。**

**ctx.fillStyle = "black";**

**ctx.fillRect(0, 0, 300, 300);**

**ctx.fill();**

**// 绘制圆形。**

**ctx.beginPath();**

**ctx.arc(150, 150, 130, 0, Math.PI \* 2, true);**

**// 裁切路径。**

**ctx.clip();**

**ctx.translate(200, 20);**

**for (var i = 1; i < 90; i++) {**

**ctx.save();**

**ctx.transform(0.95, 0, 0, 0.95, 30, 30);**

**ctx.rotate(Math.PI / 12);**

**ctx.beginPath();**

**ctx.fillStyle = "red";**

**ctx.globalAlpha = "0.4";**

**ctx.arc(0, 0, 50, 0, Math.PI \* 2, true);**

**ctx.closePath();**

**ctx.fill();**

**}**

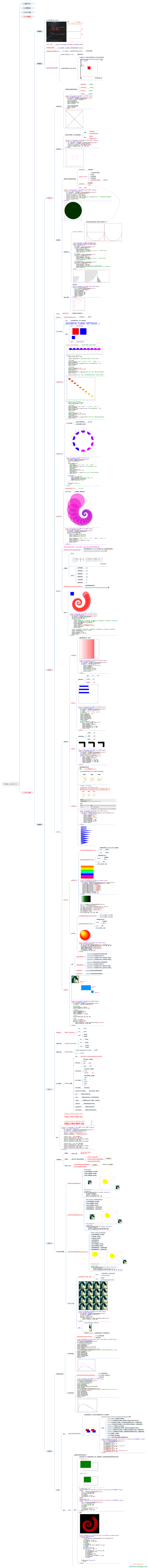
**}**

**window.onload = function () {**

**draw();**

**}**

**</script>**

****